

# Online Localized Resource Allocation Application to Urban Parking Management

Nesrine Bessghaier<sup>1,2</sup>, Mahdi Zargayouna<sup>2</sup>, Flavien BALBO<sup>1</sup>

<sup>1</sup> University Paris-Dauphine - LAMSADE.  
Paris, France.

<sup>2</sup> Université Paris-Est, IFSTTAR, GRETTIA.  
Noisy Le Grand, France.

{nesrine.bessghaier, balbo}@lamsade.dauphine.fr

hamza-mahdi.zargayouna@ifsttar.fr

**Abstract**—In this paper, we define the online localized resource allocation problem, especially relevant for modeling transportation applications. This model takes into account simultaneously the dynamic locations of the resources and the consumers, together with their online nondeterministic appearance. We instantiate the model with the urban parking management problem, and propose a multiagent system to solve it. The solution is based on a community of drivers who share their knowledge about available parking spots.

## I. INTRODUCTION

The problem of resource allocation is an important research field with several application domains. This kind of problem is faced by many applications such as telecommunications [1][2], manufacturing [3] and epidemic treatment [4]. A central parameter in the study of resource allocation problems is related to the nature of resources (discrete or not, reusable or not) and/or to the chosen allocation system. In [5] for instance, the authors consider resource allocation in a static environment (i.e. the number of the resources and the number of the consumers do not change).

In this paper, we are interested in the problem of resource allocation that takes into account both its temporal and spatial dimensions. Problems of this nature are faced by several transportation applications, which solving generally requires the simultaneous consideration of time and space. Application examples are the search of charging stations for electric cars [6], the parking spot management [7] and the sharing of vehicles (taxi, car, bike, etc.) [8]. On the one side, the time dimension has to be explicitly represented when the information about resources and/or consumers is not known at the beginning of the allocation. This kind of problem is generally modeled as an online resource allocation (ORA) problem [9]. On the other side, the space dimension has to be explicitly modeled when resources and consumers are situated and when the distance between them conditions the allocation. This kind of problem can be modeled as a localized resource allocation (LRA) problem [10].

The time dimension is considered in several applications of resource allocation. The satellite exploitation [11] is an example where image requests come over time. The ORA must take into account the satellite orientation and has to be

efficient and equitable: the satellite should not be underexploited and each member should get a return on investments that is proportional to its financial contribution. Real-time systems also take into account the time in the resource allocation. In [12] for instance, an architecture is developed supporting dynamic resource management. The objectives are to optimize and reconfigure system resources at runtime and to adapt to changing mission needs and resource status. Resource allocation in wireless cellular networks (e.g. in [1]) tackles the problem of wireless channels allocation in an online fashion because of the time-varying nature of the resources.

The space dimension is also considered in several applications. The domain that faces frequently this kind of allocation problem is telecommunication. In cellular networks for instance, one problem is to allocate the frequency to localized access points according to traffic demands [10]. In smart grids (e.g. [13]), the allocation problems are also generally LRA problems, since the objective of the allocation is to reduce the transactions price which depends strongly on the distance between resources and consumers.

Since we are particularly interested in transportation applications, we present in this paper a generic model for both ORA and LRA problems called OLRA (for Online Localized Resource Allocation) problem. Few works focus on this variant of the problem, because one of the dimensions (time or space) is often implicitly taken into account. For instance, the space dimension is implicit if the modeling considers the time to reach the resource. This partial modeling do not enable to take into consideration the constraint related to the space (respectively to the time). For instance, if the space is not modeled explicitly, we cannot express the fact that the consumer knowledge about a resource depends on the distance between them. This was the choice of the authors in [14], where they model the problem of resource allocation related to fleet management. They reduce the problem to a dynamic allocation problem by decomposing it into sub-problems since the resource requests are known in well-determined time and area. Our contribution is to introduce an explicit representation of the physical environment in the problem. This allows us to consider that consumers do not have complete knowledge about resources and their states.

The resources might also be volatile because they are in a shared space and can therefore be taken by any consumer. Finally, the resources can be uncontrolled because they are created and released in a nondeterministic way.

In this paper, we define the OLRA and we illustrate it with the application of urban parking management, for which we propose an agent-based approach.

This article is organized as follows. In section II we define the OLRA problem. Section III presents the application of urban parking, our multiagent solution and our experimental results. Finally, section IV concludes this paper.

## II. PROBLEM DEFINITION

### A. Description

Consider a set of resources and a set of consumers in different states. Both resources and consumers appear non-deterministically and can subsequently change their position at any moment. Each resource has a state and a set of properties. An allocation changes the state of the resource but its properties remain the same [14].

