A Generic Agent Model Towards Comparing Resource Allocation Approaches to On-demand Transport with Autonomous Vehicles

Alaa Daoud, Flavien Balbo, Paolo Gianessi, Gauthier Picard

A Generic Agent Model Towards Comparing Resource Allocation Approaches to On-demand Transport with Autonomous Vehicles

Alaa Daoud, Flavien Balbo, Paolo Gianessi
Mines Saint-Étienne, CNRS, UMR 6158, LIMOS
Institut Henri Fayol, Saint-Étienne, France
{alaa.daoud,flavien.balbo,paolo.gianessi}@emse.fr

Gauthier Picard
ONERA/DTIS
Université de Toulouse
gauthier.picard@onera.fr

ABSTRACT

Allocation problems are major issues in managing On-Demand Transport (ODT) systems. They have been studied for decades, and a variety of solutions were proposed. The approaches to these problems can be classified as centralized and decentralized. Each has its pros and cons in practice. In this work, we aim to provide a generic model for the problem of online on-demand transport (ODT) with autonomous vehicles and a multi-agent model dedicated to resource allocation and scheduling in vehicle fleets. This generic model supports the processing of different allocation mechanisms, and considers autonomous vehicles that communicate via peer-to-peer radio channels to meet passenger requirements and satisfy trip requests in an online ODT system. We validate this model's genericity by applying several allocation mechanisms (mathematical programming, greedy heuristic, distributed constraint optimization, and auctions) and compare their performance on synthetic scenarios in a real-world city map.

KEYWORDS

Multi-Agent Systems, Resource Allocation, Auctions, Distributed Optimization, On Demand Transport

ACM Reference Format:
Alaa Daoud, Flavien Balbo, Paolo Gianessi and Gauthier Picard. 2021. A Generic Agent Model Towards Comparing Resource Allocation Approaches to On-demand Transport with Autonomous Vehicles. In *OptLearnMAS-21 The 12th Workshop on Optimization and Learning in Multiagent Systems, at AAMAS 2021, (Online), May 4, 2021*, IFAAMAS, 8 pages.

1 INTRODUCTION

A driver-less (a.k.a. automated or self-driving) vehicle can be defined as a computer-controlled vehicle that drives itself and perform trips safely. Autonomous vehicles (AVs) can be seen as a sort of driver-less vehicles with extra capabilities, e.g., chose their route upon the traffic situation, coordinate with other vehicles to share information and improve the traffic quality. Significant issues in managing On-Demand Transport (ODT) systems are allocation problems consisting of finding feasible and reasonable allocations of requests to vehicles. Being autonomous, the vehicles within a taxi fleet can be responsible for their choice of allocation to requests (making *decentralized decisions*), or follow the schedules that are *centrally decided by a dispatcher*. In practice, the feasibility and efficiency of the choice to centralize/decentralize the solution depend on the problem complexity, its constraints and the environment's dynamics. In this work, (i) we propose AV-OLRA, a generic model for online resource allocation problem with autonomous vehicles. This model defines the hypothesis of the problem (components, constraints) and the indicators to evaluate the different allocation strategies; (ii) we propose a generic multi-agent model for ODT problem solution methods, where autonomous vehicles (agents) communicate with their neighbors via peer-to-peer communication through connected sets; (iii) we classify the different allocation methods based on the agents' coordination behavior (Selfish, Cooperative, Competitive, and Dispatching); (iv) we experimentally evaluate and compare different solution methods (centralized, greedy, auction-based, and distributed constraint optimization). The paper is structured as follows. Section 2 presents some related works to multi-agent resource allocation and ODT. Based on this literature review, we expound on the AV-OLRA problem in Section 3, and a generic multi-agent model to address it in Section 4. Section 5 discusses in more detail the different coordination mechanisms investigated in this study, which are then experimentally evaluated in Section 6. Finally, the paper concludes with some perspectives in Section 7.

2 MULTI-AGENT RESOURCE ALLOCATION AND ODT

In recent years, the number of articles devoted to applying agent-based technologies to transport and communications engineering has increased significantly. Existing simulations and models of ODT were described by [18]. To become demand-responsive, a transport system's model has to include considering people with their behavior and their interaction with a complex transport environment. A survey conducted in [1] on agent-based modeling and simulation of transport, and stated that agent-based approaches are well suited to traffic and transport management, given the geographic, functional and temporal distribution of data and control, as well as frequent and flexible interactions between participants and their environment. Multi-Agent Resource Allocation (MARA) vision is relevant for a wide range of application domains [3]. It has been applied in areas of network routing, public transport, e-commerce [2], social activities [8], and scheduling [14]. Centralization of the allocation process with an automatic dispatcher is still quite common in multi-agent approaches [5, 15, 20]. On the other hand and to achieve real time planning for ODT services, several decentralized models were proposed [9, 19]. A theoretical transport system model is developed by [11] to study the cooperation behavior of vehicles, with a global perspective; the best efficiency of cooperating vehicles should be to share knowledge in a flexible public transport. On the contrary, in the absence of communication between agents, [22] investigates

the applicability of genetic programming for developing decentralized MAS that solve ODT problems. They concluded that long-term planning is not beneficial in such settings because of the rapidly changing dynamics; thus, agents should consider only one request in advance.

One of the main issues for using MARA approaches to solve ODT related problems is the communication bottleneck. [10] proposes that each agent has a limited space of planning area and does not need to communicate with all others. Not far from the problem at hand, Monier et al. introduced the multi-agent exploration problem under limited range communication constraints to provide a performance evaluation of distributed constraint satisfaction problem (DCSP) algorithms, namely ABT, AWC, and DBS [16].

