A Multi Agent based Organizational Architecture for Dynamic Pickup and Delivery Problem

Tahar Guerram

Department of Mathematics and Computer Science, University Larbi Ben M'hidi of Oum El Bouaghi, Algeria

Pickup and Delivery Problem (PDP) consists of searching an optimal set of vehicles and an optimal set of routes, one route by each vehicle, in order to pickup items from a set of origins and deliver them to another set of destinations. Pickup and delivery problem is a class of complex systems whose complexity is NP Hard. In PDP real life applications, heuristics and meta heuristics methods are used in order to obtain an acceptable solution in reasonable execution time. When unpredictable events, like for example path cut and vehicles failure, may occur during the PDP schedule execution, we say that the PDP is dynamic (DPDP) and in this case we have to revise this schedule. In this paper, we propose a multi agent architecture for DPDP based on an organizational architecture. Supported by a formal framework, the proposed architecture allows us to show, through a case study that computed solution for the studied problem could be done in a parallel manner which attenuates substantially the problem complexity.

ACM CCS (2012) Classification: Applied computing → Operations research → Transportation
Computing methodologies → Artificial intelligence → Distributed artificial intelligence → Multi-agent systems
Computing methodologies → Artificial intelligence → Planning and scheduling → Multi agent planning.

Keywords: PDP, DPDP, agent, multi agent system, complex system

1. Introduction

The goal in a pickup and delivery problem is to form a schedule for a set of vehicles to pickup and deliver a set of items while satisfying a set of constraints and minimizing an objective function [1]. A general configuration of a pickup and delivery problem may be described as follows. We have a finite number of origin locations, a finite number of destination locations, a finite number of vehicles, a finite set of items at the origin locations, a route network giving links between origin locations and destination locations, and a finite number of requests on the above items. Vehicles are characterized by a set of constraints like for example vehicle maximum capacity, vehicle initial location, and vehicle maximum time duration. Also Items are characterized by a set of constraints such as pickup management time, delivery management time, pickup time window and delivery time window. Likewise, each request is defined by certain of parameters and constraints like the requested item and the delivery time window concerning this item. In real life applications, pickup and delivery problems are dynamic, means that a certain number of requests at least are not known in advance and requests cancellations are considered as well. Also, path cuts and vehicle failures are taken into account. In our PDP configuration, we have:

1. many origins, many destinations, many items and many vehicles

2. origins and destinations are geographically distributed

3. Each vehicle may satisfy one or more requests at once,

4. requests are indivisible,

5. vehicles' capacities are different,

6. two sources of dynamicity which are vehicle failures and path cut due to bad weather conditions,

7. The system output is a schedule which minimize three criteria: the number of vehicles used, the global distance traveled by all these vehicles and the number of requests satisfied with delay.

The remaining of this paper is organized as follows. The second section is dedicated to the study of the recent and significant work in this class of problems. For each of these papers, we try to illustrate its strengths and limitations. In the third section; we give mathematical formulation of our PDP. The fourth section is devoted to the description of the proposed multi agent based architecture. In the fifth section we explain the functionality of our multi agent system. In the sixth section we present a case study allowing us to illustrate our approach. Finally, the seventh section allows us to conclude our work and to give some future perspectives.

2. Related Work

Jaroslaw *et al.* [2] focused on the dynamic pickup and delivery problem with Time Window (PDPTW) where each request is associated by the allowed periods of time when pickup and delivery may be performed. This work proposes a model which allows, during a transport plan creation, to take into account probability of requests arrival at any node in the road network. The multi agent architecture is composed of the environment which consists of the road network and the following types of agents:

1. Agent-vehicle which represents a vehicle that travels between the nodes in the transport network and possesses plans and schedules of the requests to be realized.

2. Agent-customer: This agent is responsible of the generation and sending of transport requests to the agent-company.

3. Agent-company: is responsible for receiving requests from customer agents.

The model is implemented using Madkit multi agent platform [3] and the obtained results show that for low number of vehicles and rarely arriving requests, the applied method increases the degree of performed requests and produces a higher income.

The authors intend to enrich their model by taking into the account traffic jams and introducing cooperation between different transport companies.

Rubinstein and Smith [4] developed a mixed-initiative tool for real-time management of paratransit vehicle schedules. This tool is a dynamic execution-driven process, where unexpected events such as traffic, breakdowns, new requests and cancelations, continually force changes to yet established schedules. The paratransit management problem is an instance of the Dial-a-Ride Problem (DARP) [5]. To manage dynamic events, this system uses a temporal network mapping the different requests of the customers and their constraints, and the arriving dynamic events to adjust vehicle schedules. Experiments show potential improvements comparatively to human rearranged schedules ones. Future improvements of the proposed system are planned by the authors in order to minimize customers waiting time and then their overall travelling time. The paper neither gives details about the implementation of this system nor does allusion to the multi agent architecture which is high recommended for this type of applications.

Manal and Anjali [6] proposed A multi agent architecture to collaborative logistics systems aiming to overcome to the problem of unorganized communication between various logistical entities, considered as the principal cause of the deficiency in the satisfaction of customer's requests. This architecture is composed of the following interacting agents:

1. RFIDG Agent: The agent's major role is to receive data from the RFID reader and places them into the merchandise database

2. Retailer Agent: When a human retailer logs into the system to request a shipment; the retailer agent notifies both the supplier agent and the carrier agent.

3. Supplier Agent: When informed by the retailer agent about a shipment, the supplier agent starts searching inside its database about the requested products and then, replies back to the retailer about the status of the request as either available or not.

4. Carrier Agent: The carrier agent has a fleet of vehicles to manage for the satisfaction of shipment requests.

5. Network Agent: This agent is responsible about measuring congestion on delivery routes and informing the city administrator agent about existing severe congestion.

6. City Administrator Agent: This agent is responsible of the application of the city's rules on the freight delivery plan.

In case of rules violation, the city administrator agent recommends other alternative solutions. Both supplier agents and carrier agents collaborate using game theory in order to satisfy retailers' needs.

In this paper, implementation aspects are note given by the authors but they judge that implementing their system allows reaching several essential advantages such as real-time response, support decision making, cost reduction, increase suppliers' profits and attain time management.

Simon *et al*. [7] proposed a general multi agent based framework for scheduling and routing Agents using different meta-heuristics to search cooperatively the desired solution. This framework makes use of two types of agents: Launcher and Meta-heuristic agents:

1. Launcher agent: is responsible for queuing the problem instances to be solved for a given domain, configuring the meta-heuristic agents, successively passing a given problem instance to the meta-heuristic agents and gathering the solutions from the meta-heuristic agents.

2. Meta-heuristic agent: Each meta-heuristic agent executes one of the meta-heuristic/ local search heuristic combinations defined by the launcher agent.

Agents communicate with each other's using a routing and scheduling ontology which defines the structure of messages to be interchanged in the well known contract net protocol in order to score the solutions (routes) found by the meta-heuristics agents. The agent whose individual solution has the best scoring is designated to be the next initiator agent during the next conversation. Many conversations are needed to reach the final solution. The system is implemented using JADE multi agent platform [8] and applied on two domains, Capacitated Vehicle Routing and Permutation Flow-shop Scheduling. Experiments show Improving results for the first problem and up to date for the second one.

Another work illustrating a real life application of Dynamic PDP consisting of a method of dynamic planning of mobile agents' tasks [9]. Here, mobile agents are asked to do tasks using services offered by geographically distributed service proxies or workplaces and mobile agents build and rebuilt dynamically their plans showing paths to hosts to visit in order to accomplish their tasks. This is possible using acquaintance recommendation done cooperatively by other agents allowing having information about available workplaces and services they can provide. This dynamic planning system allows mobile agents to have adaptive behavior. Unfortunately, time and generated cost are not taken into account which calls into question the response time of the system and its feasibility.

Constrained multi agents architectures for dynamic planning of many variants of PDP were proposed [10], [11], [12], [13], [14]. Here agents search cooperatively an optimal plan satisfying a predefined set of constraints distributed over them. Although this approach gives good results in the case of a small size problem and a static environment, the performance of these systems falls in the case of a large distributed problem with a dynamic environment (important number of constraints which are distributed over agents and these constraints may change anytime). This is explained by the fact that, when searching for a global optimal plan, each agent maintaining a certain number of constraints has to share their values (values of constraints) every time with other agents of the system until reaching a global optimal plan satisfying all the constraints and generated cost verifies a predefined objective function. These interactions between agents generates a huge number of communications which negatively influences system's performance, namely when the environment's dynamicity is high.