The problem that is considered in this paper involves the assignment of this set of resources to the consumers whose requests are not known in advance. Indeed, the offers (the resources) and the demands (the allocation requests) are situated in time. On the one side, a consumer starts looking for a resource at nondeterministic moments. These moments are not predefined and are discovered during the allocation process. On the other side, the resources are available starting from unknown moments and remain available during an unknown time window. The compliance of the resource with the consumer needs is conditioned with the spatial and temporal situation of the consumer and the resource. Like all resource allocation problems, the compliance of the resource with the consumer is also conditioned by his preferences, which concern the current state and the properties of the resource. The local objective of the consumers is to maximize their own utility while the global objective of the allocation system in this kind of problems is generally to minimize the total traveled distance and total travel time needed to access resources by the consumers.

### B. Problem formulation

In this section, we formulate the Online Localized Resource Allocation (OLRA) problem and we define its various components. That allows us notably to describe different variants and use cases for the general problem. An OLRA is a tuple:

$$OLRA = \langle \mathcal{R}, \mathcal{C}, G, D \rangle$$

where:

- $\mathcal{R} = \{r\}$  is the set of resources.
- $\mathcal{C} = \{c\}$  is the set of consumers.

- $G = \langle V, E \rangle$  is a directed graph, with  $V$  the set of nodes indexed from 1 to  $N$ , and  $E = \{e_{ij} | i, j \in V \text{ and } i \neq j\}$  the set of edges.
- $D = \{d_{ij} | i, j \in V \text{ and } i \neq j, d_{ij} \in \mathbb{R}_+\}$ ,  $d_{ij}$  is the distance between two successive nodes  $i$  and  $j$ .

Each node of the network can contain one or more resources of  $\mathcal{R}$  whose properties may differ. Resources can represent, for instance, vehicle seats, parking spots, places to recharge electric cars, etc. The distances between edges are fixed, while the travel times may vary according to the dynamics of the graph. If the graph represents a transportation network, the travel times would be impacted by traffic status and congestion.

The way the resources are discovered depends on the application but this is generally done in a progressive way, i.e. at the start of the allocation, we don't have the complete list of resources. They can for example be discovered thanks to sensors, from a database of resource providers, or from information collected about consumers. Consumers also appear and disappear in a non-deterministic way and can consume resources according to their preferences. The availability of a consumer or a resource at a certain moment is caught by the following function:

$$availability : (\mathcal{R} \cup \mathcal{C}) \times T \rightarrow \{0, 1\}$$

where  $T$  is the time horizon. This function returns 0 if the resource or the consumer is not yet localized and is equal to 1 until they disappear of the system.

The two following sets describe the different characteristics of the resources:

- $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$ : It represents all the possible properties of resources.
- $\Delta_p = \{d_1, d_2, \dots, d_n\}$ : I contains the description domains of these properties.

A property  $p_i \in \mathcal{P} : \mathcal{R} \rightarrow d_j$  is a function whose description domain  $d_j \in \Delta_p$  can be quantitative, qualitative or a finite set of data. Each resource is characterized by a set of properties whose corresponding values are given by this function:

$$\varrho : \mathcal{R} \rightarrow (\mathcal{P} \times \Delta_p)^q$$

where  $q$  is the number of properties that are defined for the resource. The set  $\mathcal{R}$  is by default heterogeneous, i.e. resources may be of different nature and be represented by completely different properties. If  $\mathcal{R}$  is homogeneous, resources are defined by the same  $q$  properties.

The following functions specify the dynamics of the allocation status. The three functions define the dynamic costs and the dynamic positions of resources and consumers.

- $\tau : V \times V \times T \rightarrow \mathbb{R}_+$ ,  $\tau(i, j, t)$  returns the travel time between  $i$  and  $j$  at time  $t$ .  $T$  is the time horizon.
- $\rho_r : \mathcal{R} \times T \rightarrow V$ ,  $\rho_r(r, t)$  returns the node where the resource  $r$  is located at  $t$ .

- $\rho_c : \mathcal{C} \times T \rightarrow V$ ,  $\rho_c(c, t)$  returns the node where the consumer  $c$  is located at  $t$ .

A resource or a consumer on the edge  $e_{ij}$  is considered to be positioned on  $i$  until he reaches  $j$ .

The interest of a consumer for a resource varies over time, either following an internal process or following his context. This context may include his current position or his final destination. The usefulness of a resource to a consumer is given by the following utility function:

$$\mu : \mathcal{C} \times \mathcal{R} \times T \rightarrow \mathbb{R}$$

$\mu(c, r, t)$  returns the utility of the resource  $r$  for the consumer  $c$  at time  $t$ . However, by construction, the consumption of a resource by a consumer is conditioned by their collocation. This can be verified with the indicator function as follows:

$$\mathbb{1}_F : F \rightarrow \{0, 1\}$$

where  $F = \{(c, r, t) \in \mathcal{C} \times \mathcal{R} \times T \mid \rho_c(c, t) = \rho_r(r, t)\}$ .

