Fleet Organization Models for Online Vehicle Routing Problems

Mahdi Zargayouna[1], Besma Zeddini[2]

[1] Université Paris-Est, IFSTTAR, GRETTIA,
F-93166 Noisy-le-Grand, France
[2] IRSEEM-ESIGELEC, ITS division
Saint Etienne du Rouvray
76801 - France
hamza-mahdi.zargayouna@ifsttar.fr, zeddini@esigelec.fr

Abstract. Online vehicle routing problems with time windows are highly complex problems for which different artificial intelligence techniques have been used. In these problems, the exclusive optimization of the conventional criteria (number of vehicles and total traveled distance) leads to the appearance of geographic areas and/or time periods that are not covered by any vehicle because of their low population density. The transportation demands in these zones either cannot be satisfied or need to mobilize new vehicles. We propose two agent-oriented models that propose a particular dynamic organization of the vehicles, with the objective to minimize the appearance of such areas. The first model relies on a spatial representation of the agents' action zones, and the second model is grounded on the space-time representation of these zones. These representations are capable of maintaining an equilibrated distribution of the vehicles on the transportation network. In this paper, we experimentally show that these two means of distributing vehicles over the network provide better results than traditional insertion heuristics. They allow the agents to take their decisions while anticipating future changes in the environment.

Keywords: Vehicle Routing Problems, Multiagent Systems, Organization Models

1 Introduction

Several operational distribution problems, such as the deliveries of goods to stores, the routing of school buses, the distribution of newspapers and mail etc. are instantiations of NP-Hard theoretical problems called the Vehicle Routing Problems (VRP). In its original version, a VRP is a multi-vehicle Traveling Salesman Problem: there exists a certain number of nodes to be visited once by a limited number of vehicles. The objective is to find a set of vehicles' routes that minimizes the total distance traveled. Besides their practical usefulness, the VRP and its extensions are challenging optimization problems with academic stimulating issues. One of the most widely studied variant of the problem is the time (and capacity) constrained version: the Vehicle Routing Problem with Time Windows (VRPTW henceforth) [1], in which the requests to be handled are not simply nodes, but customers. For each customer, the following information are provided: the concerned node, two temporal bounds between which she desires to be visited and a quantity (number of goods to receive, number of persons to transport, etc.). Every vehicle has a limited capacity, which should not be exceeded by the sum of the quantities associated with the customers it visits. The addition of time windows to the basic problem restrains considerably the space of valid solutions.

The VRP and the VRPTW can be divided in two categories: static problems and dynamic problems. The distinction between these two categories relies traditionally on the knowledge (static problem) or the ignorance (dynamic problem) before the start of the solving process of all the customers that have to be visited. The operational problems are rarely fully static and we can reasonably say that today a static system cannot meet the mobility needs of the users. Indeed, in operational settings, and even if all the customers are known in advance (before the execution start), there always exists some element making the problem dynamic. These elements include breakdowns, delays, no-shows, etc. It is thus always useful to consider a problem that is not fully static.

We rely on the multiagent paradigm for solving the dynamic VRPTW. An agent is a software system, that is situated in some environment and that is able to apply autonomous actions to satisfy its goals [2], and a MAS is a loosely coupled network of agents that interact to solve problems that are beyond the individual capabilities or knowledge of each one [3]. A multiagent modeling of the dynamic VRPTW is relevant for the following reasons. First, since it's a hard problem, choosing a design allowing for the distribution of computation can be a solution to propose short answer times to customers requests. Second, with the technological developments, it is reasonable to consider vehicles with onboard calculation capabilities. In this context, the problem is, *de facto*, distributed and necessitates an adapted modeling to take profit of the onboard equipments of the vehicles. Finally, the consideration of a multiagent point of view allows to envision new measures, new heuristics, not envisaged by centralized approaches. Even if the multiagent approach does not guarantee optimal solutions, it is often capable of finding satisfactory solutions in faster execution times [2].

The MAS that we propose in this paper simulate a distributed version of the so-called "insertion heuristics". Insertion heuristics are methods that consist in inserting the customers following their appearance order in the routes of the vehicles. The vehicle chosen to insert the considered customer is the one that has to make the minimal detour to visit her. Several MAS in the literature have been proposed to distribute insertion heuristics, but very few propose new measures for the insertion cost of a customer in the route of a vehicle, as an alternative to the traditional measure of its incurred detour. In the present work, we do propose two new measures, in the context of two new organization models. They are based on a space and on a space-time representation of the vehicle agents' action zones (the zone inside which all vehicle's actions take place). The objective is to allow the MAS to self-adapt exhibiting an equilibrated distribution of its vehicle agents, and to decrease this way the number of vehicles mobilized to serve the customers. Indeed, when providing an equilibrated distribution, the MAS is more reactive to customers' requests, which appear nondeterministically in space and time.

The remainder of this paper is structured as follows. In section 2, we discuss previous proposals for the dynamic VRPTW w.r.t our approach. We provide the formal definition of the problem in section 3. The architecture of the MAS is presenbted in section 4. In sections 5 and 6, we detail the models and the use of new measures for the insertion decisions of the vehicles. We report on our experimental results in section 7 and then conclude with a few remarks.

2 Related Work

2.1 Combinatorial Optimization

Exact approaches cannot meet operational settings of VRPTW, and upon the relatively small set of benchmarking problems of [1] - 56 problems of 100 Euclidean customers[3] each, only 45 have a known optimal solution up until today [4]. However, interested readers of optimization approaches can refer to, e.g. [5] for a survey. In fact, most of the proposed solution methods are heuristic or metaheuristic methods, which provide good results in non-exponential times, and which have presented good results with benchmark problems. For instance, large-neighborhood local search [6], simulated annealing [7, 8], evolutive strategies [9] and ant colonies [10, 11] present the best performances with static problems (where the set of transport requests is known *a priori*). For an extensive survey of the literature for the VRPTW approaches, the reader is invited to refer to, e.g. [12, 13].