Unlike the previous approaches, the proposed approach aims to find an optimal plan for the PDP problem by organizing agents in an organizational architecture so that the computing of this optimal plan is done largely in a parallel manner, avoiding interactions between agents whenever possible, which will considerably attenuate the complexity of the studied problem and hence improve the performance of the searched solution.

3. Mathematical Formulation of Our PDP

3.1. Variables

The variables of our PDP problem are as follows:

- A set O of n origin locations. Each origin location represents a supplier.

$$O = \{o_1, o_2, ..., o_n\} \quad (1)$$

- We have a set M of a items distributed on the above origin locations. The same item may exist in more than one origin location with different quantities.

$$M = \{m_1, m_2, ..., m_a\} \quad (2)$$

- Each origin location o_j contains items belonging to a set $T(o_j)$ of k items, where each item is assigned a quantity.

$$T(O_j) = \{m_1, m_2, m_3, ..., m_k\} \quad (3)$$

such that: $j \in \{1, ..., n\}$ and $k \in \{1, ..., a\}$.

- A set D of m destination locations, where each destination location d_j represents a customer.

$$D = \{d_1, d_2, ..., d_m\} \quad (4)$$

- A set V of x vehicles.

$$V = \{v_1, v_2, ..., v_x\} \quad (5)$$

- A set H of s vehicle home location agents $h_i, i \in \{1, ..., s\}$

$$H = \{h_1, h_2, h_3, ..., h_s\} \quad (6)$$

The set V is divided into s subsets of vehicles and each one is assigned to a vehicle home location $h_i \in H, i \in \{1, ..., s\}$

- A set R of t requests coming from the destination locations to origin locations ordering items with certain quantities:

$$R = \{r_1, r_2, r_3, ..., r_t\} \quad (7)$$

3.2. Hypothesis, Parameters and Constraints

The hypothesis, parameters and constraints of our PDP problem are as follows:

- Each vehicle $v_i \in V, i \in \{1, ..., x\}$ has a load capacity, has a minimum speed, a maximum speed, a maximum distance to travel before having a rest of half (½) hour, and has a cost per istance unit. We suppose also that each vehicle is always free to serve when needed.

$$Params(V_i) = \begin{pmatrix} mx\_spd, mn\_spd, \\ mx\_dist, costu \end{pmatrix} \quad (8)$$

- Each item $m_i \in M, i \in \{1, ..., k\}$, is assigned a load time and unload time per quantity unit.

$$Params(m_i) = \begin{pmatrix} load\_time, \\ unload\_time \end{pmatrix} \quad (9)$$

- Each request $r_i \in R, i \in \{1, ..., t\}$ comes from a destination location, to an origin location concerning a certain item with a certain quantity, dated with a request date and with a limit date.

$$Params(r_i) = \begin{pmatrix} o, d, m, req\_date, \\ limit\_date \end{pmatrix} \quad (10)$$

If the requested quantity of an item is greater than one existing at the concerned origin location, then the request is rejected or, if possible, redirected to another origin location of the same group of origins.

- We suppose that each request $r_i \in R$ is received by the receptor origin location in the same day it is sent.

- The delivery time concerning a request $r_i \in R, i \in \{1, ..., t\}$, transported by a vehicle $v_j \in V, j \in \{1, ..., x\}$, is given by:

$$Delivery\_time_{r_i, y_j} = Load\_time_{r_i, y_j}$$
$$+ \ Unload\_time_{r_i, y_j} + Rest\_time_{r_i, y_j}$$
$$+ \ Transp\_time_{r_i, y_j} + Repair\_time_{r_i, y_j}$$
$$+ \ Deviat\_time_{r_i, y_j} \quad (11)$$

- The time delay, expressed by hours, in satisfying a request r_i transported by a vehicle v_j is computed in terms of hours as follows:

$$Time\_delay_{r_i,v_j}$$

$$= \left(\begin{array}{c} Delivery\_time_{r_i,v_j} \\ -Limit\_date_{r_i,v_j} \end{array} \right) \times 24 \quad (12)$$

- Penalties due to delays in satisfying a request $r_i \in R$ ($i \in \{1, ..., t\}$), transported by a vehicle $v_j \in V$ ($j \in \{1, ..., x\}$) is computed regarding the relation:

$$Penalty\_cost_{r_i,v_j}$$
$$= 1000 \times Time\_delay_{r_i,v_j} \quad (13)$$

- The relation (14) below gives the expression of the total distance travelled by a vehicle $v_j \in V$ ($j \in \{1, ..., x\}$ transporting a subset of requests $P_i = \{r \in R\}$; such that $i \in \{1, ..., g\}$; g is the number of requests' subsets such that each request subset P_i can be delivered by one vehicle $v_j \in V$.

$$Distance_{P_i,v_j} = Pikup\_dist_{P_i,v_j}$$
$$+ Transp\_dist_{P_i,v_j} + Delivery\_dist_{P_i,v_j}$$
$$+ Deviat\_dist_{P_i,v_j} \quad (14)$$

Pickup distance indicates a pickup path, delivery distance indicates a delivery path, and transportation distance indicates a path linking the two precedent paths. Deviation distance indicates a linking between two nodes delimitating a path cut. When requests can't be grouped, means that a vehicle satisfies only requests concerning only one destination location then pickup route and delivery route are reduced to an origin location and destination location respectively and hence the total distance is reduced only to the transportation distance plus deviation distance.

- The delivery cost due to the satisfaction of a subset of requests $P_i = \{r \in R\}$, $i \in \{1, ..., g\}$ transported by the vehicle $v_j \in V$ is computed as given in relation (15) as follows:

$$Delivery\_cost_{P_i,v_j} = \left(\begin{array}{c} Transp\_cost_{P_i,v_j} \\ +Penalty\_cost_{P_i,v_j} \\ +Repair\_cost_{P_i,v_j} \end{array} \right) \quad (15)$$

Repair cost consisting of handwork and spare parts costs spent in the case of a failure of the vehicle v_j. We give below the expressions of transportation cost and penalty cost.

$$Transp\_cost_{P_i,v_j} = Distance_{P_i,v_j} \times costu_{v_j} \quad (16)$$

$$Penalty\_cost_{P_i,v_j} = \sum_{r \in P_i} Penalty\_cost_{r,v_j} \quad (17)$$

3.3. Searching an Optimal Solution

The solution to this problem is searched in two steps. The first step consists of grouping the requests over the vehicles to be used and the second step is routing this groups of requests. We want to find solution which minimize the total number of vehicles used and also minimize the total distance travelled by these vehicles. After grouping the requests, we find the best path of each vehicle by:

a) Searching the best pickup path in the same set of vehicles of the same home location,

b) Searching the best delivery path between the destination locations having ordered items to origin locations attached to the same group location and,

c) Searching the best path linking the two precedent paths (linking the last node in the pickup path and the first node in the delivery path).

We want to minimize also the number of requests satisfied after the limit date fixed by customers. The objective function F to optimize will be expressed by the relation (18) below., where α, β and δ are weights(degrees of importance) attached to the different parameters of the objective function

$$Best\_delivery\_cost$$
$$= \mathbb{F}\left(all\_tc\_, all\_pc, all\_rc\right)$$
$$= Min\left(\alpha \times \sum_{i,j} tc_{p_i,v_j} + \beta \times \sum_{i,j} pc_{p_i,v_j} \right.$$
$$\left. + \delta \times \sum_{i,j} rc_{p_i,v_j}\right) \quad (18)$$

Such that: $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$, $0 \leq \delta \leq 1$ and $\alpha + \beta + \delta = 1$. tc, pc and rc abbreviate $transport\_cost$, $penalty\_cost$ and $repair\_cost$, respectively.

4. The Proposed Multi Agent Architecture

The proposed architecture depicted in Figure 1 is composed of the following agents and groups of agents.

- Groups of destination location agents: These groups of agents are considered as customers sending requests to the request receiver agents. Groups of destination agents are constituted using proximity criteria such as all the agents which fall in the ray of a predefined distance form one group. Each group of destination agents is associated to a request receiver agent. Figure 2 shows the life cycle of a destination location agent.

- A group of request receiver agents: Each request receiver agent is assigned to a group of destination location agents. The role of these agents is to acquire requests, characterized by the relation (10) in section 3.2, coming from the destination location agents and redirect them to the concerned vehicle home location agents. To be able to do so, each request receiver agent maintains a list of all vehicle home location agents existing in the system and all the origin location agents attached to

each vehicle home location agent. Figure 4 shows the life cycle of a request receiver agent.