$\mathbb{1}_F(c, r, t)$  returns 1 if the consumer  $c$  has the same position than the resource  $c$  at time  $t$  and 0 otherwise. Therefore,  $F$  defines the set of space-time collocations of resources and consumers.

The function  $\gamma$  specifies when a customer actually consumes a resource:

$$\gamma : \mathcal{C} \times \mathcal{R} \times T \rightarrow \{0, 1\}$$

$\gamma(c, r, t)$  returns 1 if a consumer  $c$  takes the resource  $r$  at  $t$  and 0 if not. Since a consumer cannot take a resource if they are not at the same position at the same time, then that means that  $\gamma(c, r, t) = 1$  cannot be valid unless  $\mathbb{1}_F(c, r, t) = 1$ .

OLRA is not constrained to specific resource properties. It can model problems where the resources are shareable or not, and where consumers can consume several resources at the same time or not. The considered variant of the problem is specified by two parameters  $k$  and  $l$ . The considered problem has to comply with the two following constraints, which depend on  $k$  and  $l$ .

$$\sum_{c \in \mathcal{C}} \gamma(c, r, t) \leq k, \forall r \in \mathcal{R}, \forall t \in T \quad (1)$$

$$\sum_{r \in \mathcal{R}} \gamma(c, r, t) \leq l, \forall c \in \mathcal{C}, \forall t \in T \quad (2)$$

The constraint (1) specifies that the resources can be shareable and be taken simultaneously by at most  $k$  consumers ( $k \in \mathbb{N}$ ). If the resources are not shareable,  $k$  is equal to 1. If several resources are collocated with a consumer, the problem definition may allow him to consume them simultaneously (constraint (2)). The number of resources that can be taken simultaneously is a parameter  $l \in \mathbb{N}$ . Again, if this is not allowed,  $l$  is set to 1. The values of  $k$  and  $l$  are model parameters and enable to take into account

different problem variants and therefore different application types.

The quality of a resource allocation in OLRA is generally related to the distance and the travel time of consumers. Their cumulative positions throughout the allocation process are specified with the three following functions.

$$\pi : \mathcal{C} \rightarrow (\{1, \dots, N\} \times T)^n, n \in \mathbb{N}$$

$\pi$  defines the path of a consumer. Applied to a consumer  $c$ ,  $\pi$  returns the nodes that the consumer has visited while moving towards a resource, together with the times corresponding to his visits.  $\pi(c)[i, 1]$  allows to access the index of the  $i^{th}$  visited node, while  $\pi(c)[i, 2]$  allows to access the corresponding visit time.

For instance,  $\pi(c_1)[2, 1] = 10$  indicates that the second node visited by consumer  $c_1$  is  $v_{10}$ , while  $\pi(c_1)[2, 2] = t_3$  indicates that this visit occurs at time  $t_3$ .

$$\delta(c) = \sum_{i=0 \dots |\pi(c)|-1} d_{\pi(c)[i,1], \pi(c)[i+1,1]}$$

$\delta$  determines the total distance traveled by  $c$ . The term  $d_{\pi(c)[i,1], \pi(c)[i+1,1]}$  represents an element  $d_{xy}$  of the  $D$  matrix of distances, where  $x = \pi(c)[i, 1]$  and  $y = \pi(c)[i+1, 1]$  are, respectively, the  $i^{th}$  and the  $(i+1)^{th}$  node indices returned by  $\pi(c)$ .  $|\pi(c)|$  gives the total number of nodes visited by  $c$ .

For instance, if  $\pi(c_1) = [(5, t_1), (10, t_3)]$ , i.e. consumer  $c_1$  visits node  $v_5$  at time  $t_1$  then node  $v_{10}$  at time  $t_3$ ; if the distance  $d_{5,10} = 6$ , then  $\delta(c_1) = 6$ .

$$\varphi(c) = \pi(c)[|\pi(c)|, 2] - \pi(c)[1, 2]$$

$\varphi$  gives the total travel time of a consumer  $c$ . The expressions  $\pi(c)[|\pi(c)|, 2]$  and  $\pi(c)[1, 2]$  are the instants of visits, respectively, of the last node and the first node visited by  $c$ .

With the same above example with  $\pi(c_1) = [(5, t_1), (10, t_3)]$ ,  $\varphi(c) = t_3 - t_1$ .