So far, the solutions for resource allocation problems in ODT systems' dynamic environments must challenge vehicles' schedules in real-time. This challenge makes the achievement of an optimal solution in practice an elusive goal. However, designing improving approaches for feasible solutions is a suitable alternative to tackle the dynamic aspect issues; this requires taking the communication aspect into account and providing robust and efficient communication and coordination mechanisms[9–11].

Similarly, [24] proposed a generic, i.e., independent of the solution/strategy, modeling of the Online Localized Resource Allocation (OLRA) problem, and a multi-agent system to solve the management of urban parking problem. The solution relies on a driver community that shares its local knowledge of the availability of parking spaces. This work relies on this later model to propose our specific model for the on-demand resource allocation problem in autonomous vehicle fleets.

3 AV-OLRA PROBLEM

In this paper, we define the AV-OLRA problem, a specialization of OLRA for online resource allocation with autonomous vehicles, and an extension with the communication and additional time constraints modeling.

We thus formulate the AV-OLRA problem as follows:

$$\text{AV-OLRA} := \left(\mathcal{R}, \mathcal{V}, \mathcal{G}, \mathcal{T}\right) \tag{1}$$

$$\mathcal{R} = \{r_i | i \in \mathbb{N}\} \tag{2}$$

$$\mathcal{V} = \{v_i | i \in \mathbb{N}\} \tag{3}$$

$$\mathcal{G} = \left(\mathcal{N}, \mathcal{E}, \omega\right) \tag{4}$$

$$\mathcal{T} := \{t_0, t_1, \ldots, t_{end}\} \tag{5}$$

where $\mathcal{R}$ defines a dynamic set of resources that occur to be available for a specific time window at the time of execution, representing passengers' requests; the set of consumers is represented by a fleet $\mathcal{V}$ of m autonomous vehicles that are mobile and can only communicate within a limited range; The spatial environment of the problem is the urban network defined by the directed graph $\mathcal{G}$, with $\mathcal{N}$ the set of nodes, and $\mathcal{E}$ the set of edges, $e_{ij} \in \mathcal{E}$ is the edge between the nodes i and j, ω is a valuation function that ω associates each edge $e \in \mathcal{E}$ with the value ω_e based on a temporal distance measure (e.g., average driving time in minutes), which will be used to calculate the operational costs of vehicle trips; In addition to the temporal distance measure, we use another notion of distance in this spatial environment. The function $distance : \mathcal{N} \times \mathcal{N} \rightarrow \mathbb{R}+$ calculates the euclidean distance between two locations

Definition 1. *An autonomous vehicle $v \in \mathcal{V}$ is characterized by its capacity $c : \mathcal{V} \rightarrow \mathbb{N}$, its driving cost per traveled distance $cpd : \mathcal{V} \rightarrow \mathbb{R}+$ and a limited communication range $rng : \mathcal{V} \rightarrow \mathbb{R}+$ that never changes upon time*

$$v := \left(c_v, cpd_v, rng_v\right)$$

while it has also a set of time-dependant properties which are its current location $\texttt{loc} : \mathcal{V} \times \mathcal{T} \rightarrow \mathcal{N} \cup \mathcal{E}$, its current destination, $\texttt{dest} : \mathcal{V} \times \mathcal{T} \rightarrow \mathcal{N}$ and the number of currently available seats $\texttt{seats} : \mathcal{V} \times \mathcal{T} \rightarrow \mathcal{N}$.

Connectivity between two components in the system is achieved if the distance between them is less than or equals their communication range. However, as the vehicles' communication range is limited, and to maximize their connectivity, two vehicles can be connected by transitivity. The binary function $\texttt{d\_ctd} : \mathcal{V} \times \mathcal{V} \times \mathcal{T} \rightarrow \{0, 1\}$ defines if two vehicles are connected directly to each other.

$$d\_ctd(i, j, t) = \begin{cases} 1, & \text{if } distance\left(\texttt{loc}_i^t, \texttt{loc}_j^t\right) \leq r \\ & : r = min(rng_i, rng_i) \\ 0, & \text{otherwise} \end{cases}$$

Two vehicles v_i and v_j are connected by transitivity if there exists v_k that is connected directly or by transitivity to both of them. The binary function $\texttt{ctd} : \mathcal{V} \times \mathcal{V} \times \mathcal{T} \rightarrow \{0, 1\}$ generalizes the $\texttt{d\_ctd}$ with the transitive connectivity.

$$ctd(i, j, t) = \begin{cases} 1, & \text{if } d\_ctd(i, j, t) \\ & \text{or } \exists k : ctd(i, k, t) \& ctd(k, j, t) \\ 0, & \text{otherwise} \end{cases}$$

This leads to the definition of connected sets:

Definition 2. *A connected set is a set of entities that are connected directly or by transitivity.*

$$CS : \mathcal{V} \times \mathcal{T} \rightarrow 2^V$$
$$CS(i, t) = \{j \in \mathcal{V} | ctd(i, j, t)\}$$

The connected sets are dynamic entities; they are created, split, merged at run-time based on the vehicles' movement (see Figure 1). Thus, based on the previous definitions, a vehicle may communicate at time t only with the members of its connected set by directed or broadcast messages. The limited communication range implicitly partitions the fleet into multiple connected sets.

Definition 3. *A solution for AV-OLRA is defined for each connected set as an aggregation of the allocations of all consumers in this set, avoiding all conflicts that could happen.*

This definition implies that a solution to an AV-OLRA problem defined for the consumers and the resources may be sub-optimal because several vehicles consider the same resources or because the optimal solution is not the union of the optimal solutions in each connected set. Also, any solution is time-dependent according to the online dimension of the problem.