Generally speaking, most of the works dealing with the dynamic VRPTW are more or less direct adaptations of static methods. For instance, the large-neighborhood local search is adapted to a dynamic context in [14]. In [15], the authors propose to adapt the genetic algorithms to deal with the dynamic VRPTW. The proposed algorithm starts by creating a population of initial solutions and tries continually to improve their quality. When a new customer reveals, she is inserted in all current solutions in the position minimizing the additional cost. Upon the static methods, insertion heuristics are the most widely adapted in a dynamic environment (e.g. [16–19]). Insertion heuristics are, in their original version, greedy algorithms, in the sense that the decision to insert a given customer in the route of a vehicle is definitive. They are also combined with metaheuristics to improve the quality of the solutions. The advantage of using insertion heuristics is that they are intuitive and fast. However, when they are applied in a dynamic context, their solving process is said to be short-sighted. Indeed, the system doesn't know which customers will appear once it has assigned the known

[3] Euclidean customers have cartesian coordinates, and the distance and the le travel times between each pair of customers are calculated following the Euclidean metric.

customers to the vehicles; and even if we could have an optimal assignment and scheduling of the known customers, a new coming customer could make the old assignment sub-optimal, which would - in the worst case - necessitate a whole recomputation of all the routes. While preventing this reconsideration of previous decisions, insertion heuristics exhibit the fastest execution times but suffer a serious handicap.

Nevertheless, in their wide majority, agent-oriented approaches of the literature rely, at least partially, on insertion heuristics.

2.2 Multiagent Systems

In [20], the authors propose a multiagent architecture to solve a VRP and a multi-depot VRP. In [21], the authors propose a multiagent architecture to solve a dial-a-ride problem. The principle of these two proposals is the same: distribute an insertion heuristic, followed by a post-optimization step. In [20], the customers are handled sequentially. They are broadcasted to all the vehicles, which in turn propose insertion offers and the best proposal is retained by the customer. In the second step, the vehicles exchange customers to improve their solutions, each vehicle knowing the other agents of the system. Since vehicles are running in parallel, the authors envision to apply different heuristics for each vehicle, without changing the architecture. In-Time [21] is a system composed of *customer* agents and *vehicle* agents. The customer agent announces itself and all the vehicle agents calculate its insertion cost in their routes. Again, the customer agent selects the cheapest offer. The authors propose a distributed local search method to improve the solutions. Indeed, they allow a customer to ask stochastically to cancel its current assignment and to reannounce itself to the system, with the objective of having a better deal with another vehicle. MARS [22] models a cooperative scheduling in a maritime shipping company in the form of a multiagent system. The solution to the global scheduling problem emerges from the local decisions. The system uses an extension of the Contract Net Protocol (CNP) [23] and shows that it can be used for having good initial solutions to complex problems of tasks assignment. The MAS profits from an *a priori* structuring of the agents, since each vehicle is associated with a particular company and can handle the only customers of this company.

For the reasons that we have given in the introduction, we choose a multiagent modeling to solve the dynamic VRPTW. For their fast execution times and their adaptation to dynamic settings, we privilege a solving grounded on insertion heuristics. Thus, from a protocol and an architecture point of view, our system sticks with the multiagent systems we have just described, since we propose a distributed version of insertion heuristics. However, in these proposals, none have focused on the redefinition of the insertion cost of a customer. The traditional insertion cost of a customer in the route of a vehicle is based on the incurred detour of the vehicle. We propose two new insertion cost measures, focused on the space and space-time coverage of the transportation network by the vehicles. Our goal is to counterbalance the short vision of the traditional measures, by privileging an insertion process that is future-centered.

3 Problem Definition

In the following, we provide a formal definition of the VRPTW in order to define the parameters and the constraints of the system in an unambiguous way. It is noteworthy that although the objective in this definition is to minimize the routes global costs, the hierarchical and primary objective of minimizing the number of vehicles is traditionally used [4]. Indeed, the size of the vehicles fleet is not fixed when the system does not propose an exact solving of the problem. This size becomes a criterion to minimize.

Definition 1 (VRPTW). *An instance* $I = (G, D, T, S, F, R, \kappa)$ *of the VRPTW is defined as follows. Let* $G = (V, E)$ *a graph with a set of nodes* $V = \{(v_i)\}, i = \{0, ..., N\}$ *(node* v_0 *is the depot) and a set of arcs* $E = \{(v_i, v_j)|v_i \in V, v_j \in V, v_i \neq v_j\}$*. Let two matrices* $D = \{(d_{ij})\}$ *et* $T = \{(t_{ij})\}$ *of costs, of dimensions* $N \times N$ *(the arc* (v_i, v_j) *has a distance of* d_{ij} *and a travel time of* t_{ij}*), a M-array* F *of vehicles, and a N-array* R *of tuples (*R *for requests)* $(q_i, s_i, [e_i, l_i])$ *(node* v_i *has a demand* q_i*, a service time* s_i *and a time window* $[e_i, l_i]$*,* $q_1 = (0, 0, [e_0, l_0])$*). The window* $[e_0, l_0]$ *is the "scheduling horizon" of the problem. All the time windows have to be comprised between these two bounds. A vehicle has to be in* i *before* l_i *but can be in* i *before* e_i*, in which case it has to wait for the service start. Every request is supposed to be inferior to* κ*.*

Two decision variables are defined: $X = (x_{ijk})$ *of dimension* $N \times N \times M$ *and* $B = (b_i)$ *of dimension* N *having the following interpretation:*

$$x_{ijk} = \begin{cases} 1 \text{ if the vehicle } k \text{ visits node } v_i \text{ immediately after node } v_j \\ 0 \text{ otherwise} \end{cases}$$

$b_i = t \Leftrightarrow v_i$ *is visited at* t

The objective function:

$$\min \sum_{i,j=0}^{N} d_{ij} \sum_{k \in F} x_{ijk} \tag{1}$$

Solving a VRPTW consists in finding X and B optimizing the objective function for all instance of I subject to the following constraints:

$$\sum_{k \in F} \sum_{j=1}^{N} x_{ijk} = 1 \; \forall v_i \in V \backslash v_0 \tag{2}$$

$$\sum_{j=1}^{N} x_{0jk} = 1 \; \forall k \in F \tag{3}$$

$$\sum_{i=0}^{N} x_{ijk} - \sum_{i=0}^{N} x_{jik} = 0 \ \forall k \in F, \forall v_j \in V \backslash v_0 \tag{4}$$

$$\sum_{j=1}^{N} x_{j0k} = 1 \ \forall k \in F \tag{5}$$

$$\sum_{i=0}^{N} q_i \sum_{j=0}^{N} x_{ijk} \leq \kappa \qquad \forall k \in F \tag{6}$$

$$x_{ijk}(b_i + s_i + t_{ij} - b_j) \leq 0 \ \forall k \in F, \forall (v_i, v_j) \in E \tag{7}$$

$$e_i \leq b_i \leq l_i \ \forall k \in F, \forall v_i \in V \tag{8}$$

$$x_{ijk} \in \{0, 1\}, \forall (v_i, v_j) \in E \ \forall k \in F \tag{9}$$

The function (1) expresses the system's objective: the minimization of overall cost. Constraints (2) restrict the assignment of every customer (but the depot v_0) to exactly one vehicle. Constraints (3) to (5) characterize the path to follow by a vehicle k: k has to leave the depot only once (3), for every served customer (if any), it has to leave it (4) and get back to the depot exactly once (5). Constraints (6) guarantee the non-violation of the capacity limits of the vehicles. Constraints (7) $-$ (8) ensure the non-violation of time constraints.