The objective of the study of OLRA is generally to minimize the time and/or the distance spent in the search of resources. This social objective can be expressed as:

$$\min \sum_{c \in \mathcal{C}} [\alpha \delta(c) + \beta \varphi(c)]$$

where  $\alpha$  and  $\beta$  are positive numbers weighting the relative importance of time and space in the specific problem that is considered.

Besides, every consumer has the local objective of maximizing his own satisfaction by obtaining the resources that best satisfy his preferences and maximize his utility. This personal objective is defined as follow:

$$\max \sum_{r \in \mathcal{R}, t \in T} [\mu(c, r, t) \times \gamma(c, r, t)]$$

These two objective functions are used to compare the effectiveness of solutions proposed to solve a particular OLRA problem. A system might behave well w.r.t the local objectives of the consumers while the social objective is not optimized. Or it may exhibit good results for the social objective, while the individual objectives are sub-optimal. As usual in this kind of problems, there is a compromise between these two objectives that the proposed solutions to this problem have to find.

### III. APPLICATION: URBAN PARKING SPOTS MANAGEMENT

#### A. Urban parking spot search problem

Many applications can be modeled as an OLRA, especially in the transportation domain, such as fleet management [14], parking spot management [15], etc. In this paper, we present an application of urban parking management which is one of the important issues in the domain of transportation. Indeed, an important part of carbonic gas emissions is due to the traffic generated by drivers looking for parking spots. In fact, we can consider urban parking spot search problem as an OLRA where parking spots represent the localized resources and drivers are the resources consumers. The issue here is to adapt to a complex problem necessitating the consideration of a dynamic and open environment. A solution to this problem has to use minimal information on a shared, volatile and uncontrollable resource. It has also to be able to work without initial information and ensure to its consumers to have a resource availability information that is the most up-to-date possible.

#### B. Model instantiation

In this application, the set of resources  $\mathcal{R}$  is homogeneous, and composed of the parking spots. At the start of execution,  $\mathcal{R}$  might be equal to  $\emptyset$ , and is enriched by the discovery of parking spots by drivers. The set of consumers  $\mathcal{C}$  is composed of the drivers who belong to a community and desire to use the service. There are drivers outside the community who have the same rights to consume the spots. The drivers in the community inform the system when they take or release a parking spot while the others do not.

$G$  is the transport network of the considered town or region. The nodes of the network represent either a crossroad or a parking spot on an edge. The time horizon  $T$  is the considered timeframe for the execution, typically 24 hours. The *availability* function defines when a spot is discovered or when a driver joins the system.

The possible properties of a parking spot are its size in centimeters ( $d_{size} = \mathbb{R}_+$ ), the rating of the neighborhood ( $d_{rating} = \mathbb{N}$ ) and its safety ( $d_{safety} = \{0, 1\}$ ). For a parking spot  $r_1$ ,  $\varrho(r_1)$  can then be equal to  $\{(size, 200), (rating, 9), (safety, 1)\}$ . We could think of much more properties for the spots.

The drivers utility  $\mu$  can give a rating to a resource that combines conditions on the resource and on his context. For instance, a driver can be interested in the only safe resources, which size is longer than 2 meters and that are not further from his final destination than 500 meters.

In this application, the spots can be taken by anyone, but not more than one driver can take a spot. As a consequence, the parameter  $k$  is equal to 1. In addition, not more than one spot can be collocated with a driver, and one driver cannot take more than one spot at the same time. The parameter  $l$  is then also equal to 1.

In the following, we propose a solution to the problem of urban parking. The solution is based on a distributed architecture. In section III-D, we compare this proposal with an approach that centralizes the processing, w.r.t the objective functions defined in the model.

#### C. Agent-based solving

To solve this problem, we propose an agent-based transport information system that helps to find parking spots in an urban agglomeration. This approach is particularly relevant for the management of parking spots, since the problem is to take into account human behaviors that interact in a complex, dynamic and open environment. Our agent-based approach is totally decentralized and we employ an inter-vehicular communication (V2V) to allow vehicles to receive and broadcast information to the other vehicles of the same community. The choice of a distributed approach allows us a.o. to minimize the infrastructure to implement this solution and to limit investment.

1) *Agents Model*: The proposed system for the search of spots in an urban area is modeled by a type of agent designated by *assistant* agent who represent a driver. The internal architecture of the assistant agent is composed of three modules: a *Communication* module, an *Itinerary* module and a *Decision* module. The first module enables the agent to communicate with his neighbors, which have to belong to the community as well. This communication is based on messages and allows to exchange information about the availability of parking spots. The itinerary module ensures the calculation of the route to a particular parking spot starting from the driver's current position, but also monitors his movement. Finally, the decision module takes care of the decision making. This module proposes a parking spot to the driver. The latter must meet the criteria specified by the driver, which may concern for instance its distance, the time since its release, or the safety of its location. In other words, the decision module implements the utility function  $\mu$  defined in the model.