The communication range value depends only on the used communication technology standard, considering it in our model adds

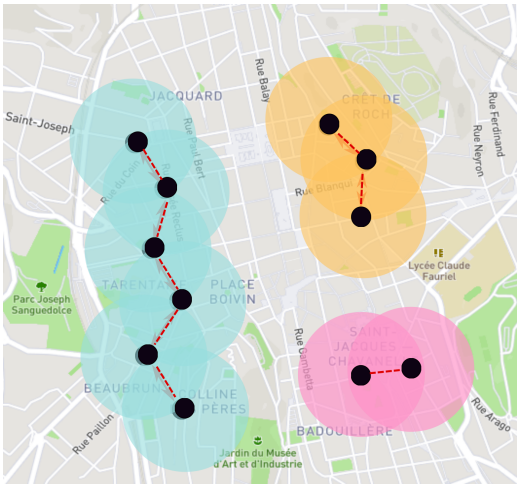

Figure 1: Dynamic composition of connected sets

another dimension of genericity. The lower the communication range is, the more connected sets exist; this means that for short-range communication, even with centralized dispatching, the solution computation is decentralized to several dispatchers. A fleet with a long enough communication range could result in one city-scale connected set with global knowledge sharing. a centralized dispatching mechanism then leads to one central dispatcher to calculate the global solution.

The quality of allocation is characterized by functional and technical indicators whose computation is independent of the allocation mechanisms but helps compare their feasibility and quality. The functional indicators are measures of optimality of the allocation process defined by its objective function, while the technical indicators are used to assess the feasibility and applicability of the allocation process and to predict its costs in different settings. In this paper, we characterize the quality of AV-OLRA solution in ODT scenarios by the following indicators. **Quality** is the *percentage of satisfied (consumed) requests* from all known requests. Therefore, this indicator points to the Quality of Service (QoS) level. **Utility** is the *total utility* of vehicle schedules, derived from the distances of successful trips (driven with a passenger on board from source to destination), which defines the profit for the company. **Cost** is the *operational cost*, derived from the total driven distances of the vehicles. The relation between **Utility** and **Cost** indicators defines the Quality of Business (QoB). **MsgCount** is the *total number of messages* exchanged during the allocation process. **MsgSize** is the *average size of messages* exchanged during the allocation. These two last communication indicators can be used to estimate the technical cost of the solution and predict if such a solution is applicable in terms of communication, i.e., if it could cause critical communication bottlenecks.

4 A MULTI-AGENT APPROACH TO AV-OLRA

In this section, we describe our multi-agent model to AV-OLRA problem. The environment of the AV-OLRA problem represents the

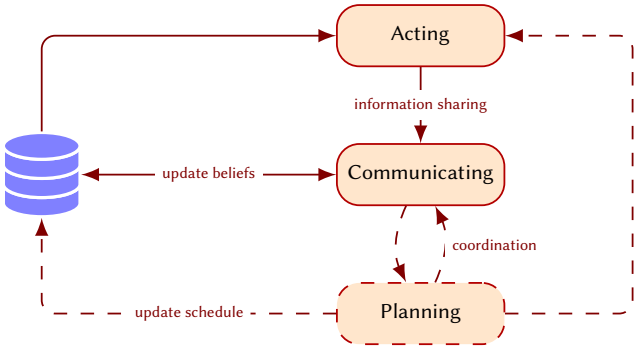

Figure 2: Generic vehicle behavior

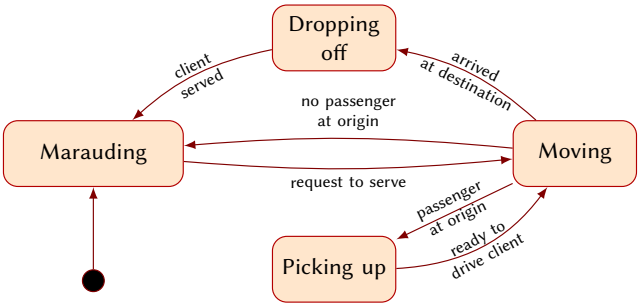

Figure 3: Acting sub-behavior

urban infrastructure topology $\mathcal{G}$ and the communication model of agents as described by Definition 2.

There is only one type of agents in our model. An autonomous vehicle agent (AV) is associated with each vehicle in the system. We can distinguish three different sub-behaviors (*acting*, *communicating*, and *planning*). As we model AV-OLRA in discrete time space, the time horizon is defined as set of ticks. At each time tick every agent performs the following actions as shown in Figure 2: (1) read the received messages and update the context (communicating sub-behavior) (2) choose the locations to visit (planning sub-behavior) (3) act by performing a driving action (acting sub-behavior) (4) broadcast its context information (communicating sub-behavior)

4.1 AV Acting Sub-behavior

Based on the existence of passengers on-board, the vehicle location, and its knowledge about upcoming requests, the AV agent can be in one of the following states, as shown in Figure 3: ***Marauding*** (the vehicle has no passenger on-board and is looking around for its next destination), ***Moving*** (the vehicle has a destination and is moving in the urban topology towards this destination), ***Picking up*** (the vehicle is located at the origin location of the passenger p's request to perform the `pick_up`(p) action and then start moving again), ***Dropping off*** (the vehicle is located at the destination location of the passenger p's request to perform the `drop_off`(p) action and then look for a new destination). Transitions between these states are based on the following set of events shown in Figure 3. ***request to serve*** (vehicle v has no passenger on board and a request r is

selected to be served, this implies the vehicle has a new destination to go to), ***passenger at origin*** (vehicle v arrived at request r's origin location o_r, passengers of r are present in o_r so v can start to pick up), ***ready to drive*** (the passengers of a request r got on vehicle v that is now ready to drive to the destination location d_r), ***arrived at destination*** (vehicle v has passengers on board belonging to a request r and arrived at the destination location d_r), ***client served*** (vehicle v is free to choose a request to serve, after delivering passengers at their destination, and may look for the next request to serve), ***no passenger at origin*** (the vehicle v has no request on board, arrived at the origin o_r of a request r, but the passengers of r are absent. the vehicle becomes free again).