4 The Multiagent Systems Architectures

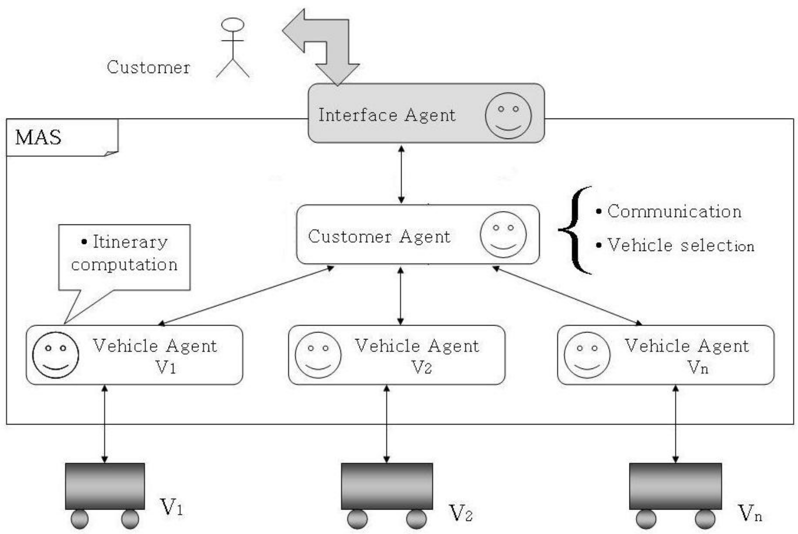
Our systems are composed of a dynamic set of agents which interact to solve the dynamic VRPTW. A solution consists of a series of vehicles routes. Each route contains a sequence of customers with their associated visit time. We define three categories of agents. Customer agents, which represent users of the system (persons or goods), vehicle agents, which represent vehicles in the MAS and interface agents which represent an access point to the system (Web server, GUI, simulator, etc.). When a user logs in the MAS, the data she provides are verified (existing node, valid time windows, etc.) and, if the data are correct, a customer agent representing her is created.

In [24], we have designed, implemented and compared three possible architectures to model the dynamic VRPTW: a centralized architecture, a decentralized architecture and a hybrid architecture. The hybrid architecture has provided the best results in terms of robustness and execution times.

In the centralized approach, all the processing is performed by a central entity, which create vehicle plans and schedules. In the decentralized approach, vehicle agents exchange messages trying to insert the customers that reveal online. Each vehicle agent is responsible of the customers scheduling in its plan.

The hybrid architecture (cf. Fig. 1) is a compromise between the centralized and the decentralized approach. Indeed, the vehicle agents don't exchange messages, they all interact with the new coming customer agent. Once created, the customer agent sends its request to all the available vehicles, collects bids from the vehicle agents and chooses the one offering the best cost. The process describes a CNP (Contract Net Protocol) [23]. If there is no vehicle agent that can insert the customer, a new vehicle is created. Finally, the customer agent informs the vehicles about its choice. The question is now to define the criteria to choose the best vehicle candidate for the insertion of the new customer.

Fig. 1. MAS Architecture



We observe that the direct and exclusive focus on the conventional criteria for the VRPTW (the traveled distance and the fleet size) leads to the appearance of uncovered areas because of their low density. Indeed, the fact that we deal with a dynamic and nondeterministic problem can lead to the appearance of two different but non independent phenomena. The first is the concentration of vehicles in some geographical zones which are more attractive and may lead to the second phenomenon, which is the lack of service elsewhere. The idea behind the organization models - that we detail in the following - is that when the positioning of vehicles is made such as to cover as much territory as possible, the risk of customers whose demand is unsatisfied, and the obligation to mobilize new vehicles to serve them, decreases. The choice that we make to solve this problem is to use the multiagent paradigm coupled with insertion heuristics. In this context, we have only one lever to change the system's behavior, which is the way in which the vehicle agents calculate the insertion cost of a customer. These calculation methods are two dimensional: spatial and spatiotemporal. The two organization models that we propose have the objective of minimizing the number of used vehicles, while keeping the use of a "pure" insertion heuristics, i.e. without any further improvements or post-optimization.

Following the description above, the customer agent chooses between several vehicle agents the one with the minimal proposed insertion cost. The systems that are based on this heuristic use generally the measure of Solomon [1] as an insertion cost. This measure consists in inserting the customer which has the minimal impact on the general cost of the vehicle (which is generally function of the vehicle's incurred detour). This measure is simple and the most intuitive but has a serious drawback, since inserting the current customer might make lots of future customers' insertions infeasible, with the current number of vehicles. Its problem is that it generates vehicles' plans that are very constrained in time and space, i.e. plans that offer a few possibilities of insertion between each pair of adjacent planned customers. In this situation, the appearance of a new customer might oblige the system to create a new vehicle to serve it. Through the modeling of vehicle agents' action zones, we propose a new way to compute the customer's insertion cost in the route of a vehicle, and a new choice criterion between vehicles for the insertion of a given customer. We propose a new method that allows the customer to choose the vehicle agent "whose decrease in the probability to participate in future insertions is minimal", to serve the new customer. The logic of our models is different from the traditional models, which focus on the increase of the traveled distance, neglecting the impact of the current insertion decision on future insertion possibilities.