In addition to proposing parking spots, the decision module manages a memory containing information related to the spots. This knowledge evolves over time with information acquired through the exchange of messages with different

assistant agents and to the perception of the agent. This memory is composed of two disjoint lists.

- FS list (for Free Spots) contains a set of pairs  $\{ \langle spot, time \rangle \}$ , each pair refers to a specific spot: its geographic position and the moment since which it was released.
- OS List (for Occupied Spots) contains the spots that were in FS but which turned out to be occupied with the moment since which this information was known.

Both lists are exchanged by the assistant agents and are updated gradually by the knowledge of each one. The combined use of the two lists provides a dynamic update of the system information. Indeed, one consequence of the volatility of information regarding the availability of spots is illustrated when an agent chooses a spot on its FS list - supposed to be free - but, once there, it finds it occupied. In this case, the FS lists contain incorrect information about this spot. That's why, the OS list enables agents to filter the information received and to have the best information possible.

To allow an update of the lists without specific information, the decision module of each assistant agent filters outdated information after a time  $\theta$ , i.e. the spots in FS and OS with an associated time that is inferior to the current time minus  $\theta$ . This parameter takes into account the network activity. Thus, a low value reflects a high volatility as the case may be in rush hours in downtown, while a high value keeps a longer sharing of information and reflects, for instance, the lower volatility in a residential area.

The density of the network can generate a large number of messages. However, the communications take place very locally between vehicles and the total number of messages by agent is less important than in a centralized architecture. In the experiments section, we measure the average number of exchanged messages per agent in our system.

```

 $FS_A = \{ \dots \langle sp_{i_1}, t_{i_1} \rangle, \dots \}$ 
 $FS_B = \{ \dots \langle sp_{i_2}, t_{i_2} \rangle, \dots \}$ 
 $OS_A = \{ \dots \langle sp_{i_3}, t_{i_3} \rangle, \dots \}$ 
 $OS_B = \{ \dots \langle sp_{i_4}, t_{i_4} \rangle, \dots \}$ 
for all  $\langle sp_i, t_i \rangle \in FS_B$  do
  if  $(sp_i \in OS_A)$  then
    if  $(t_i^B \geq t_k^A)$  then
      delete( $sp_i, OS_A$ )
      add( $sp_i, t_i^B, FS_A$ )
    end if
  else
    if  $(sp_i \in FS_A)$  then
      if  $(t_k^A < t_i^B)$  then
        update( $(sp_i, t_i^B, FS_A)$ )
      end if
    else
      add( $sp_i, t_i^B, FS_A$ )
    end if

```

```

end if
end for
for all  $\langle sp_i, t_i \rangle \in OS_B$  do
  if  $(sp_i \in OS_A)$  then
    if  $(t_j^A < t_i^B)$  then
      update( $(sp_i, t_i^B, OS_A)$ )
    end if
  else
    if  $(sp_i \in FS_A)$  then
      if  $(t_j^A < t_i^B)$  then
        delete( $sp_i, FS_A$ )
        add( $sp_i, t_i^B, OS_A$ )
      end if
    end if
  else
    add( $sp_i, t_i^B, OS_A$ )
  end if
end for

```

2) "*Coopetition*" model: To manage parking in an urban environment, our MAS is based on the coopetition [16] of agents to share information regarding the availability of spots. Indeed, despite their competition, the agents of the community cooperate to achieve their own aim and that of their community. This coopetition uses two types of broadcast. The first type is related to their cooperative behavior and concerns all the information that the agent has when not looking for a parking spot. The second type of broadcast is the consequence of their competitive behavior and is illustrated by the fact that an agent only broadcasts the information that does not interest him.

The messages exchanged between assistant agents from the same community include their lists (FS and OS) which contain, respectively, the spots that are possibly free and those probably taken. The update of the lists is illustrated in Figure 1. The communication module of the assistant agent extracts the lists  $FS_B$  and  $OS_B$  from each received message and forwards it to the decision module. This corresponds respectively to edges (1) and (2) in Figure 1. The decision module updates both lists by aggregating the received  $FS_B$  and  $OS_B$  lists with its own  $FS_A$  and  $OS_A$  lists.

The idea is to browse each received list ( $FS_B$  and  $OS_B$ ) and update the local lists ( $FS_A$  and  $OS_A$ ) with the date associated with the spots. If there are two conflicting informations, then the newest one is kept, since the last driver who has visited this spot has the information that is more probably correct about its availability. This update is made by using the algorithm below. After the update of the two lists  $FS_A$  and  $OS_A$ , the decision module hand them to the communication module, which is responsible for the transmission of the information to neighbors in the community. These steps are followed whether the driver is looking for a spot or not.