4.2 AV Communicating Sub-Behavior

As communicating agents, AVs have a communication behavior with other surrounding entities; they can join/leave connected sets, broadcast, send, and receive messages.

- `join`(c): agent joins a connected set c as a result of being in the communication range of one of its members,
- `leave`(c): agent leaves its connected set c as a result of being disconnected from all its members,
- `send`(m, a): agent sends a message m to another agent a in condition they are in the same connected set,
- `receive`(m): agent receives a message m from another agent in its connected set (once received and read, the message is stored in the agent's belief base),
- `broadcast`(m) similar to `send`(m, a) but here the agent doesn't specify the receiving agent, instead it broadcasts the message to the whole connected set members.

4.3 AV Planning Sub-Behaviors

AVs' planning behavior depends on the chosen coordination mechanism (centralized/decentralized, cooperative/competitive). Figure 4 illustrates the abstract and generic planning behavior of AVs. For updating its schedule, an AV continuously looks for planning options. If any option is found, the AV selects one and depending on the coordination mechanism it communicates or not its decision with its neighbors. The neighborhood reaches an agreement or disagreement, depending on the coordination mechanism and the selected option. On agreement, the AV updates its schedule and look for the next option and until no option is available. The nature of planning options is also dependent on the coordination mechanism.

In Section 5, we present in details a set of different coordination mechanisms that we use to validate our model, including some state-of-the-art cooperative mechanisms (DCOP algorithms) and an auction-based mechanism for agent coordination behavior, designed to meet the requirements of the AV-OLRA problem in order to obtain a solution using lightweight calculations, dynamic and continuously subject to improvement.

4.4 Utility, Constraints and Objective

In any MARA problem, the utility function represents the degree of satisfaction of an agent for a given allocation [3]. Every agent has a utility value expressed as an explicit value or a relationship that reveals the most satisfactory solution (optimal). An allocation

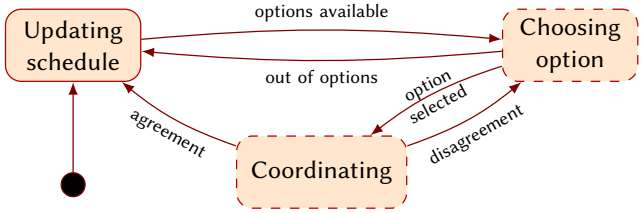

Figure 4: AV generic planning sub-behavior

procedure attempts to provide agents with alternative resources that match their utilities as much as possible. In this model, we define the AV agents' utility function based on the indicators of the quality of solution described in section 3. AVs are gluttons for request satisfaction to maximize their utility. That means the more requests an agent satisfies, the more value of utility he gains. Hence, the **Quality** indicator if considered individually for each agent $a \in \mathcal{V}$ defines its utility:

$$u_a = \frac{|R_a|}{|K_a|} \tag{6}$$

where R_a is the set of requests that are allocated to a (inserted in a's schedule to be served) and K_a is the set of all the known requests by a. Of course, allocating request r to agent a is constrained by the spatial and temporal availability of both a and r. We consider each request's origin and destination are constants, and a request is available to pick-up only at its origin during its defined time-window $w_r[l_r, u_r]$. Thus, allocating r to a requires that a can arrive in r's origin point at a time t in between the lower bound l_r and the upper bound u_r of r's time validity.

In this paper, we do not consider car-sharing scenarios. We assume a vehicle trip is dedicated to one request, but we still need to take the request size (required seats) and the vehicle capacity into account. This implies that the availability definition should also include the capacity constraint:

$$\exists t \in w_r[l_r, u_r] : \texttt{seats}(a, t) \geq s_r \;\&\; \texttt{loc}(a, t) = o_r \tag{7}$$

However, being members of a fleet impose that AVs are cooperative and must follow the coordination mechanism that is predefined to achieve their global objective. From a global perspective of ODT as a business model, the main objective of ODT service providers is optimize their benefits by reducing costs and raise the profit. From this point of view we can define the objective function $\mathcal{F}$ to be maximized by the allocation process based on the relation between the **Utility** and **Cost** indicators:

$$\mathcal{F} = \sum_{r \in \mathcal{R}_s} (P + p * \texttt{dist}(r)) - \sum_{v \in \mathcal{V}} cpd_v * \texttt{driven}(v) \tag{8}$$

where $\mathcal{R}_s \subseteq \mathcal{R}$ is the set of all satisfied requests, P is a fixed price (service fee) per request, p is a pricing factor per unit of travelled distance, `dist`(r) is the total trip distance for a request r and `driven`(v) is the total driven distance by v.

5 COORDINATION MECHANISMS

A coordination mechanism is defined by 3-tuple (DA, AC, AM), where DA denotes the level of decision autonomy which is either

centralized (C) or decentralized (D); AC denotes the agents' cooperativeness level with (S) or without sharing (N) of schedule information, and AM is the allocation mechanism name.

Although we support several coordination mechanisms, we consider in any scenario that the same fleet agents are homogeneous,i.e. they have the same coordination mechanism to prevent any ambiguous action in any scenario.