5 Spatial Organization Model

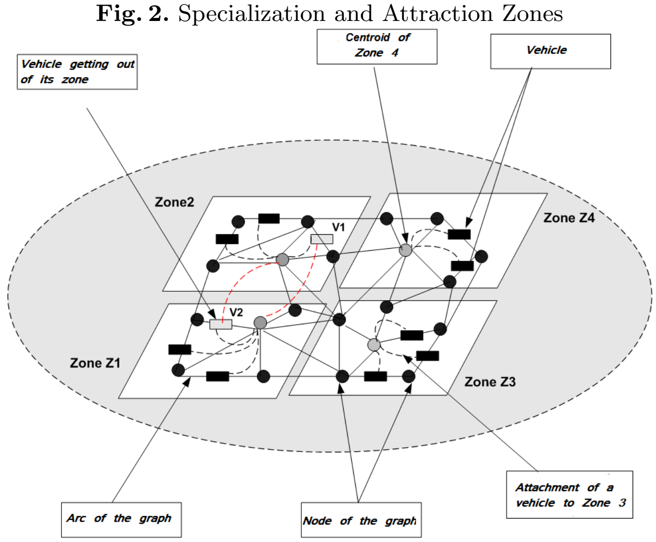
The objective of the spatial organization model is to allow the specialization of the vehicles to zones while maintaining a wide coverage of the network (cf. Fig. 2). Thus, we define action zones on the transportation network, to which the vehicles are attached. The attachment of vehicles to their zones is not encoded in their behavior, but has an effect on how they calculate their customers insertion costs. This computation should ensure that a vehicle agent plans its itinerary while being encouraged to stay in its zone[4]. Each zone is defined by a set of nodes and a centroid. In Fig. 2, vehicles V1 and V2 might be tempted to leave their zones to serve new customers, the mechanism that we propose should make it more expensive for them to do so.

Definition 2 (Spatial Action Zone). *Given $G = (V, A)$ the graph describing the network (cf. Definition 1), we define the zone $\zeta = (N_\zeta, A_\zeta)$ as a subgraph of G.*

[4] The segmentation of the network in geographical zones is treated as a graph partitioning problem and is left out of the scope of this paper. We assume that the definition of these zones is a system parameter, which is the responsibility of an expert.

Definition 3 (centroid of a Zone). *The centroid of zone ζ is a node $n^{*\zeta}$ of N that minimizes $\sum_{y \in N_\zeta} d_{n^{*\zeta},y}$.*

Fig. 2. Specialization and Attraction Zones



Each zone is defined by a centroid and a set of nodes (cf. Fig. 3). The centroid of a zone corresponds to the node which is the closest to all other nodes in the zone. At any point in time, each vehicle agent has a distance from its action zone. This distance is equal to the average distance of the nodes in its route from the centroid of its zone. It depends of the customers inserted in its route. If the vehicle has a node in its route that is outside its zone, the distance of this node is multiplied by a factor β (> 1) which is a system parameter. Since the insertion cost proposed by a vehicle agent to the customer depends of the vehicle distance from its zone (see definition 5), the penalty β discourages the vehicle from inserting customers that are located outside its action zone.

Definition 4 (Vehicle Distance from its Zone). *The distance of a vehicle v from its zone ζ_v at a given moment is equal to the average distance of the nodes in its route from the centroid of ζ_v:*

$$d_{v,\zeta_v} = \frac{\sum_{n^v \in Nodes(v)} d_{n^v,n^{*\zeta}}}{|Nodes(v)|} \tag{10}$$

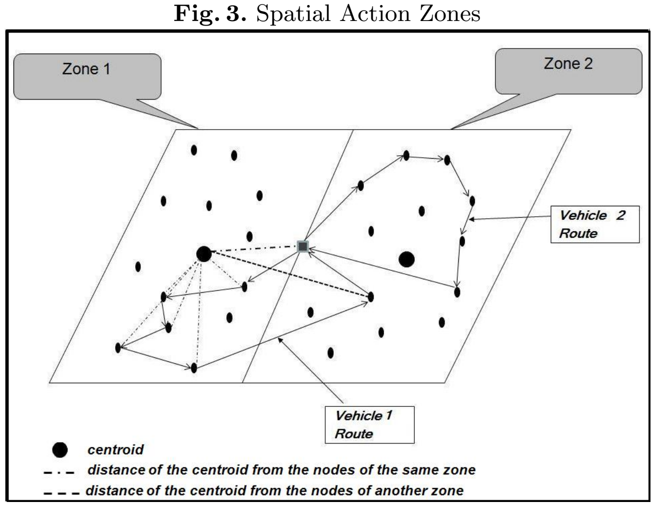
with

$$\forall c \in N, d_{n^v,c} = \begin{cases} d_{n^v,c}, & \text{if } n^v \in \zeta_v \\ \beta \times d_{n^v,c} & \text{else} \end{cases}$$

Nodes(v) represents the nodes of the vehicle agent's route, $\mid Nodes(v) \mid$ is the number of nodes in $Nodes(v)$ and β is the penalty imposed to the vehicle distance, if its route integrates nodes which are outside its zone.

The objective of the new measure is to encourage vehicles to stay in the vicinity of network zones to which they are allocated. This is done by integrating in the insertion cost, besides the increase in the traveled distance, a factor that is function of the distance from the centroid, and a penalty if it had to leave its zone to satisfy a request. The insertion cost of a customer in the route of a vehicle agent becomes equal to the incurred detour to insert the new customer, multiplied by the variation of the vehicle agent's distance from its zone (we denote $v \succ c^*$ the vehicle v with c^* in its route).

Fig. 3. Spatial Action Zones



Definition 5 (Insertion cost of a customer). *The insertion cost proposed by the vehicle for the insertion of the customer c^* is equal to:*

$$cost(v, c^*) = \frac{d_{v \succ c^*, \zeta_v}}{d_{v, \zeta_v}} \times (Dist_{v \succ c^*} - Dist_v) \tag{11}$$

$Dist_v$ is the total distance traveled by vehicle v. There exists several possible insertion positions of a customer c^* in the route of a vehicle agent. To each of these positions, there corresponds a value of $cost(v, c^*)$. The considered cost is the one for which $cost(v, c^*)$ is minimal.

The offer that a vehicle agent proposes to a customer for its insertion is then weighted by the difference between the old distance of the vehicle from its zone and its new one. The bigger β is, the more the vehicles are organized so that they stay in their zones. When a vehicle plans to quit it zone, it is penalized with an increase of its insertion cost and sees thus its competitiveness limited w.r.t the other vehicles that are candidates for the insertion of the considered customer. This penalty plays the role of an attractive force exerted on the vehicle. As if a sort of a spring were fixed to the centroid of the zone and to the vehicle.

Unlike ADART [25], our spatial action zones are not rigid. Indeed, a vehicle in our system has the right to quit its zone, with a penalty, while it can't at all in [25]. This choice is motivated by the fact that insertion heuristics consider only a very small subset of all the possible routes combinations for a given set of customers, and this is precisely why they are so fast. Narrowing this set even more, by completely preventing vehicles from leaving their zones, would limit the search space more and would lead, *In Fine*, to the mobilization of new vehicles to serve the unsatisfied customers.