When the driver  $dr$  is looking for a parking spot, he

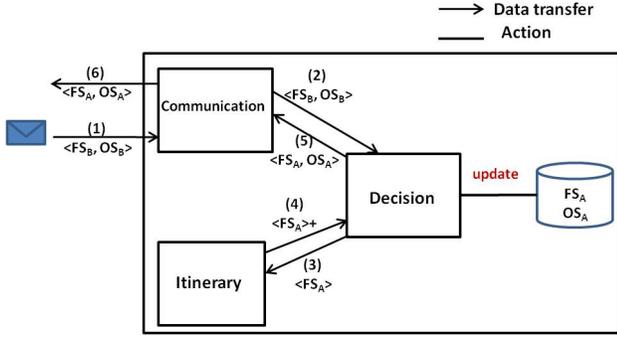


Figure 1. Assistant agent internal data flows

requests help from his embedded system. The decision module of his assistant agent updates  $FS$  and  $OS$  with the received messages, then sends  $FS$  to the itinerary module. This corresponds to the edge (3) in Figure 1. The itinerary module computes the routes for each spot on this list and forwards the result back to the decision module. Based on the utility function  $\mu$ , the decision module proposes a spot that best meets the needs of the driver. In order to assess our proposal with an objective decision criteria, we have used an utility function that takes into account the time to reach the resources:

$$\mu(c, r, t) = \frac{1}{\tau(\rho(r, t), \rho(c, t), t)}$$

Then, the decision modules deletes the information corresponding to the proposed spot from  $FS$ . Finally, it sends the rest of the list together with  $OS$  to the communication module which takes care of their distribution to the neighbors. The removal of the information about this spot reduces its spread within the community. Thus, the assistant agent increases the driver's chances of finding the spot free. In addition, during the movement of the driver to the chosen spot, the assistant agent can suggest an alternative spot that would better meet his needs.

#### D. Experiments

1) *Configuration*: The objective of the community management of parking spots is to allow the sharing of information about available spots in a decentralized manner. For the validation, it is necessary to compare the effectiveness of the process of finding spots of drivers who use the system with drivers who don't. We have chosen the proximity to the current position as the decision criterion to choose a spot. This test has the advantage not to rely on qualitative information such as the safety of the location and limits the effects of bias. Besides, the use of an absolute temporal criterion solely like choosing the last released spot would have the consequence to direct all searching agents to the same spots.

The parameters that are selected for the simulation are the following. First, the number of agents within and outside the

community. This parameter impacts the controllability of the spots. The more we have drivers outside the community, the more unexpected "disappearance" of spots we might encounter. Then, the time spent by an agent on a spot (OT, for occupation time). The more drivers spend time occupying a spot, the less spots are available for the other drivers. The third parameter is the lifetime of the information on the availability of a spot ( $\theta$ ). A  $\theta$  that is too big would lead to the exchange of obsolete information and therefore unnecessarily increase the bandwidth use. A  $\theta$  that is too small would lead to the deletion of still relevant information about spots and make drivers loose parking opportunities.

To evaluate the different scenarios, we choose the two following criteria. The first is the average time spent to find a spot per agent (ST). This criterion verifies that a driver earns to use the system rather than finding a spot on his own, and quantifies this gain. In other words, we set the coefficient  $\alpha$  in the objective function of OLRA to 0 and the value of  $\beta$  to 1. The second criterion is the success rate (SR, or effective use rate of the system) by the agents of the community. It represents the ratio between the number of drivers in the community who have found a parking spot thanks to the system by the total number of drivers in the community. All time variables are expressed in number of execution cycles.

We have used MadKit [17] as a simulation platform. For each simulation we have performed 15 executions, each one running on 50 cycles. Results given in the next section contain the means of these different executions. The network of our simulation contains 17 edges and it takes 10 cycles to traverse an edge. At the start of execution, we set the agents randomly on the network in which we set a fixed number of spots in each edge. Some of them are stopped and the others are traveling. Each edge is bidirectional, and an agent can turn around if needed.

2) *Results*: To demonstrate the effectiveness and utility of our proposal, we have conducted many series of simulations. In the first series, we vary the number of agents in the community (NbA). This allows us to verify the impact of the system use on search time. In the second series, we study the impact of the rareness of spots on the success rate (SR). Finally, we compare the average number of exchanged messages per agent in our system and in a centralized approach.