- Groups of origin location agents: These groups of agents are considered as suppliers which satisfy or reject destination agent requests. These groups are constituted using proximity criteria such as all the agents which fall in the ray of a predefined distance form one group. Each origin location agent maintains a list of items. Figure 3 shows the life cycle of an origin location agent.

- A group of vehicle home location agents: The number of these agents is equal to the number of vehicle home locations. The role of each vehicle home location agent is to search the best grouping of its local requests over all the vehicles attached to it. Each vehicle home location agent outputs one or more vehicles which will be used to satisfy the quoted requests. In order to achieve their missions, each vehicle home location agent launches a contract net protocol with vehicle agents under its responsibility asking them to do proposals satisfying specified requests under given. conditions (specified by the parameters of the requests). In order to have a minimal
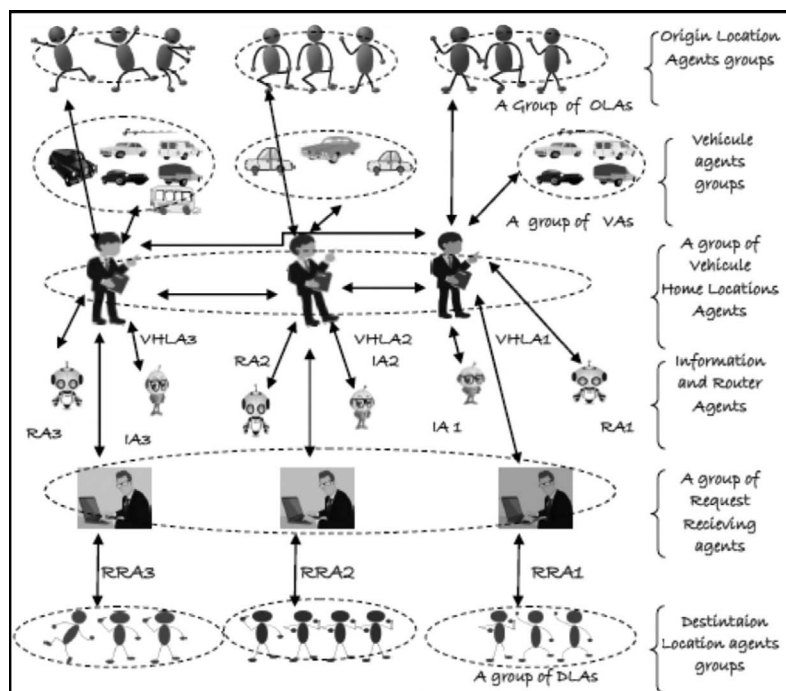

Figure 1. The proposed multi agent architecture for Dynamic PDP.

global set of vehicle agents, vehicle home location agents are invited to cooperate with each other's by sharing their information about locally determined vehicle agent and the requests they will satisfy. When this minimal set of vehicle is determined, each vehicle home location agent creates a router agent for each vehicle agent deliverer in order to compute an optimal path. When informed by the concerned vehicle agent about path cut event, vehicle home location agent creates a router agent in order to compute the best alternative path. As soon as informed by the concerned vehicle agent about vehicle failure event (diagnostic results and failure place address) vehicle home location agent ask an information agent to search the best proposal (considering time and cost) in the neighborhood of the failure place allowing to troubleshooting the vehicle failure. Figure 5 shows the life cycle of a vehicle home location agent.

- A group of information agents: Each vehicle home location agent is attached to an information agent whose the role is to supply the former with needed information agent has the capacity and the role, if asked, to deliver items to the concerned destinations using a path plan computed by a router
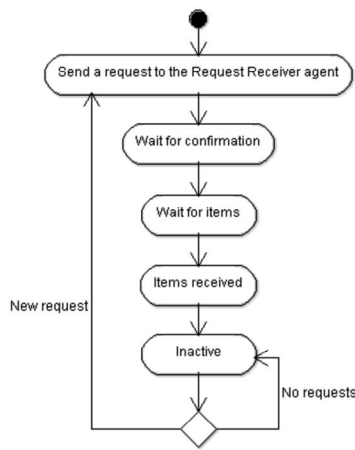

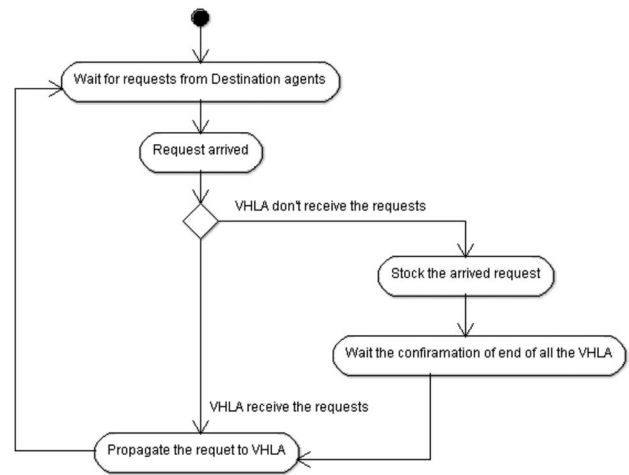*Figure 2.* Life cycle of a destination location agent.

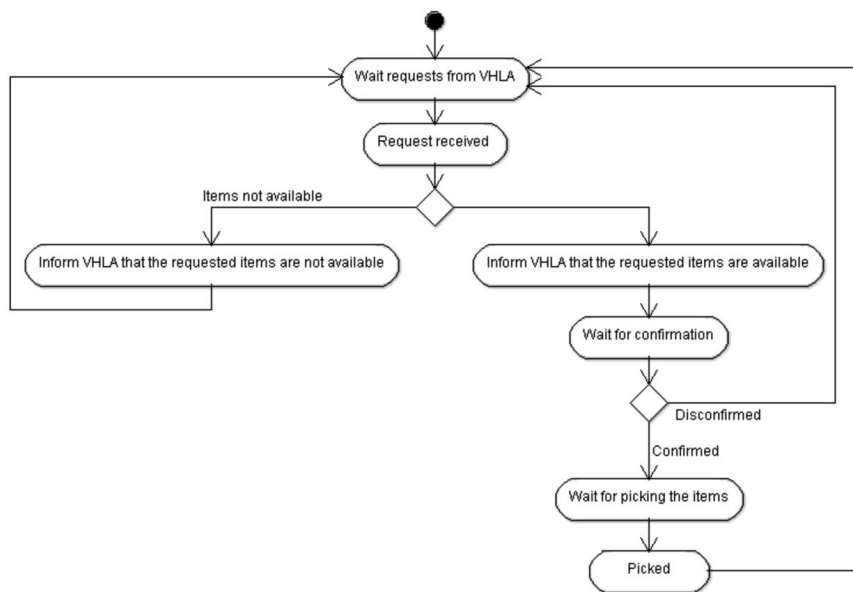*Figure 4.* Life cycle of a request receiver agent.