We can thus instantiate our generic model to implement coordination mechanisms from the literature, like: classical *selfish* behavior $\langle D, N, \text{Greedy}\rangle$ [22], centralized *dispatching* $\langle C, S, \text{MILP}\rangle$ [6, 13, 23], *cooperative* team using DCOP to coordinate $\langle D, S, \text{DCOP}\rangle$ [7], and *auction*-based allocation $\langle D, S, \text{Auction}\rangle$ [4, 5].

5.1 *Selfish* Behavior

In this case, the coordination mechanism is based on a decentralized allocation ($DA = D$) with competitive agents and without coordinated allocation process. In this model, agents do not rely on each other's decisions and never exchange their plans ($AC = N$). In real-world scenarios, one allocation mechanism to this model is a greedy-based one (AM = Greedy), in which the vehicle may consider only one request in advance (e.g., the closest one to shorten the empty driving distance). One of the recent examples of the Selfish behavior approaches is the work done in [22]. When a vehicle is not already carrying customers, it has to decide which request it will handle first, depending on the information it has about available requests. A heuristic computes a priority value for each request. Then, the agent handles the request with the highest priority value first. Conflicts can arise, but are solved simply by applying the first arrival policy. The set of options for an agent is its set of known requests; if the agent has no passenger on board, otherwise no option is considered. And because of that, the state *Coordinating* is ignored (as if it reaches an agreement for any chosen option), so that the quality of the solution is dependent on the agent's ranking and priority functions for the upcoming requests.

5.2 Centralized Dispatching

The difference from the previous coordination mechanism is that the allocation process is centralized ($DA = C$). The agent's role is to update its schedule based on what he receives from the dispatcher. In our model, we need one dispatcher per connected set. Thus, when created (or updated), a member of a connected set (e.g. the one with the lower index in the set) becomes the dispatcher that will be responsible for: gathering the information from other agents and about the request ($AC = S$), and doing the calculation on its own, or by calling an external service to obtain an optimal allocation (e.g. a *MILP* solver in this case $AM = MILP$) , and then send to each other vehicle its potential schedule, as in [6, 13, 23]. In this centralized model, the role of AVs' planning behavior is to request a communication portal (the responsible agent) continuously to update their schedule. In this case, the only available option is to request the portal, and the *coordinating* state consists of a request/response protocol that will send the new schedule as an agreement.

5.3 DCOP-based Cooperative Coordination

In this case, the coordination mechanism is decentralized ($DA = D$), the agents are cooperative ($AC = S$), and a coordination protocol is applied by the allocation process. In this class of coordination mechanisms, agents exchange information and cooperate to achieve a common objective, avoiding conflicts and optimizing the solution quality. There exist several approaches to implement an allocation mechanism for this behavior like distributed constraint optimization (DCOP) [7]. Here, agents decide on their own but coordinate with the same connected set agents using a distributed constraint optimization algorithm to avoid conflicts within the connected set. At each time a connected set changes, a DCOP: (A, X, D, C) is generated from the AV-OLRA instance to maximize objective function in Equation 8, as follows. A defines the set of agents in the connected set. X defines the set of decision variables in three subsets (x_{ij}'s, y_{ij}'s and z_{ij}'s): $x_{ij} \in X$ is a binary variable that takes 0 as a value if vehicle v_i serves the request r_j; y_{ij} is a binary variable that takes 1 only if the request r_j is the first request to be served by v_i. Finally, z_{ij} is an integer variable that defines at what time a request r_j is visited by v_i. D defines the domains of variables: binary $\{0, 1\}$ for x_{ij}'s and y_{ij}'s, and a set of time ranges domains defining the time-window range $[l_j, u_j]$ for each z_{ij}. C defines the set of constraints, which consists of hard constraints (capacity, Spatio-temporal availability, and time-windows) and soft constraints defining the cost and utility of the allocation decision (used to calculate the value of the objective function).

DCOP algorithms are varied, and the choice between them is dependent on the objective of the solution and the context of the problem. The run-time characteristics of the DCOP algorithm (execution-time, number/size of messages, and memory requirement per agent) is an essential factor when dealing with on-line dynamic problems. Considering the characteristics of our very problem, we chose to implement local search algorithms to solve AV-OLRA, namely MGM [17] and DSA [25].

5.4 *Auction*-based Coordination

The difference from the previous behavior is that the agents are competitive so that they follow market-based protocols to achieve agreements. Auctions are very common in everyday situations and provide a general conceptual basis for understanding resource allocation problems within sets of self-interested agents [21]. Here we present our previous work *ORNInA* [4] as an example of an auction-based coordination mechanism ($AM = Auctions$) to coordinate in a Peer-to-Peer manner ($AC = S$) the decentralized scheduling decisions of fleets of autonomous vehicles ($DA = D$). This mechanism is proposed to perform in dynamic settings, between vehicle agents that belong to a connected set in which they can receive and send direct or broadcast messages. Agents interested in some request initiate first-price auctions for this request, and the winner adds it to its schedule. The winner determination is completely decentralized process. To improve the scheduling efficiency in dynamic settings, agents are allowed to exchange their scheduled requests at run-time, with additional auction rounds to decide if this exchange increase the value of the objective function within the connected set. Agents communicate with each other by direct and broadcast messages to share information or coordinate their decisions.