In the graph shown in Figure 2, we represent the average time spent to find a spot for the agents in and outside the community. The total number of agents in the simulation is 300, i.e. the number of agents outside the community is the complement to 300 of those in the community. The abscissa axis gives the number of agents that are taken into account. For instance, the value 100 means that 100 agents into the community spend on average 13.62 cycles to find a free spot whereas 100 agents outside community spend 20.57 cycles on average.

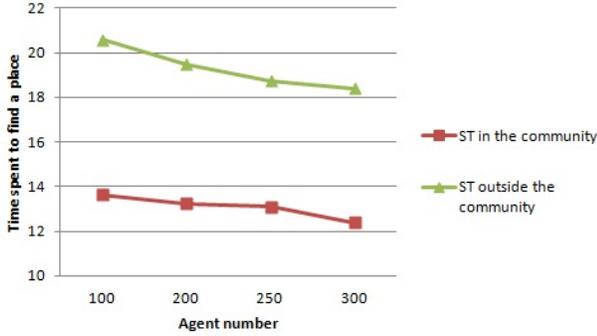


Figure 2. Profit of the system

We can notice that, the more agents in the community we have, the less time they spend searching spots. This result is due to the fact that spots availability information is better propagated in the community, when the number of its members is important. Therefore, vehicles using the system spend less time to find a parking spot. Moreover, we can also note that the average time to find a spot (ST) for an agent of the community is much lower than that of an agent outside the community. The difference varies from one to seven cycles when all agents are in the community. For example, if there are 100 agents in the community, the average ST is equal to 13.62 cycles, whereas the average ST of an agent outside the community (i.e. 200 agents) is 19.48 cycles. According to these results, we can conclude that our proposal is useful and effective, especially when the community size is large enough.

In the next series of simulations, we fixed the number of spots in the network and varied the occupation time of a spot by a vehicle (OT). This way, we increase the spots rareness since vehicles monopolize spots more time and leaves less possibilities for others to find a free spot. The Figure 3 illustrates the variation of the success rate according to the rareness of spots. For example the SR is 33.82 % when the OT is equal to 2 cycles. However this rate increases to 39,85 % when OT is 4 cycles. Then the SR stagnates since the number of spots becomes really limited. Indeed, when a driver perceives several free spots, he does not really need help to find one. However, if they are rare the proposed system turns out to be very useful. These results show that the less spots we have, the more the system is useful, until a certain threshold where the number of resources becomes too limited.

In Figure 4, we report the number of messages handled by each agent in each cycle in our proposal, which we compare with a centralized solution. In the centralized approach, when the driver leaves a spot, two messages are exchanged with a central agent for each parking spot search (request and response) and a message informing that the chosen spot is taken. This agent is unique, which limits the total number

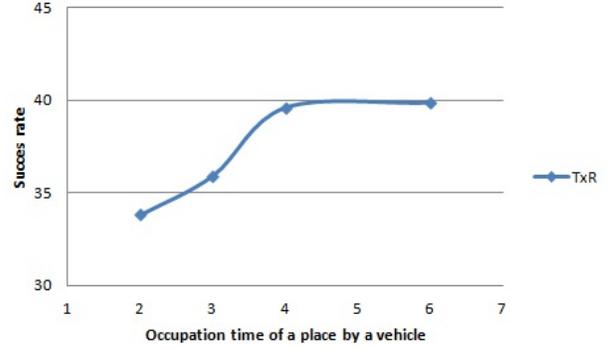


Figure 3. Impact of spot rareness

of messages but, he forms a communication bottleneck. Our proposal is based on local communication. The number of messages exchanged depends on the number of vehicles in the community with the same communication range and does not depend on the total number of the community members. Then, the number of messages may differ from one area to another in the network, depending on its activity. So we have not to compare the total number of messages sent in the two approaches.

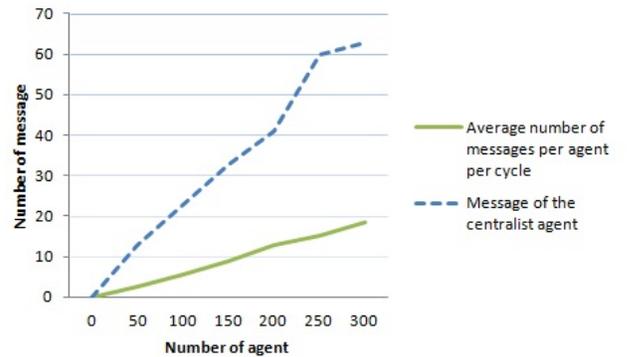


Figure 4. Comparison of the number of messages per agent

#### IV. CONCLUSION

In this paper, we propose a modeling for the resources allocation problem taking into account simultaneously the location and the moment when the resources are available. Our modeling is well adapted to the transportation domain where many applications are characterized by the difficulty to take into account their space-time dimension. Our modeling is able to take into account several kinds of constraints: space, the resources or consumers are static or not; time, the availability of the resources and needs of consumers change according to time; space-time, the resource and consumer have to be at the same location.