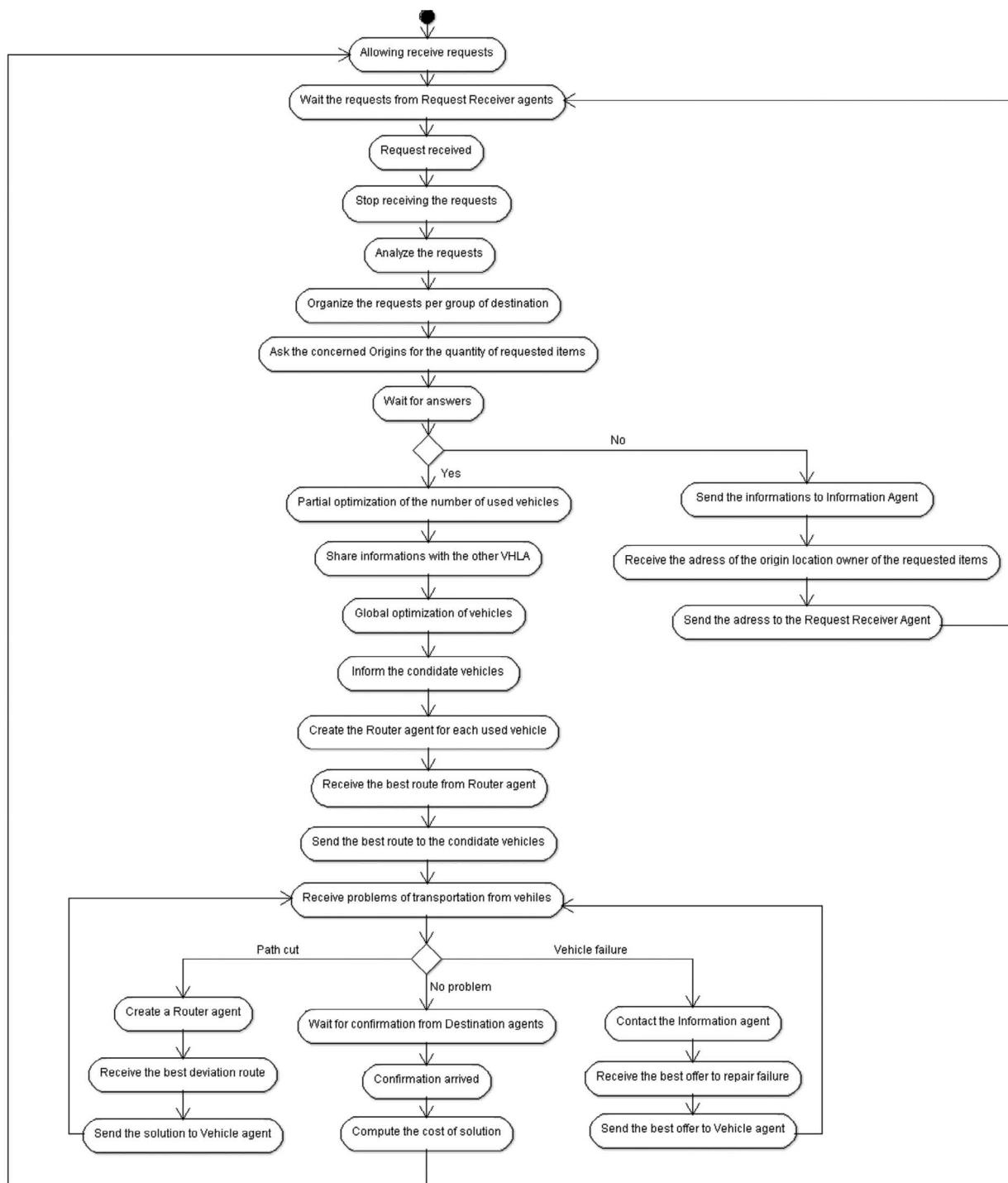*Figure 3.* Life cycle of an origin location agent.

Figure 5. Life cycle of a vehicle home location agent.

agent. For this, it enters in a contract net protocol with the corresponding vehicle home location agent. In the case of a path cut, the vehicle agent deliverer informs the corresponding home vehicle location agent about the event in order to handle it and overcome to the problem. Finally, In the case of a vehicle failure, the vehicle agent deliverer diagnoses the failure and informs the corresponding vehicle home location agent about the results of its diagnostic and about the place's address where happened the failure. Figure 6 shows the life cycle of an information agent.
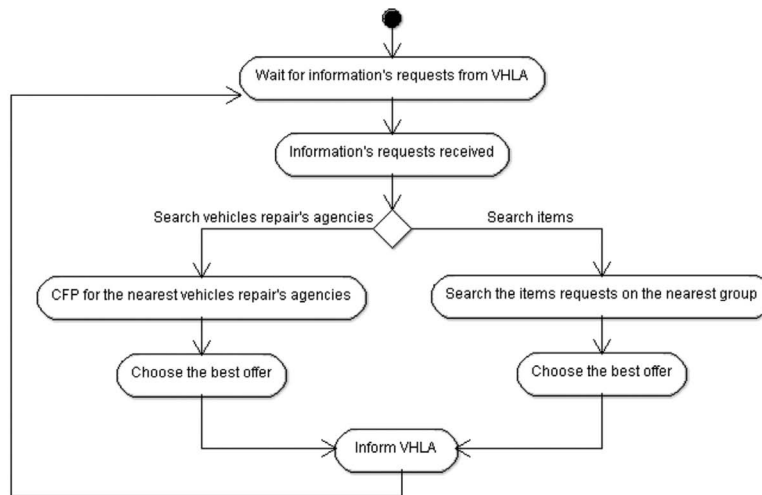
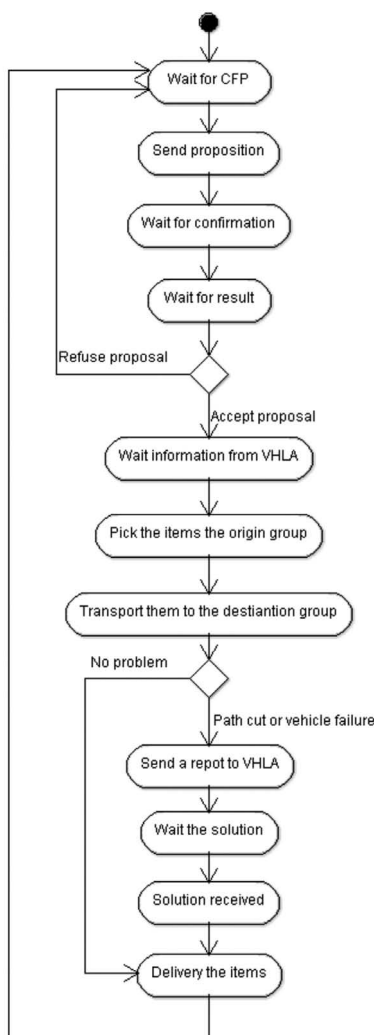*Figure 6.* Life cycle of an information agent.

Figure 7. Life cycle of a vehicle location agent.

- Groups of vehicle location agents: Each vehicle location agent has the capacity and the role, if asked, to deliver items to the concerned destinations using a path plan computed by a router agent. For this, it enters in a contract net protocol with the corresponding vehicle home location agent. In the case of a path cut, the vehicle agent deliverer informs the corresponding home vehicle location agent about the event in order to handle it and overcome to the problem. Finally, In the case of a vehicle failure, the vehicle agent deliverer diagnoses the failure and informs the corresponding vehicle home location agent about the results of its diagnostic and about the place's address where happened the failure. Figure 7 shows the life cycle of a vehicle location agent.

- A group of router agents: Router agents are created dynamically by vehicle home location agents (a router agent per a vehicle agent) and are responsible of searching the best paths to the concerned destination locations or to the group of destination locations, using a road map provided by the corresponding vehicle home location agent.

In the case of a path cut, a router agent created by the corresponding vehicle home location agent in order to find an alternative or a deviation path and informs its creator.

After computing paths, this agent is destructed by its creator. Figure 8 shows the life cycle of a router agent.
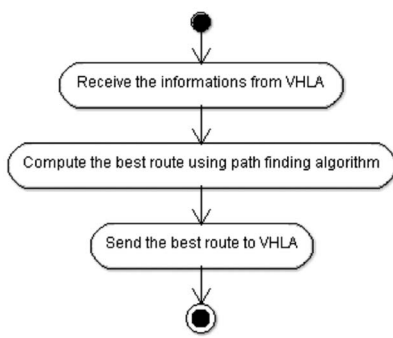
Figure 8. Life cycle of router agent.

5. Functionality of the Multi Agent System

We have a network of routes between the origins and the destinations. Origin location agents belonging to the different groups can send requests to their corresponding request receiver agents. Each home vehicle location agent is responsible of finding the optimal assignment of vehicles it manages in order to satisfy requests concerning the group of origins. When a requested item at an origin location is not available or available but with a quantity less than the ordered one, the request is redirected to another origin location of the same group or it is

rejected. In the sake of optimizing the number of vehicles used, a request could be also redirected to another group of origin location agents after cooperation between concerned vehicle home location agents. Once vehicle agents to be used are defined, each vehicle home location agent concerned creates a router agent in order to define the best path allowing delivering items to the corresponding destination location agents.

The multi agent system proposed allows the assignment of vehicles to be used by the different groups of origin locations in a parallel manner, also planed best paths are computed by the same manner and as well as, the execution of the transportation plan is done with the same manner allowing to attenuate the complexity of the problem. As mentioned above, requests can be satisfied with delays and the system aims to minimize the objective function given in the relation (18) above.

During schedule execution and in the case of a traffic jam or a path cut between two nodes due to bad weather conditions, a router agent is created dynamically by the corresponding vehicle home location agent and is invited to find the best path deviation.