6 Space-Time Organization Model

Even if it allows a better spatial coverage of the network, the spatial organization model has two major drawbacks. First, it assumes *a priori* geographical segmentation. This task requires a great calibration effort to specify the most efficient zones' segmentation. Second, it doesn't incorporate the temporal dimension of the problem, since a vehicle might not be able to serve a customer even if it is located in its zone, because of the time constraints. In the following, we propose to integrate the temporal dimension in the vehicle agents' action zones and to eliminate any *a priori* definition of these zones.

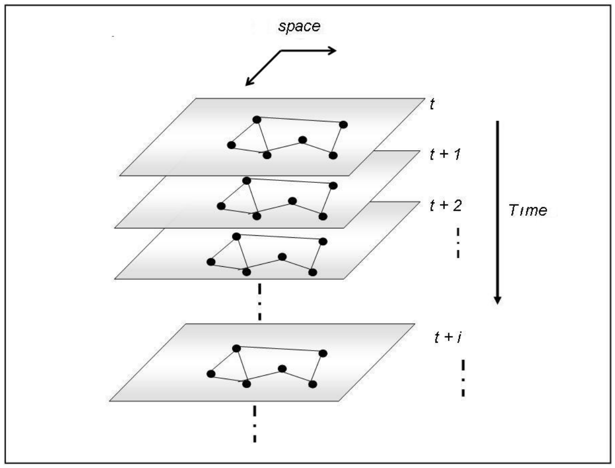
In the heuristics and multiagent methods of the literature, the hierarchical objective of minimizing the number of mobilized vehicles is considered in priority w.r.t the distance traveled by all the vehicles. The vast majority of the literature heuristics are as a consequence based on a two-phase approach: the minimization of the number of vehicles, followed by the minimization of the traveled distance [4]. The model that we propose in this section has the objective of minimizing the number of used vehicles.

To this end, our model allows vehicle agents to cover a maximal space-time zone of the transportation network, avoiding this way the mobilization of a new vehicle if a new customer appears in an uncovered zone [26]. A space-time pair $\langle i, t \rangle$ - with i a node and t a time - is said to be "covered" by a vehicle agent v if v can be in i at t. In the context of the dynamic VRPTW, maximizing the space-time coverage of vehicle agents results in giving the maximum chance to satisfy the demand of a future (unknown) customer. Through the new modeling of vehicle agents' space-time action zones, we propose a new way to compute the customer's insertion cost in the route of a vehicle.

6.1 Environment Modeling

The space-time action zone of a vehicle agent is composed of a subset of the network nodes, together with the times that are associated to them. We model the MAS environment in the form of a space-time network, inferred from the network graph. Each node of the graph becomes a pair $\langle space, time \rangle$, which represents the "state" of the node in a discrete time period. The space-time network is composed of several subgraphs, where each subgraph is a copy of the static graph, and corresponds to the state of the graph in a certain period of time

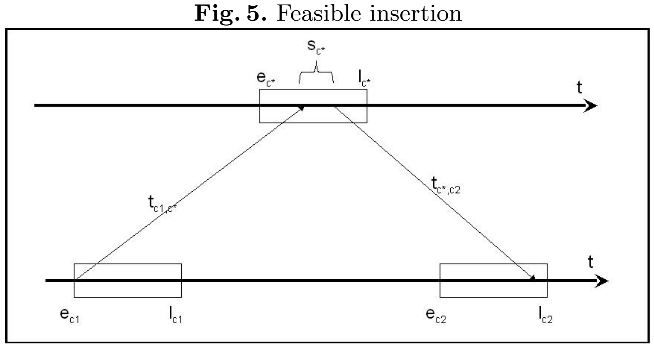
Fig. 4. Space-Time Network



(cf. Fig. 4). We index the nodes of a subgraph as follows: $\langle 0, t\rangle, \ldots, \langle N, t\rangle$, with $t \in \{1, ..., h\}$, where $0, \ldots, N$ are the nodes of the network and h the number of considered discrete periods. The total number of nodes in the space-time network is equal to $h \times N$. The edges linking the nodes of a subgraph are those of the static graph, and the costs are the travel times as described in the problem definition.

6.2 Intuition of the Space-Time Action Zones

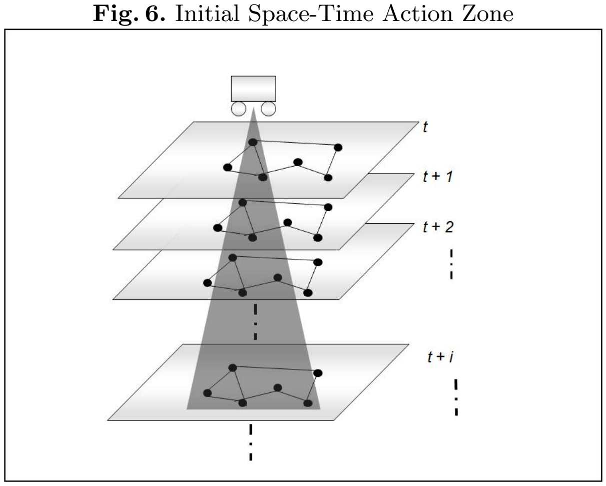
Fig. 5. Feasible insertion



Consider a vehicle agent v that has an empty route. In order for this agent to be able to insert a new customer c - described by: n a node, $[e, l]$ a time window,

s a service time, and q a quantity - l has to be big enough to allow v to be in n without violating its time constraints[5]. More precisely, the current time t, plus the travel time between the depot and n has to be less or equal to l (cf. Fig. 5). Starting from this observation, we define the action zone of a vehicle agent as agent as the set of pairs $\langle n, t \rangle$ of the space-time network that remain valid given its current route (n can be visited by the vehicle at t). The action zone of a vehicle agent with an empty route is illustrated by the triangular shadow[6] in the Fig. 6.

Fig. 6. Initial Space-Time Action Zone



When a vehicle agent inserts a customer in its route, its action zone is recomputed, since some $\langle node, time \rangle$ pairs become not valid because of its insertion. In Fig. 7, a new customer is inserted in the route of the vehicle. The action zone of the vehicle agent after inserting the customer is represented by the interior of the contour of the bold lines, which represent the space-time nodes which remain accessible after the insertion of the customer (the computation of the new action zone is explained later).

The associated cost to an offer from a vehicle agent v for the insertion of a customer agent c corresponds to the hypothetical decrease of the action zone of v following the insertion of c in its route.