We have used our modeling to specify the management of parking spots in an urban area and proposed a multiagent solution. The system is based on a community of drivers that interact to keep up-to-date information regarding the availability of parking spots. Communication between agents is supported by an inter-vehicular network with a radius of restricted broadcast, ensuring the consideration of local information. Our system works without prior information on the places and no central storage of information. We have focused our validation on the average search time and showed a decrease regardless of the density of the vehicular network.

We are considering different perspectives to this work. Among them, we want to improve the decision function of the agents in order to take into account other criteria than the time and space and to evaluate how the aggregation of their preference impacts the quality of the solutions.

#### REFERENCES

- [1] R. Madan, S. P. Boyd, and S. Lall, "Fast algorithms for resource allocation in wireless cellular networks," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 973–984, Jun. 2010.
- [2] F. Meshkati, H. V. Poor, S. C. Schwartz, and R. V. Balan, "Energy-efficient resource allocation in wireless networks with quality-of-service constraints," *IEEE Transactions on Communications*, vol. 57, pp. 3406–3414, 2009.
- [3] L. Münch, M. Stehli, R. Schulz, and M. S. Rol, "An agent-based architecture for solving dynamic resource allocation problems in manufacturing," in *Proceedings of the 14th European Simulation Symposium. Simulation in Industry*.
- [4] G. S. Zaric and M. L. Brandeau, "Dynamic resource allocation for epidemic control in multiple populations," *Ima Journal Of Mathematics Applied In Medicine And Biology*, vol. 19, no. 4, pp. 235–255, 2002.
- [5] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar, and P. Sousa, "Issues in multiagent resource allocation," *Informatica*, vol. 30, pp. 3–31, 2006.
- [6] S. Acha, T. C. Green, and N. Shah, "Optimal charging strategies of electric vehicles in the uk power market," *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pp. 1 – 8, 2011.
- [7] X. Bo, A. Ouksel, and O. Wolfson, "Opportunistic resource exchange in intervehicle ad-hoc networks," in *IEEE International Conference on Mobile Data Management.Proceedings.2004*. IEEE, 2004, pp. 4–12.
- [8] R. Katzew, "Car sharing: A new approach to urban transportation problems," *Analyses of Social Issues and Public Policy*, vol. 3, no. 1, pp. 65–86, 2003.
- [9] G. Tesauro, "Online resource allocation using decompositional reinforcement learning," in *Proceedings of the 20th national conference on Artificial intelligence*, vol. 2. AAAI Press, 2005, pp. 886–891.
- [10] B. Golkar and E. Sousa, "Adaptive localized resource allocation with access point coordination in cellular networks," in *IEEE International Conference on Communications (ICC)*, 2011.
- [11] A. Narula-Tam, T. Macdonald, E. Modiano, and L. Servi, "A dynamic resource allocation strategy for satellite communications," in *In proceeding of: Military Communications Conference*, vol. 3. IEEE, 2004, pp. 1415 – 1421.
- [12] P. Lardieri, J. Balasubramanian, D. C. Schmidt, G. Thaker, A. Gokhale, and T. Damiano, "A multi-layered resource management framework for dynamic resource management in enterprise dre systems," *Journal of Systems and Software*, vol. 80, pp. 984–996, July 2007.
- [13] B. HomChaudhuri and M. Kumar, "Market based allocation of power in smart grid," in *American Control Conference (ACC),2011*. IEEE, 2011, pp. 3251–3256.
- [14] H. Topaloglu and W. B. Powell, "A Distributed Decision-Making Structure for Dynamic Resource Allocation Using Nonlinear Functional Approximations," *Operations Research*, vol. 53, March 2005.
- [15] T. Delot, N. Cenerario, S. Ilarri, and S. Lecomte, "A cooperative reservation protocol for parking spaces in vehicular ad hoc networks," in *Proceedings of the 6th International Conference on Mobile Technology, Application &#38; Systems, ser. Mobility '09*. New York, NY, USA: ACM, 2009, pp. 31–38.
- [16] Y. Luo, "A coopetition perspective of global competition," *Journal of World Business*, vol. 42, no. 2, pp. 129 – 144, 2007.
- [17] O. Gutknecht and J. Ferber, "The madkit agent platform architecture," in *Agents Workshop on Infrastructure for Multi-Agent Systems*, ser. Lecture Notes in Computer Science, T. Wagner and O. F. Rana, Eds., vol. 1887. Springer, 2001, pp. 48–55.