Likewise, in the case of a vehicle failure, the concerned vehicle agent is invited to diagnose it
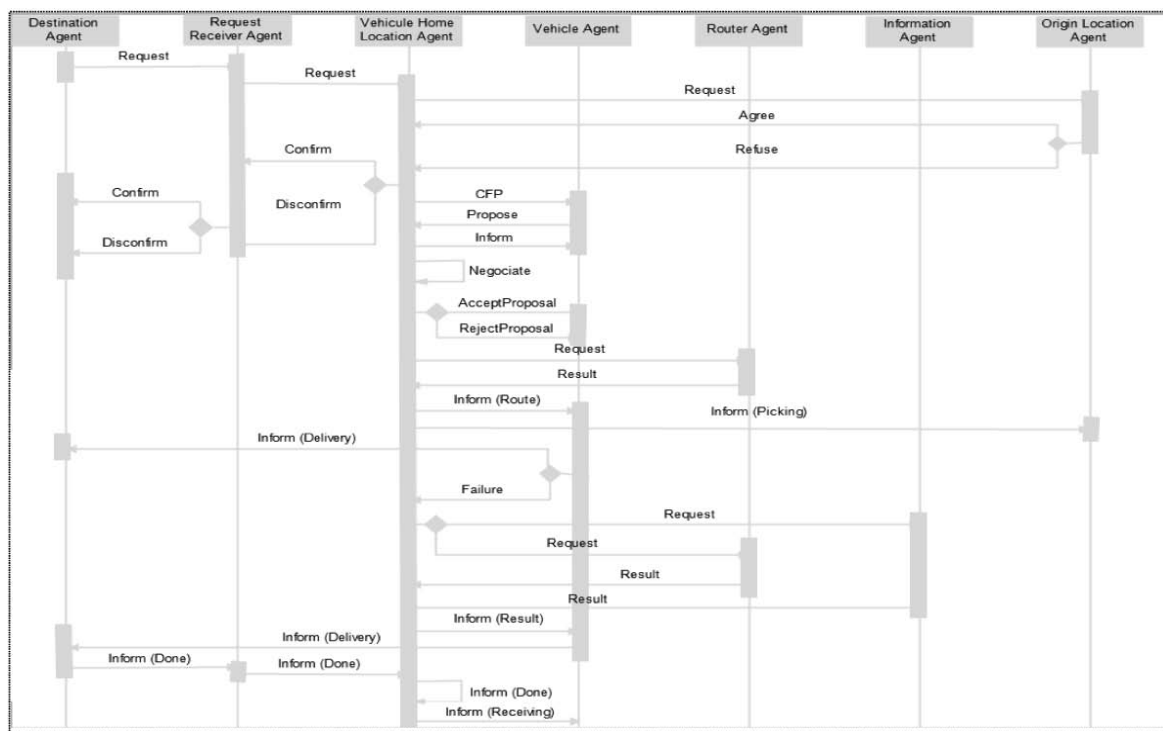

Figure 9. Sequence diagram of our multi agent system.

and transmit the result of the diagnosis and the address where vehicle failed to the corresponding vehicle home location agent.

This last interacts with the information agent attached to it in order to find an optimal proposal for troubleshooting the failure in the neighborhood of the failed vehicle. Figure 9 shows sequence diagram of our multi agent system.

6. Case Study

6.1 PDP Configuration

Suppose we have the following PDP configuration:

- 7 origin locations, $O = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}$ geographically grouped in two groups: $\{o_3, o_6, o_7\}$ and $\{o_2, o_4, o_5\}$

- A set T of available items and subsets T_j giving the items and the corresponding quantities of the origin location $O_j, j \in \{1, ..., 7\}$, $T = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, ..., m_{20}\}$, $T_1 = \{(m_1, 1000), (m_3, 1000), (m_5, 1000), (m_9, 1000)\}$; $T_2 = \{(m_2, 1500), (m_{11}, 2500), (m_6, 1500), (m_7, 1500)\}$; $T_3 = \{(m_9, 1200), (m_{14}, 1200)\}$; $T_4 = \{(m_{11}, 1400), (m_{12}, 2500), (m_{20}, 1400)\}$; $T_5 = \{(m_{11}, 1600), (m_{15}, 1000), (m_{12}, 1600), (m_9, 1600)\}$; $T_6 = \{(m_{16}, 1300), (m_{17}, 1300), (m_{18}, 1200)\}$; $T_7 = \{(m_2, 2000), (m_5, 2000), (m_{13}, 2000), (m_{11}, 2000)\}$

- 5 destination locations, $D = \{d_1, d_2, d_3, d_4, d_5\}$ geographically grouped in two groups: $\{d_1, d_3\}$ and $\{d_2, d_4, d_5\}$.

- A set of vehicles $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, a set of vehicle home locations, $H = \{h_1, h_2\}$. Each one contains vehicles as follows: $h_1 = \{v_1, v_2, v_3, v_4\}$ and $h_2 = \{v_5, v_6\}$.

- Vehicles' parameters are as follows (See relation (8) above):

  $Params(v_1) = (3000, 20, 140, 200, 500);$

  $Params(v_2) = (4000, 20, 180, 200, 800);$

  $Params(v_3) = (3000, 20, 130, 200, 500);$

  $Params(v_4) = (5000, 20, 120, 200, 1000);$

  $Params(v_5) = (4500, 20, 130, 200, 800);$

  $Params(v_6) = (2000, 20, 170, 200, 300);$

- Suppose that the parameters of the given items are as follows (See relation (9) above):

  $Params(m_1) = params(m_3) = params(m_5)$
  $\quad = params(m_7) = (0.010, 0.010);$
  $Params(m_2) = params(m_{11}) = params(m_6)$
  $\quad = params(m_7) = (0.010, 0.010);$
  $Params(m_9) = params(m_{14}) = (0.10, 0.10);$
  $Params(m_{11}) = params(m_{12}) = params(m_{20})$
  $\quad = (0.010, 0.010);$
  $Params(m_{11}) = params(m_{15}) = params(m_{12})$
  $\quad = (0.010, 0.010);$
  $Params(m_{16}) = params(m_{17}) = params(m_{18})$
  $\quad = (0.010, 0.010);$
  $Params(m_2) = params(m_5) = params(m_{13})$
  $\quad = params(m_{11}) = (0.010, 0.010).$

- And suppose we have four requests r_1, r_2, r_3 and r_4 with the following parameters (See relation (10) above):

  $Params(r_1) = (o_1, d_3, m_9, 500, current\_date,$
  $\quad current\_date + 2)$
  $Params(r_2) = (o_2, d_4, m_{11}, 1000, current\_date,$
  $\quad current\_date + 1)$
  $Params(r_3) = (o_1, d_2, m_9, 200, current\_date,$
  $\quad current\_date + 3)$
  $Params(r_4) = (o_4, d_2, m_{12}, 400, current\_date,$
  $\quad current\_date + 1)$

6.2 The Multi Agent System

After this description, our multi agent system will be composed of the following agents:

- 2 groups of origin location agents $\{o_1, o_3, o_6, o_7\}$ and $\{o_2, o_4, o_5\}$.

- 2 groups of destination location agents: $\{d_1, d_3\}$ and $\{d_2, d_4, d_5\}$;

- 2 groups of vehicle agents $\{v_1, v_2, v_3, v_4\}$ and $\{v_5, v_6\}$. Vehicle agents' role is the transportation of items from origin agents to the destination agents. Vehicle agents can also diagnoses their failures in order to estimate the repairing time.

- 2 vehicle home location agents: h_1 and h_2; h_1 manages the set $\{v_1, v_2, v_3, v_4\}$ of vehicle agents and h_2 manages the set $\{v_5, v_6\}$

of vehicle agents. Also h_1 is linked to the group of origin agents $\{o_1, o_3, o_6, o_7\}$ and h2 is linked to the other group $\{o_2, o_4, o_5\}$.

- 2 request agents: req_1 receives requests from the group $\{d_1, d_3\}$ and req_2 receives requests from the group $\{d_2, d_4, d_5\}$

- One or more router agents created dynamically by the corresponding vehicle home location agent after the affectation of vehicle agents to be used for the transportation of items from origins to destinations. Using the route network, each router agent has the role to compute the optimal route to be taken by a vehicle agent. We have so many router agents as vehicle agents.

6.3. Interactions Among Agents

Interactions among agents are as follows:

- Destination Agent d_3 sends the request r_1 to the request agent req_1;

- Agent req_1 notifies reception of r_1 to the destination agent d_3;

- Destination Agent d_4 sends the request r_2 to the request agent req_2

- Agent req_2 notifies reception of r_2 to the destination agent d_4

- Destination Agent d_2 sends the request r_3 to the request agent req_2;

- Agent req_2 notifies reception of r_3 to the destination agent d_2

- Destination Agent d_2 sends the request r_4 to the request agent req_2

- Agent req_2 notifies reception of r_4 to the destination agent d_2.

Simultaneously, agent req_1 and agent req_2 analyze received requests. Agent req_1 switches request r_1 to agent h_1, while agent req_2 switches request r_3 to h_1 while r_2 and r_4 are switched to agent h_2. See Figure 10.