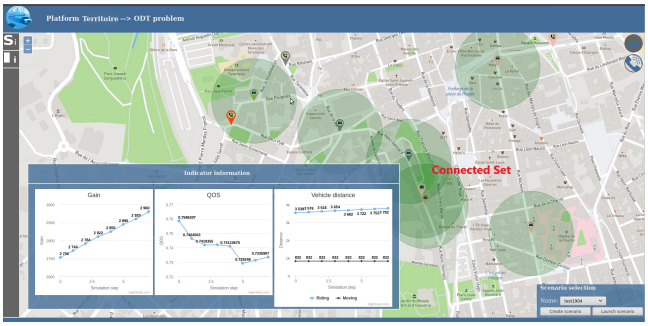

Figure 5: The simulator interface of *Plateforme Territoire*

6 EXPERIMENTAL EVALUATION

In this section, we present the experimental results of instantiating the AV-OLRA model with the multi-agent model described in Section 4, supporting the different types of coordination mechanisms of Section 5. The model is implemented as a multi-agent system with a discrete-time transport simulator.

Experimental Setting. We use a unique urban infrastructure map for all our experiments. For a district located between (45.4325,4.3782) and (45.437800,4.387877), more than 1400 edges have been extracted from Open Street Map (OSM)[1] and post-processed by *Plateforme Territoire*[2] to produce a graph of 71 edges. The passenger requests are generated randomly with pick-up and delivery locations belonging to a specific set of locations called sources. 40 locations uniformly distributed through the map were selected for being demand emission sources. When vehicles have to directly exchange messages, we consider they communicate via DSRC[3] with a realistic communication range of 250 meters. Vehicles are also able to communicate by transitivity to any other vehicle in their connected set. The number of generated requests and the number of vehicles are parameters of the simulation. All scenarios were 1000-cycle long, and at each time cycle, 0 or 1 request is generated. We will evaluate the performance of five coordination mechanisms: *Selfish*, *Dispatching*, *Auction*-based, *Cooperative* with DSA (variant A, $p = 0.5$) [25], and *Cooperative* with MGM-2 [17]. These benchmarks are picked and implemented to be proof of concept for the genericity of our model. We chose the local search *MGM-2* and *DSA* as examples of DCOP algorithms that require linear memory space. Once modeled, DCOP instances can be passed to any DCOP local search or inference algorithm. The difference here is in the performance, infrastructure requirements, and completeness of the chosen benchmarks. The Java-based multi-agent system and simulator have been executed on an octa-core Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, with 32GB DDR4 RAM. DCOP algorithms have been implemented using the FRODO library [12].

Quality of the Solutions. Figures 6 and 7 illustrate the performance of the five coordination mechanisms in terms of the QoS and QoB indicators. Every point on this diagram represents the indicator value aggregated over 1000 cycles of simulations. These two figures show how the quality of the solution evolves with increasing fleet size. We can notice the increase in QoS and QoB with the increasing number of vehicles in the fleet until reaching a threshold of repletion, after which it is not possible to improve the quality by adding more vehicles.

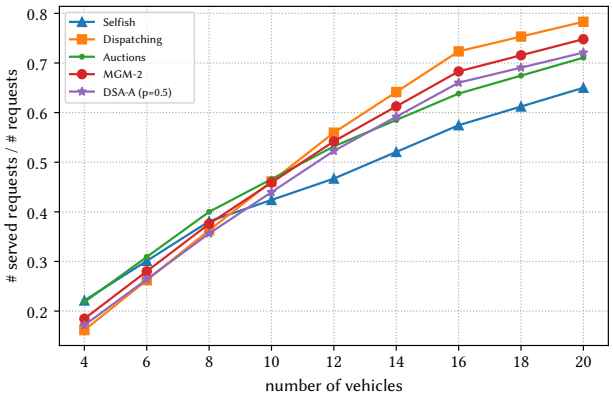

Figure 6: QoS evolution

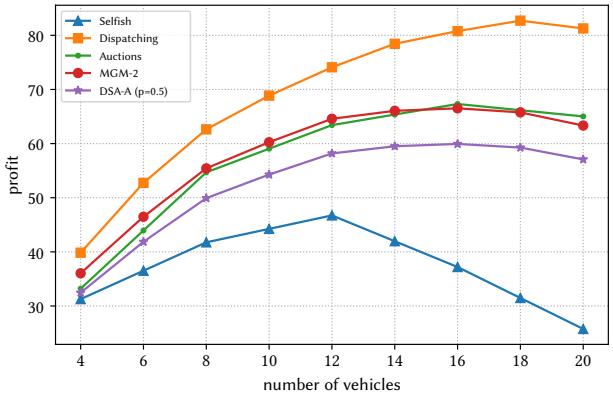

Figure 7: QoB evolution

The values obtained by the *Passive* behavior represent somehow an upper-bound for the objective function (QoB) as the central dispatcher calculates for each instance the optimal solution (locally optimal considering the context of the connected set). The performances of the four other mechanisms vary between the indicators. So, while the auction-based coordination performs better than the DCOP search algorithms in QoB, it is outperformed by both of them in terms of QoS.

The *Selfish* mechanism is very efficient in terms of time to make decisions. The reason is that it does not require many computations to select the closest request. The drawback here is the ignorance of what lies ahead of the taken decisions and the fact that the conflicts in decisions of different vehicles (e.g., two vehicles are going to pick the same passenger) are not resolved until a very late stage, which reduces the QoB. With a low number of vehicles, the connected sets

[1] https://www.openstreetmap.org
[2] https://territoire.emse.fr/
[3] Vehicle-to-vehicle communication via Dedicated Short-Range Communication (DSRC) provides low latency, fast network connectivity within a communication range up to 300 meters.