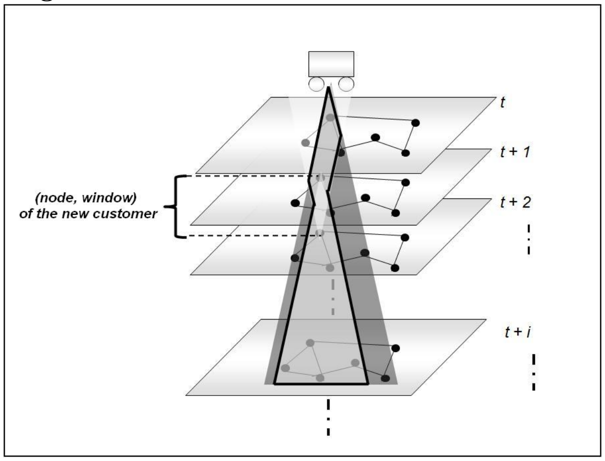
The idea is that the chosen vehicle for the insertion of a customer is the one that looses the minimal chance to be candidate for the insertion of future customers. Thus, the criterion that is maximized by the society of vehicle agents is the sum of their action zones, i.e. the capacity that the MAS has to react to the appearance of customer agents, without mobilizing new vehicles.

---

[5] Note that we assume that only one customer is considered by the vehicle agents until it is inserted in one of their routes.

[6] it is actually a conic shadow in a three-dimensional space

Fig. 7. Action Zone after the Insertion of a Customer



To illustrate the action zones and their dynamics, we present the version of the measure that is related to an Euclidean problem, i.e. where travel times are computed following the Euclidean metric. The following paragraphs detail the measure as well as its dynamics.

6.3 The Computation of Action Zones

In the Euclidean case, the transportation network is a plane, and the travel times between two points i (described by (x_i, y_i)) and j (described by (x_j, y_j)) is equal to

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}(12) \tag{12}$$

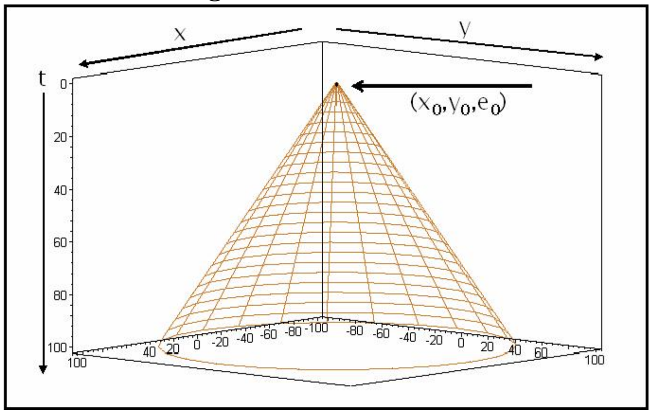
Therefore, if a vehicle is in i at the moment t, it cannot be in j earlier than $t_i + \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

We can compute at any time, from the current position of a vehicle, the set of triples (x, y, t) where it can be in the future. Indeed, considering a plane with an X-axis in $[x_{min}, x_{max}]$ and a Y-axis in $[y_{min}, y_{max}]$, the set of space-time positions is the set of points in the cube delimited by $[x_{min}, x_{max}]$,$[y_{min}, y_{max}]$ and $[e_0, l_0]$ (recall that e_0 and l_0 are the scheduling horizon and are the minimal and maximal values for the time windows). Consider a vehicle in the depot (x_0, y_0) at t_0. The set of points (x, y, t) that are accessible by this vehicle are described by the following inequality:

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} \leq (t - t_0)(13) \tag{13}$$

The (x, y, t) satisfying this inequality are those that are positioned inside the cone $\mathcal{C}$ of vertex (x_0, y_0, t_0) and with the equation $\sqrt{(x - x_0)^2 + (y - y_0)^2} = (t - t_0)$ (c.f Fig. 8). This cone represents the action zone of a vehicle agent, with an empty

Fig. 8. Initial Action Zone



route, in the Euclidean case. It represents all the possible space-time positions that this vehicle agent is able to have in the future.

We use the action zone of the vehicle agents when a customer agent has to choose between several vehicle agents for its insertion. We have to be able to compare the action zones of different vehicle agents. To do so, we propose to quantify it, by computing the volume of the cone $\mathcal{C}$ representing the future possible positions of the vehicle:

$$Volume(\mathcal{C}) = \frac{1}{3} \times \pi \times (l_0 - e_0)^3 \tag{14}$$

This is the quantification of the initial action zone of any new vehicle agent joining the MAS. When a new customer agent appears, a vehicle agent computes its new action zone, the cost that it proposes to the customer agent is the difference between its old action zone and its new one. The new action zone computation is detailed in the following paragraph.

6.4 Dynamics of the Action Zones

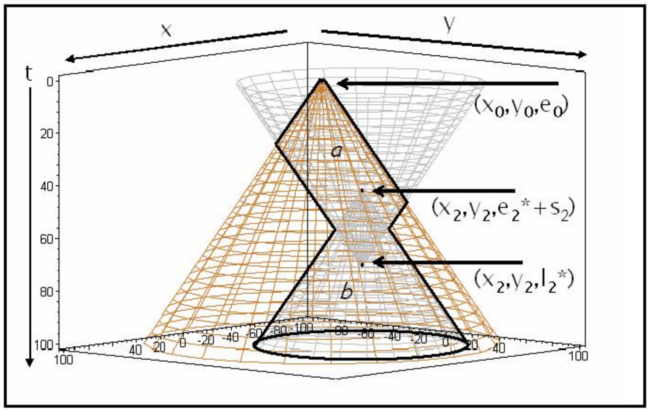
Consider a customer c_2 (of coordinates (x_2, y_2) and with a time window $[e_2, l_2]$) that joins the system, and suppose that v is temporarily the only available vehicle agent of the system and has an empty route. The agent v has to infer its new space-time action zone, i.e. the space-time nodes that it can still reach without violating the time constraints of c_2. The new action zone answers the following questions: "if v had to be in (x_2, y_2) at l_2, where would it have been before? And if it had to be there at e_2 where would it be after $e_2 + s_2$?". The triples (x, y, t) where the vehicle agent can be before visiting c_2 are described by the inequality (15), and the triples (x, y, t) where he can be after visiting c_2 are describe by the inequality (16).

$$\sqrt{(x - x_2)^2 + (y - y_2)^2} \leq (l_2 - (t)) \tag{15}$$

$$\sqrt{(x-x_2)^2+(y-y_2)^2} \leq (t-(e_2+s_2)) \quad (16)$$

The new action zone is illustrated by the Fig. 9: the new measure consists of the volume of the intersection of the initial cone $\mathcal{C}$ with the union of the two new cones described by the inequalities (15) and (16) (denoted respectively by $\mathcal{C}_1$ and $\mathcal{C}_2$). The new measure of the action zone is equal to the volume of the intersection[7] of $\mathcal{C}$ with the union of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Fig. 9. Space-Time Action Zone after the insertion of c_2



The cost of the insertion of a customer in the route of a vehicle is equal to the measure associated with the old action zone of the vehicle minus the measure of the new action zone, after the insertion of the customer. The measured quantity represents the space-time positions that the vehicle cannot have anymore, if it had to insert this customer in its route. The retained vehicle agent is the one for which the insertion of the new customer causes the minimum loss in its space-time action zone. This corresponds to choosing the vehicle that looses the minimal chances to be candidate for future customers.