In a parallel manner, agent's h_1 and agent h_2 analyze received requests. h_1 asks agent o_1 whether the quantity of 700 (500 + 200) of item m_9 is available and agent o_1 answers "yes" (because the quantity available is 1000 greater than 700). In the other hand, agent h_2 ask agents o_2 and o_4 if the quantities of 1000 and 400 are available about the items m_{11} and m_{12} respectively and agents o_2 and o_4 both answer "yes" because the quantity available of each item is 2500.

In a parallel manner, agents h_1 and h_2 try to find the number of vehicles to use to satisfy requests. Agent h1 remarks that two vehicles among candidate vehicles $\{v_1, v_2, v_3, v_4\}$ must be used because requests r_1, and r_3 concerns respectively destinations d_2 and d_3 belonging to two different groups of destination agents. In the other hand, agent h_2 will use only one vehicle since the two requests r_2 and r_4 concern only one group of destination agents.

Before the routing of these requests, vehicle home location agents h_1 and h_2; cooperate by sharing their information about requests, vehicle agents and items in the aim to optimize globally the number of vehicle agents to use. Effectively, request r_3 can be redirected to agent h_2 since item m_9 is available at the origin location agent o_5 with a quantity greater than
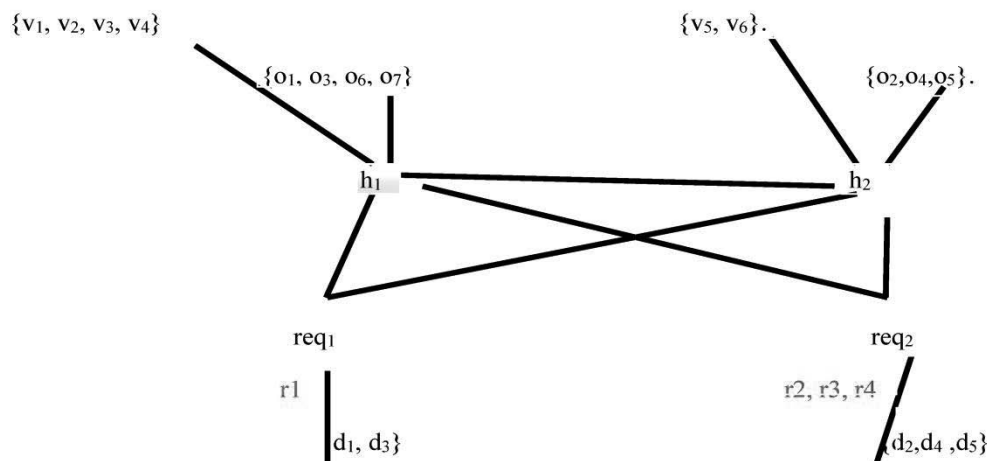

Figure 10. Pickup and Delivery case study.

the ordered one. The redirected request r_3 will be expressed as:

Params $(r_{3r}) = (o_1 \rightarrow o_5, d_2, m_9, 200,$
$\qquad current\_date, current\_date + 3)$.

Agent h_2 can then use only one vehicle to satisfy the requests r_2, r_4 and r_3 under the condition that the vehicle to be used has a capacity greater than or equal to the sum of loads (1000 + 200 + 400 = 1600), which is in our case verified for the two vehicle agents $\{v_5, v_6\}$ under the responsibility of h_2. So only two vehicles agents will be used, one to satisfy request r_1 and another one to satisfy requests r_2, r_3 and r_4. The candidate vehicle agents for satisfying r_1 are $\{v_1, v_2, v_3, v_4\}$ and the candidate ones to satisfy r_2, r_3 and r_4 are $\{v_5, v_6\}$.

In a parallel manner, agent h_1 and agent h_2, create each one a router agent in order to find the best routes giving a road map and a path finding algorithm. The first route is only a transportation route begins by o_1 and ends at d_3 whether the second one is composed of a pickup route going through the points $[o_2, o_4, o_5]$, a transportation route and a delivery route going through the points $[d_2, d_4]$. These computed best routes are used by home location vehicle agents h_1 and h_2 to select the best vehicle agents to use considering the objective function quoted above in relation (18).

If we suppose that the road map given by to the router agents is depicted by Figure 11. then: The outputs of the router agent are as follows:

The best route for the vehicle one is $[o_1 \rightarrow t_2 \rightarrow t_4 \rightarrow t_3 \rightarrow d_3]$ with a distance equals to (65 + 60 + 60 + 85) = 270 km.

The best route for the vehicle two is $(o_4 \rightarrow o_2 \rightarrow o_5 \rightarrow t_4 \rightarrow d_2 \rightarrow d_4)$ with a distance equals to (7 + 9 + 85 + 120 + 8) = 229 km.

This information is transmitted to the vehicle home location agents h_1 and h_2 by router agent$_1$ and router agent$_2$ respectively in order to use it in selecting vehicle one and vehicle two respectively. Agent h_1 launches a contact net protocol to receive proposals from candidates $\{v_1, v_2, v_3, v_4\}$
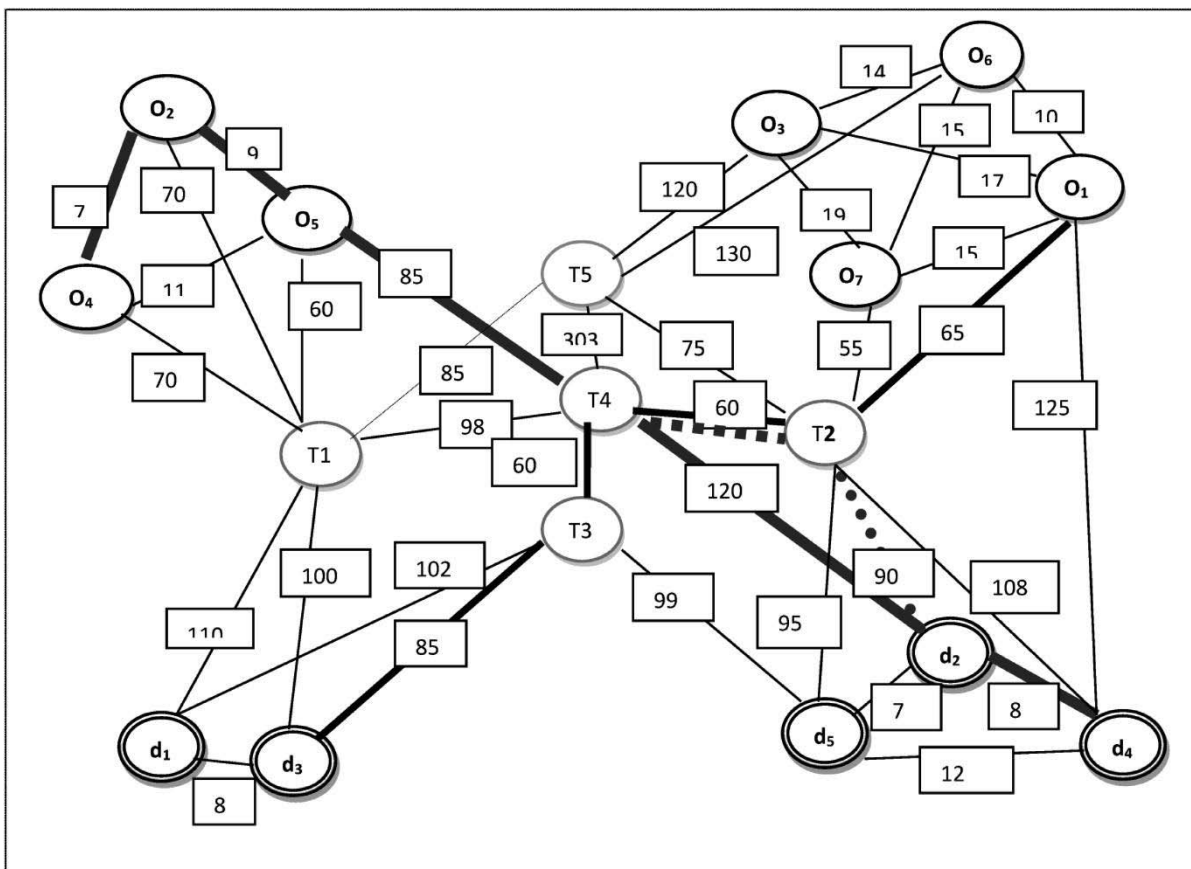

Figure 11. The road map between the origins and the destination locations showing optimal planned paths

for vehicle v_1 and v_6.

for the satisfaction of the requests r_1. Simultaneously, vehicle location agent h_2 launches also a contract net protocol to receive proposals from vehicle agents $\{v_5, v_6\}$ for the satisfaction of the requests r_2, r_3, r_4. The cost per km of vehicle agent v_1 and v_3 are equal and they are the minimum ones (500 DA/km) but the average speed of v_1 ((20 + 140)/2) is higher than v_3's one ((20 + 130)/2), then v_1 is selected. No matter about the delays because the distance (270 Km) can be travelled in less than one day. The vehicle home location h_2 does the same and selects from the candidate vehicles $\{v_5, v_6\}$ the vehicle v_6 which has the minimum cost per Km (200 DA Km). Likewise, no matter about delays because the distance 229 km can be travelled in less than one day.