Table 1: Communication cost and statistics for different coordination mechanisms for scenarios with ten vehicles

Coordination	max msg size	avg msg size	msg per agent	comm. load	reschedule rate
Selfish	140	88	6	2.21 MB	2.0
Dispatching	3500	168	21	11.2 MB	3.0
Auction	140	112	53	37.7 MB	1.5
MGM-2	210	25	5040	297.6 MB	12.0
DSA	236	20	5015	75.1 MB	13.0

are small and, as a consequence, the amount of shared information is reduced.

The performance of the *Cooperative*, *Auction* and the centralized *Dispatching* mechanisms is highly dependant on the amount of information so that there are no quality discrepancies between the four approaches. With higher fleet sizes, more information is shared in the connected sets. Additionally, vehicles switch from one connected set to another more frequently. The cooperative and auction approaches perform almost similarly and acheive reasonable values of QoS. To achieve the same values with the *Selfish* behavior, more vehicles in the fleet are required.

Communication Load. Table 1 shows values of the indicators related to communication obtained by simulating a scenario with ten vehicles over 1000 cycles with different behaviors. Here the second and third columns report the maximum and average size of exchanged messages (in bytes) representing the **MsgSize** indicator. The fourth column reports the **MsgCount** indicator in terms of the average number of messages received by an agent per simulation cycle.

Even with *Selfish* behavior, agents exchange information messages about the new requests announced. New types of messages are used in the *Dispatching* mechanism: the query and response messages exchanged between the vehicles and the central dispatcher. Query messages are simply the whole context of the connected set of vehicles that ask the dispatcher to build their schedules. Response messages are sent from the dispatcher to the individual vehicles and contain each individual's potential schedule. These messages can be large, depending on the size of the sub-problem. Bid and answer messages used by the *Auction*-based coordination mechanism are light-weight, so that the values of the **MsgSize** indicator stay close to the no-coordination one, while the **MsgSize** value becomes polynomial in the number of agents in the connected set and number of their known requests. In the two *Cooperative* coordination mechanisms (DSA and MGM-2), agents in a connected set instantiate a DCOP framework between each other each time they need to decide on a schedule update. Achieving a solution by one of these algorithms requires the exchange of a large number of messages, both of these algorithms are not complete, meaning that they continue their trials to improve the solution until reaching the timeout or local optimum. This will lead to more message exchange. On the other hand, the size of messages exchanged by these two approaches is very small compared to the other approaches.

Stability of Schedules. Table 1 reports also the rescheduling frequency by considering the average interval between two simulation cycles in which vehicles update their schedules. The higher this value is, the more stable the vehicle schedules. In dynamic settings, having stable schedules for a long time means that no new requests are inserted, affecting the QoS. On the other hand, when vehicle schedules change frequently, vehicles may change their destination and oscillate for a while before performing a successful trip, which could decrease QoB. In our scenarios, *Cooperative* coordination provides very stable and good quality schedules at the expense of a higher communication load. If stability is not a constraint, but communication is limited, *Auction* is a very good candidate.

7 CONCLUSION AND FUTURE DIRECTIONS

This paper proposes a model for a resource allocation problem encountered in the management of autonomous vehicle fleets. Our model is well suited to the ODT domain, where fleets respond to passenger demands in dynamic online environments. Our model can handle different types of constraints and allow different approaches to find solutions and coordinate vehicles. We have implemented a multi-agent system that delivers this model in which agents can communicate with each other via radio channels using peer-to-peer messages. The communication model supports direct, broadcast, and transitive message transmission and is based on the concept of connected sets. We aimed to provide a generic model; the implemented MAS offers genericity on both communication and coordination dimensions. On the one hand, the limited communication range defines an attribute for the problem that affects the level of connectivity and thus bounds the achievable centralization. On the other hand, being dependant on allocation process, the choice of AV's planning sub-behavior defines the coordination mechanism that affects the dynamic spatial-temporal context of the problem instances.

To experiment and assess this generic model, we instantiated several coordination mechanisms and we briefly compare them according to functional and technical indicators. Relying on DCOP or auctions to coordinate decentralized decisions provides reasonable quality allocations compared to optimal one-shot allocation and non-coordinated taxis. DCOP-based allocation strategies do not change vehicle schedules too frequently but still induce more communication than the auction-based strategy.

A limitation of our communication model is the phenomena of spatially obscure demands. Those are requests announced far from vehicles and could remain unknown to any connected set for a while until a vehicle passes close to their sources, so they may not be met within their time-window constraints. However, in this work, we assumed very dynamic scenarios in the spatial and temporal dimensions so that no such situation would occur in any of our experimental scenarios.

We plan to explore the direction of defining further constraints on vehicle motion to achieve more connectivity between vehicles or to ensure that each emission source is located within the communication space of at least one vehicle. We plan also to analyze in-depth the relationship between stability, completeness, and feasibility of the solutions in the future. To do so, we need to implement more sustained approaches of different types and systematically compare performance, quality, feasibility, stability, and technical issues for these approaches' practical application.

REFERENCES

[1] Ana LC Bazzan and Franziska Klügl. 2014. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review* 29, 3 (2014), 375–403.

[2] Deng-Neng Chen, B. Jeng, Wei-Po Lee, and Cheng-Hung Chuang. 2008. An agent-based model for consumer-to-business electronic commerce. *Expert Systems with Applications* 34, 1 (Jan. 2008), 469–481. https://doi.org/10.1016/j.eswa.2006.09.020

[3] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian A. Padget, Steve Phelps, Juan A. Rodr, and Paulo Sousa. 2006. Issues in Multiagent Resource Allocation. *Informatica* 30 (2006), 37.

[4] Alaa Daoud, Flavien Balbo, Paolo Gianessi, and Gauthier Picard. 2020. ORNInA: A decentralized, auction-based multi-agent coordination in ODT systems. *AI Communications* (2020), 1–17.