6.5 Coordination of Action Zones

The objective of the space-time organization model is to allow a better space-time coverage of the transportation network. This improvement is materialized by a minimal mobilization of vehicles when confronted to the appearance of new customers. With the mechanism described until now, every vehicle agent tries to maximize its own action zone independently from the other agents of the MAS. However, it would be more interesting to incite the agents society in its whole to cover the network in the most efficient way. More precisely, the fact that a

[7] The complete computation of the volume of the intersection of these two cones is reported in the Appendix A of [26].

vehicle looses space-time nodes that it is the only one to cover should be more costly than to loose nodes that are also covered by other agents.

To this end, to every node of the space-time network, we start by associating the list of vehicles covering it. Then, to every creation of a new vehicle agent, the set of space-time nodes that are part of its action zone is computed. The vehicle proceeds then with the notification of these nodes that they are part of its action zone. Every node updates its list, containing the vehicles that are covering it, at each notification from a vehicle agent. Similarly, when the action zone of a vehicle agent loses a node, the node is notified and its list of vehicles is updated.

Now, when the insertion cost of a customer is computed, every vehicle agent starts by determining the space-time nodes that it would loose if it had to insert the new customer. Then, it interrogates each of these nodes about the "price to pay" if it were not covering it anymore. This price is inversely proportional to the number of vehicles covering this node. More precisely, the price to pay is equal to

$$\frac{1}{\mid v_{\langle n,t\rangle} \mid} \tag{17}$$

with $v_{\langle n,t\rangle}$ denoting the vehicle agents covering the space-time node $\langle n,t\rangle$ and $\mid v_{\langle n,t\rangle} \mid$ the number of such vehicles.

This way, more or less penalty is associated with the decisions of non-coverage of the network by the vehicles as time progresses. Thus, the vehicle agents are incited to cover the whole network in a coordinated way, improving by doing so the reactivity of the MAS.

7 Results

Marius M. Solomon [1] has created a set of different static problems for the VRPTW. It is now admitted that these problems are challenging and diverse enough to compare with enough confidence the different proposed methods. In Solomon's benchmarks, six different sets of problems have been defined: C1, C2, R1, R2, RC1 and RC2. The customers are geographically uniformly distributed in the problems of type R, clustered in the problems of type C, and a mix of customers uniformly distributed and clustered is used in the problems of type RC. The problems of type 1 have narrow time windows (very few customers can coexist in the same vehicle's route) and the problems of type 2 have wide time windows. Finally, a constant service time is associated with each customer, which is equal to 10 in the problems of type R and RC, and to 90 in the problems of type C. Short service times would represent problems where the loading and unloading of the transported entities is fast (transport of persons for instance). In every problem set, there are between 8 and 12 files containing 100 customers each.

We choose to use Solomon benchmarks, while following the modification proposed by [27] to make the problem dynamic. To this end, let $[0, T]$ the simulation time. All the time related data (time windows, service times and travel times) are multiplied by $\frac{T}{l_0 - e_0}$, with $[e_0, l_0]$ the scheduling horizon of the problem. The authors divide the customers set in two subsets, the first subset defines the customers that are known in advance, and the second the customers who reveals during execution. We do not make this distinction, since we consider no customers known in advance. For each customer, an occurrence time is associated, defining the moment when the customer is known by the system. Given a customer i, the occurrence time that is associated is generated randomly between $[0, \overline{e}_i]$, with:

$$\overline{e}_i = e_i \times \frac{T}{l_0 - e_0} \tag{18}$$

It is known that the behavior of insertion heuristics is strongly sensitive to the appearance order of the customers to the system. For this reason, we do not consider only one appearance order. We launch the process that we have just described ten times with every problem file, creating this way ten different versions of every problem file.

We have implemented three MAS with almost the same behavior, the only difference concerns the measure used by vehicle agents to compute the insertion cost of a customer. For the first implemented MAS, it relies on the Solomon measure (noted Δ Distance). The second relies on the space-time model (noted Δ Space-Time) and the third on the spatial model (noted Δ Space)[8]. We choose to run our experiments with the problems of class R and C, of type 1, which are the instances that are very constrained in time (narrow time windows). Our system is coded in JAVA and executed on a PC with a Core 2 Duo® 2.77 GHZ processor, with 4 GB of RAM.

Table 1. Results summary (Criterion: Fleet Size)

	Δ Distance	Δ Space-Time	Δ Space
Problem	|Fleet|	|Fleet|	|Fleet|
R1 25 customers	64	53	58
C1 25 customers	34	31	32
R1 50 customers	107	92	101
C1 50 customers	60	53	58
R1 100 customers	181	150	164
C1 100 customers	121	108	113

For each problem class and type, we have considered different customers numbers in order to verify the behavior of our models w.r.t to the problem

[8] After several test runs, we set the penalty β to 1.2 and the geographical zones to four equal zones; these values have provided the best results. As we said earlier, the optimal definition of zones is a hard problem and is left out of the scope of this paper.

size. To this end, we have considered successively the 25 first customers, the 50 first customers, and finally all the 100 customers contained in each problem file. Table 1 summarizes the results. Each cell contains the best obtained results with each problem class (the sum of all problem files). The results show, with the two classes of problems, that the use of the space-time model mobilizes less vehicles than the spatial model ($53 < 58, 31 < 32, 92 < 101, 53 < 58, 150 < 164, 108 < 113$). However, the spatial model behaves better than the traditional measure, whatever the number of considered customers ($58 < 64, 32 < 34, 101 < 107, 58 < 60, 164 < 181, 113 < 121$). These results validate the intuition of the models, which consists of maximizing the future insertion possibilities for a vehicle agent.

Once this result validated, it is interesting to check the results with respect to the total distance traveled by all the vehicles. Table 2 summarizes the results[9]. With respect to this criterion, the space model behaves better than the two others, while the behavior of the space-time model is less efficient, since it gives better results for the problems C1 with 25 customers and R1 with 100 customers, but is dominated by the traditional measure for the others. The fact remains that our results for both models provide better results than the traditional heuristic, provided the primary objective of the problem, which is to minimize the number of vehicles mobilized by the system.