6.4. Dynamic Events Handling

6.4.1. Path Cut Event Handling

We suppose that there is a path cut in the portion (t_4, d_2) during schedule execution by vehicle agent's v_1 and v_6, due to bad weather conditions. This path cut is a piece of v_6's path. In this case vehicle agent v6 informs vehicle home location agent h_2 about this event. This latter creates a router agent and ask it to find a shortcut (a substitute to the portion (t_4, t_2)), giving the road map and the whole v_6 path. The handling of this path cut by the router agent is by searching the best deviation path using a path finding algorithm. In our case this gives the sub-path (t_4, t_2, d_2) (see Figure 11) which is transmitted to its creator agent h_2 which sends a copy to the agent v_6 for execution. The revised path for the vehicle v_6 becomes: $(o_2 \rightarrow o_5 \rightarrow t_4 \rightarrow t_2 \rightarrow d_2 \rightarrow d_4)$ and the revised distance becomes equal to: 229 + (90 + 60 − 120) = 259 km. We note that, vehicle agent v_1 is not concerned by this event since this path cut is not a piece of its whole path.

6.4.2. Vehicle Failure Event Handling

Suppose that a failure happened to the vehicle agent v_1 at any point of its planned path. This last is invited to diagnose its failure and estimate the necessary time to troubleshoot (remove) the

failure. The failure removing cost and the estimated time are transmitted to the home location agent h_1 in order to update the delivery cost.

6.5. Cost of the Solution

The cost of the solution (set of vehicles used and their generated cost) is computed using the relations 11 to 18 and regarding given case study data. The set of vehicles to use is defined after the computing of the optimal routes and also estimated delays are computed before the beginning of the satisfaction of the requests, while additional expenses due to dynamic events, like path cut or vehicle failure, are considered also. Table 1 shows a configuration ($\{v_1, v_6\}$) among 08 possible configurations ($\{v_1, v_2, v_3, v_4\} \times \{v_5, v_6\}$) and the computation of the corresponding delivery cost by home vehicle agents h_1 (use v_1) and h_2 (use v_6). Figure 13 gives the best delivery cost revision algorithm while Table 2 shows the same configuration $\{v_1, v_6\}$ but in the case of dynamic events of path cut and vehicle failure and gives the computation of the revised delivery cost.

We note that the computation of the cost of the solution is done in a parallel manner so that each vehicle agent computes its proper cost and all the individual costs are finally gathered to obtain the cost of the solution. We give in Figure 12, our searching algorithm for the best solution.

7. Conclusion

In this paper we've proposed a multi agent architecture for dynamic pickup and delivery problem where agents are organized by groups and roles. The aim of this MAS architecture is searching collaboratively an optimal global solution for the dynamic pickup and delivery problem by optimizing a multi objective function. The optimization process passes by two phases which are requests grouping optimization and vehicle routing optimization. The first phase allows defining the optimal set of vehicles, and for this task we use a contract net protocol for each group of vehicles initiated by the corresponding vehicle home location agent. In the second phase, called also the routing phase, a router agent is created for each selected vehi-

Find_vector_number_of_vehicles (received_requests, available_items, available_vehicles);
{** Done in a parallel and cooperative manner by vehicle home locations agents ***}
{* the elements of this vector indicate the number of minimum vehicles to use for each origin location, **}
    {** This vector is under the form VMNV = (nb_veh_o1, nb_veh_o2, ..., nb_veh_on) ***}
Find all the best routes (road_map, origin locations<s>, destination location<s>);
{*** Done in a parallel manner by router agents. Each route corresponds to one vehicle to use ***}
 {** computing of the best delivery cost. Done in a parallel manner by vehicle home location agents **}
Min_delivery_cost := max
{*** max is a maximum delivery_cost to support ***}
repeat
{*** Find a new possible configuration of VMNV among the available vehicles in origins ***}
 Actual_config := FindConfiguration (VMNV);
  Delivery_cost_all := 0
  for each vehicle_used v_i in Actual_config
   Compute delivery time (v_i) {** using relation 11 above **}
   Compute delay (v_i)   {** using relation 12 above **}
   Compute penalty_cost (v_i) {** using relation 13 above **}
   Compute transport_cost (v_i) {*** compute delivery_cost using relations 14, 15, 16 & 17 above ***}
    Compute delivery_cost (v_i)
    Delivery_cost_all := delivery_cost_all + delivery_cost (v_i)
  end for
if Delivery_cost_all < Min_delivery_cost_all **then** Min_delivery_cost := Delivery_cost_all;
Best_config := Actual_config.
end if
until no_possible_configuration
end Best_DeliveryCost

Figure 12. Best cost delivery algorithm.

cle in order to compute an optimal path using a path planning algorithm. Our proposed multi agent system handles not only static pickup and delivery problem but also the dynamic one. The two sources of dynamicity handled by our architecture are path cut and vehicle failure. The proposed architecture is supported by a formal framework for pickup and delivery problem taking in the account dynamic events. We have also proposed two algorithms computing

Table 1. Computation of the delivery cost ($\alpha = 0.7, \beta = 0.2, \delta = 0.1$) in relation (18) and no dynamic events

Veh. (reqs)	Dist. (km)	Transp. Cost (Dinar)	Delivery time (Hour)		Delay (Hour)	Penalty cost (Dinar)	Delivery cost (Dinar)
v_1 $\{r_1\}$	270	270*500 =135000	2*(500*0.01) + (270/80 + 0.5) = +13.5 h		0.00	0.00	135000
v_6 $\{r_4, r_3, r_2\}$	229	229*300 =68700	2*(400*0.01) + 221/95 + 0.5 + 2*(200*0.01) = 8 + 4 + 2.33 = 14.33 h	01 day + 10.33 h	10.33	10330	68700 + 10330
			2*(1000*0.01) + 8/95 = 20 + 0.08 = 20 h				
					Total Delivery_ cost		0.7*(135000 + 68700) + 0.2*10330 + 0.1*0.00 = 144656

if Dynamic_Event = Cut_path **then**

Alternative_P := Find Alternative_path (road_map, OriginPoint, DestinationPoint);

{*** Done in a parallel manner by router agents if there are several cut paths. If a cut path is shared by several vehicle agents then it is hold by only one router agent, and the information is therefore communicated by the corresponding vehicle home location agent to the concerned vehicle agents and also to all other vehicle home location agents***}

Time_Dev := Distance (Alternative_P) / Average_speed (v_i);

end if

if Dynamic_Event = Vehicle_Failure

Time_repair := Dignose_to_estimate_time (v_i);

end if

{*   Other dynamic events not included here, to consider in the future   *}

{** computing of the revised best delivery cost. Done in a parallel manner by vehicle home location agents for each concerned vehicle **}

    Compute Delivery_time (v_i)   {** Using relation 11 above **}
    Compute Delay (v_i)   {** Using relation 12 above **}
    Compute Penalty_cost (v_i)   {** Using relation 13 & 17 above **}
    Read Repair_cost (v_i)
    Compute Transport_cost (v_i)   {** Using relations 14 and 16 and relation above **}
        {** Compute Delivery_cost using the relation 15 above **}
        Revised_delivery_cost (v_i) := Compute Delivery_cost (v_i)

end Revised_Best_DeliveryCost

Figure 13. Best delivery cost revision algorithm.

Table 2. Revised computation of the delivery cost ($\alpha = 0.7$, $\beta = 0.2$, $\delta = 0.1$) in relation (18) with two dynamic events: vehicle failure for v_1 and path cut for v_6.