[5] Malcolm Egan and Michal Jakob. 2016. Market mechanism design for profitable on-demand transport services. *Transportation Research Part B: Methodological* 89 (2016), 178–195.

[6] Mohamad El Falou, Mhamed Itmi, Salah El Falou, and Alain Cardon. 2014. On demand transport system's approach as a multi-agent planning problem. In *2014 International Conference on Advanced Logistics and Transport (ICALT)*. IEEE, IEEE, Tunis, Tunisia, 53–58.

[7] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. 2018. Distributed Constraint Optimization Problems and Applications: A Survey. *Journal of Artificial Intelligence Research* 61 (March 2018), 623–698. https://doi.org/10.1613/jair.5565

[8] Vasco Furtado, Adriano Melo, André Coelho, and Ronaldo Menezes. 2007. A crime simulation model based on social networks and swarm intelligence. In *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*. ACM Press, Seoul, Korea, 56. https://doi.org/10.1145/1244002.1244014

[9] Andrey Glaschenko, Anton Ivaschenko, George Rzevski, and Petr Skobelev. 2009. Multi-Agent Real Time Scheduling System for Taxi Companies. *AAMAS* (2009), 8.

[10] Xu Jin and Luo Jie. 2012. A Study Of Multi-Agent Based Model For Urban Intelligent Transport Systems. *International Journal of Advancements in Computing Technology* 4, 6 (April 2012), 126–134. https://doi.org/10.4156/ijact.vol4.issue6.15

[11] Adrien Lammoglia, Roger Marcelin Faye, and Didier Josselin. 2012. A dynamic cooperation modelling for improving taxi fleet efficiency. In *proceeding of AGILE'2012*. Jérôme Gensel, Didier Josselin and Danny Vandenbroucke, Avignin, 6.

[12] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek. 2009. FRODO 2.0: An Open-Source Framework for Distributed Constraint Optimization. In *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*. Pasadena, California, USA, 160–164. https://frodo-ai.tech.

[13] Der-Horng Lee, Hao Wang, Ruey Cheu, and Siew Teo. 2004. Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions. *Transportation Research Record: Journal of the Transportation Research Board* 1882 (Jan. 2004), 193–200. https://doi.org/10.3141/1882-23

[14] Pangfeng Liu, Yi-Fang Lin, Jan-Jan Wu, and Zhe-Hao Kang. 2007. An optimal scheduling algorithm for an agent-based multicast strategy on irregular networks. *The Journal of Supercomputing* 42, 3 (Oct. 2007), 283–302. https://doi.org/10.1007/s11227-007-0116-6

[15] Michał Maciejewski and Kai Nagel. 2013. The influence of multi-agent cooperation on the efficiency of taxi dispatching. In *International conference on parallel processing and applied mathematics*. Springer, 751–760.

[16] Pierre Monier, Arnaud Doniec, Sylvain Piechowiak, and René Mandiau. 2011. Comparison of DCSP Algorithms: A Case Study for Multi-agent Exploration. In *Advances on Practical Applications of Agents and Multiagent Systems (Advances in Intelligent and Soft Computing)*, Yves Demazeau, Michal Pěchoucěk, Juan M. Corchado, and Javier Bajo Pérez (Eds.). Springer, Berlin, Heidelberg, 231–236. https://doi.org/10.1007/978-3-642-19875-5_30

[17] Jonathan P. Pearce and Milind Tambe. 2007. Quality Guarantees on K-Optimal Solutions for Distributed Constraint Optimization Problems. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (Hyderabad, India) *(IJCAI'07)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1446–1451.

[18] Nicole Ronald, Russell Thompson, and Stephan Winter. 2015. Simulating demand-responsive transportation: a review of agent-based approaches. *Transport Reviews* 35, 4 (2015), 404–421.

[19] Kiam Tian Seow, Nam Hai Dang, and Der-Horng Lee. 2010. A Collaborative Multiagent Taxi-Dispatch System. *IEEE Transactions on Automation Science and Engineering* 7, 3 (July 2010), 607–616. https://doi.org/10.1109/TASE.2009.2028577

[20] Wen Shen and Cristina Lopes. 2015. Managing Autonomous Mobility on Demand Systems for Better Passenger Experience. *arXiv:1507.02563 [cs]* 9387 (2015), 20–35. https://doi.org/10.1007/978-3-319-25524-8_2 arXiv: 1507.02563.

[21] Yoav Shoham. 2009. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press. 532 pages.

[22] Rinde R.S. van Lon, Tom Holvoet, Greet Vanden Berghe, Tom Wenseleers, and Juergen Branke. 2012. Evolutionary synthesis of multi-agent systems for dynamic dial-a-ride problems. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12*. ACM Press, Philadelphia, Pennsylvania, USA, 331. https://doi.org/10.1145/2330784.2330832

[23] Li Yang, Zhao Jieru, Chen Jingxin, and Tang Zhiyong. 2017. Central Decision Intellective Taxi System and Multi Ride Algorithm. In *Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies (AIACT '17)*. ACM, New York, NY, USA, 5:1–5:6. https://doi.org/10.1145/3080845.3080850 event-place: Wuhan, China.

[24] Mahdi Zargayouna, Flavien Balbo, and Khadim Ndiaye. 2016. Generic model for resource allocation in transportation. Application to urban parking management. *Transportation Research Part C: Emerging Technologies* 71 (2016), 538 – 554. https://doi.org/10.1016/j.trc.2016.09.002

[25] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenburg. 2005. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence* 161, 1 (2005), 55 – 87. https://doi.org/10.1016/j.artint.2004.10.004 Distributed Constraint Satisfaction.