Table 2. Results summary (Criterion: Total Traveled Distance)

	Δ Distance	Δ Space-Time	Δ Space
Problem	Distance	Distance	Distance
R1 25 customers	6372	6561	5732
C1 25 customers	3167	3152	3014
R1 50 customers	12036	12089	11307
C1 50 customers	6712	7093	6682
R1 100 customers	17907	17348	16680
C1 100 customers	16011	16512	15206

8 Conclusion

In the vehicle routing problems, the exclusive optimization of the conventional criteria leads to the appearance of geographic areas and/or time periods that are not covered by any vehicle because of their low population density. In this paper, we have proposed two agent-oriented organization models for the dynamic VRPTW based on the agents' action zones. The action zones of the vehicle agents reflect their coverage of the transportation network. We use these action zones to reduce the short-sighted behavior of traditional metrics. By optimizing the

[9] In Solomon's benchmarks, there is no unit associated with the distances.

coverage of the environment by the vehicle agents, our models allow the MAS to self-adapt by exhibiting an equilibrated distribution of the vehicles, and to lessen this way the number of vehicles mobilized to serve the customers.

Our current works are oriented in two different directions. We envision to observe the behavior of our two systems following more qualitative criteria, such as the existence of emergent phenomena and their usefulness for the optimization process. Besides, like the quasi-totality of the state-of-the-art proposals, vehicle travel times are static. If the systems are implemented in urban zones or in time periods which are subject to congestion, the vehicle routes might become not valid. To overcome this limitation, we envision to use an options mechanism on customers requests. An option is a reservation for serving a customer by a vehicle. The withdrawal of an option by a vehicle could take place when trafic predictions, which become more and more precise over time, result in the violation of the customer time windows by the vehicle.

References

1. Solomon, M.: Algorithms for the vehicle routing and scheduling with time window constraints. Operations Research 15, 254–265 (1987)
2. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10, 115–152 (1995)
3. Sycara, K.P.: Multiagent Systems. AI Magazine 19, 79–92 (1998)
4. Nagata, Y., Bräysy, O., Dullaert, W.: A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Computers & Operations Research 37, 724–737 (2010)
5. Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D.: Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research 56, 497–511 (2008)
6. Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Computers & Operations Research 34, 2403–2435 (2007)
7. Czech, Z.J., Czarnas, P.: A parallel simulated annealing for the vehicle routing problem with time windows. In: Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands (Spain), 376–383 (2002)
8. Oliveira, H.C.B.d., Vasconcelos, G.C., Alvarenga, G.B.: A multi-start simulated annealing algorithm for the vehicle routing problem with time windows. In: Proceedings of the Ninth Brazilian Symposium on Neural Networks, Washington, DC, USA, IEEE Computer Society, 24– (2006)
9. Mester, D., Brysy, O.: Active guided evolution strategies for large scale vehicle routing problems with time windows. Computers & Operations Research 32, 1593–1614 (2005)
10. Gambardella, L.M., Taillard, E.D., Agazzi, G.: MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M.D., Glover, F., eds.: New Ideas in Optimization, McGraw-Hill (London), 63–76 (1999)
11. Barán, B., Schaerer, M.: A multiobjective ant colony system for vehicle routing problem with time windows. In: Applied Informatics, 97–102 (2003)
12. Golden, B., Raghavan, S., Wasil, E.: The vehicle routing problem, latest advances and new challenges. Volume 43 of Operations research/computer science interfaces. Springer Verlag (2008)

13. Desaulniers, G., Desrosiers, J., Solomon, M., Soumis, F., Cordeau, J.F.: The VRP with time windows. In Vigo, D., Toth, P., eds.: The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM (2002) 157–193
14. Gendreau, M., Guertin, F., Potvin, J.Y., Sguin, R.: Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transportation Research Part C 14, 157–174 (2006)
15. Housroum, H., Hsu, T., Dupas, R., Goncalves, G.: A hybrid ga approach for solving the dynamic vehicle routing problem with time windows. In Society, I.C., ed.: Proceedings of the IEEE Conference on Information and Communication Technologies: from Theory to Applications, Damascus, Syria, 3347–3352 (2006)
16. Madsen, O.B., Ravn, H.F., Rygaard, J.M.: A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Operations Research 60, 193–208 (1995)
17. Fu, L., Teply, S.: On-line and off-line routing and scheduling of dial-a-ride paratransit vehicles. In: Computer-Aided Civil and Infrastructure Engineering. Volume 14. Blackwell Publishers, Oxford (UK) (1999) 309–319
18. Horn, M.E.: Fleet scheduling and dispatching for demand-responsive passenger services. Transportation Research C 10, 35–63 (2002)
19. Diana, M.: The importance of information flows temporal attributes for the efficient scheduling of dynamic demand responsive transport services. Journal of advanced Transportation 40, 23–46 (2006)
20. Thangiah, S.R., Shmygelska, O., Mennell, W.: An agent architecture for vehicle routing problems. In: Proceedings of the 2001 ACM symposium on Applied computing (SAC '01), New York, NY (USA), ACM Press, 517–521 (2001)
21. Kohout, R., Erol, K.: In-Time agent-based vehicle routing with a stochastic improvement heuristic. In: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence (AAAI'99/IAAI'99), Menlo Park, CA (USA), AAAI Press, 864–869 (1999)
22. Fischer, K., Muller, J., Pischel, M., Schier, D.: A model for cooperative transportation scheduling. In Lesser, V.R., Gasser, L., eds.: Proceedings of the First International Conference on Multiagent Systems (ICMAS'95), Menlo park, CA (USA), AAAI Press / MIT Press, 109–116 (1995)
23. Smith, R.G.: The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Trans. on Comp. C-29, 1104–1113 (1980)
24. Zeddini, B.: Modèles d'Auto-Organisation Multi-Agent pour le problème du transport à la demande. PhD thesis, University of le Havre, Le Havre (France) (2009) 164 pages. In french.
25. Dial, R.B.: Autonomous dial-a-ride transit introductory overview. Transportation Research Part C: Emerging Technologies 3, 261–275(15) (1995)
26. Zargayouna, M.: Modèle et langage de coordination pour les systèmes multi-agents ouverts. Application au problème du transport à la demande. PhD thesis, University of Paris-Dauphine, Paris (France) (2007) 165 pages. In french.
27. Gendreau, M., Guertin, F., Potvin, J.Y., Taillard, E.D.: Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science 33, 381–390 (1999)