Veh. (reqs)	Dist. (km)	Transp. Cost (Dinar)	Repair. Cost (Dinar)	Time_ repair (Hour)	Time_ Dev. (Hour)	Delivery time (Hour)	Delay (Hour)	Penalty_ cost (Dinar)	Delivery cost (Dinar)
v_1 $\{r_1\}$	270	270*500 =135000 + 4000	4000	1 Day (24 h)	0 h	2*(500*0.01) + 270/80 + 0.5 + 24 = 1 day + 13.5 h	0.00	0.00	135000 + 4000
v_6 $\{r_4, r_3, r_2\}$	(101 + 150 + 8) = 259	259*300 =77700 + 0.00	0.00	0 h	150/95	2*(400*0.01) + 101/95 + 150/95 + 0.5 + 2*(200*0.01) = 8 + 4 + 2.95 = 13 h 1 day + 9 h 2*(1000*0.01) + 8/95 = 20 + 0.08 = 20 h	9 h	9*1000 = 9000	77700 + 9000
								Revised Total Delivery_ cost	0.7*(135000 + 77700) + 0.2*9000 + 0.1*4000 = **153890**

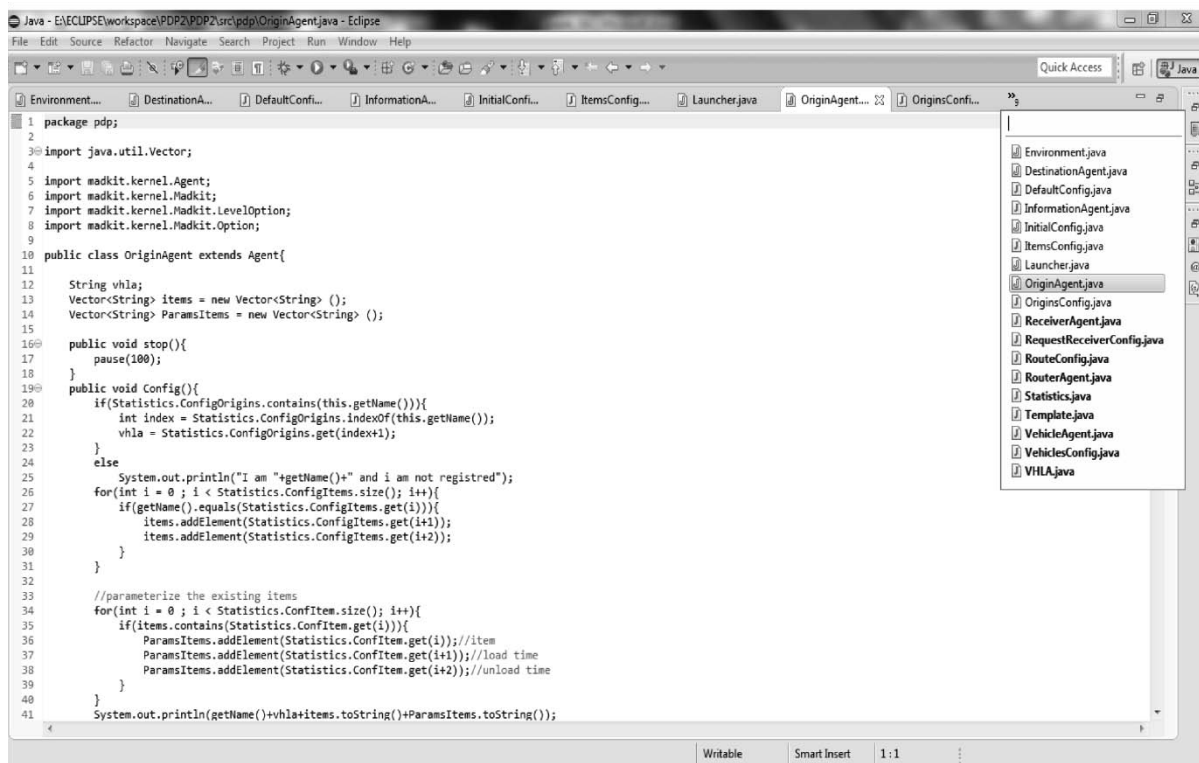*Figure 14.* A screen snapshot of destination location agent's code.

Figure 15. A screen snapshot of origin location agent's code.

this optimal solution, the first in the absence of dynamic event and the second in the presence of path cut and/or vehicle failure. As depicted by these algorithms, almost all tasks in searching such optimal solution are done in a parallel manner by the concerned agents, which should attenuate considerably the complexity of the studied problem. Actually, we are testing our approach on MADKIT multi agent platform based on AGR (Agent Group Role) model

Figure 16. A screen snapshot of vehicle home location agent's code .

[15], [16] using Aspect oriented programming technology. Figure 14, Figure 15 and Figure 16 show screen snapshots of the written code. However, our work remains opened to future contributions like taking into account other dynamic events like considering requests cancelling and/or requests updating. Another major future work is to consider the case when we have serious vehicles' failures and then we are invited to substitute the failed vehicles.

References

[1] C. Brian, "Multi Agent Pickup and Delivery Problem with Transfers", Ph.D. dissertation, The robotic institute of Carnegie Melon University, Pittsburgh, Pennsylvania, 2014.

[2] K. C. Jaroslaw *et al*., "Multi-agent Approach to Dynamic Pick-up and Delivery Problem with Uncertain Knowledge about Future Transport Demands", *Int. J. of Fundamenta Informaticae*, vol. 71, no. 1, pp. 27–36, 2006.

[3] J. Ferber *et al*., "From Agents to Organizations: An Organizational View of Multi-agent Systems", In International Workshop on Agent-Oriented Software Engineering, Springer, Berlin, Heidelberg, 2003, pp. 214–230.

[4] Z. B. Rubinstein and S. F. Smith, "Dynamic Management of Paratransit Vehicle Schedules", in *Proceedings of the International Workshop on Scheduling and Planning Applications*, 2011, vol. 124.

[5] J. F. Cordeau and G. Laporte, "The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms", *Quart. J. of the Belgian, French and Italian Operations Research Societies*, vol. 1, no. 2, pp. 89–101, 2003.

[6] K. Manal and A. Anjali, "An Intelligent Multi-agent Based Model for ollaborative Logistics Systems", *Science Direct, Transportation Research Procedia*, vol. 12, pp. 325–338, 2016.

[7] M. Simon *et al*., "A Multi Agent based Cooperative Approach to Scheduling and Routing", *J. of European Operational Research*, vol. 254, no. 1, pp. 169–178, 2016.

[8] F. L. Bellifemine *et al*., "Developping Multi Agent Systems with JADE", Wiley & sons, 2007, ISBN 0470057475.

[9] X. Wang and G. Zeng, "A Dynamic Planning Method for Mobile Agent Implementation based on Service Recommendation", *Int. J. of Applied Mathematics & Information Sciences*, vol. 8, no. 3, pp. 1247, 2014.

[10] S. J. Anderson *et al*., "Constraint-based Planning and Control for Safe, Semi-autonomous Operation of Vehicles", in Intelligent Vehicles Sympo-

sium , 2012, pp. 383–388.
http://dx.doi.org/10.1109/IVS.2012.6232153

[11] I. Muscalagiu *et al*., "Enhancing DisCSP-Netlogo from Simulation to Real-execution of Agents in Distributed Constraints", *Procedia Computer Science*, vol. 35, pp. 261–270, 2014.

[12] P. Castejon, *et al*., "Multi-agent Planning by Distributed Constraint Satisfaction", in *Proceedings of CAEPIA*, 2015, pp. 41–50.

[13] A. A. Amanatiadis *et al*., "A Multi-objective Exploration Strategy for Mobile Robots Under Operational Constraints", *Int. IEEE Access Journal*, vol. 1, pp. 691–702, 2013.
http://dx.doi.org/10.1109/ACCESS.2013.2283031

[14] A. Ahmadzadeh *et al*., "Multi-vehicle Path Planning in Dynamically Changing Environments", in *Proc. IEEE Int. Conference on Robotics and Auto-mation, ICRA'09*, 2009, pp. 2449–2454.
http://dx.doi.org/10.1109/ROBOT.2009.5152520

[15] J. Ferber *et al*., "From Agents to Organizations, an Organizational View of Multi-Agent Systems", Int. Workshop on Agent-Oriented Software Engineering, Berlin, Heidelberg, 2003, pp. 214–230.

[16] P. Vincent and H. Gäelle, "IUT Monpellier", Dep. of Comput. Science, Monpellier 2 University, Tech. Rep. 2011.

Contact address:
Tahar Guerram
University Larbi Ben M'hidi of Oum El Bouaghi
DISE Team of Relac(s)2 Rsearch Laboratory
Departement of Mathematics and Computer Science
Algeria
e-mail: tahar.guerram@gmail.com

TAHAR GUERRAM is assistant professor at the Departement of Mathematics and Computer Science of the University of Larbi Ben M'hidi of Oum El Bouaghi. He received the PhD degree in computer Science in 2011 from the University of Constantine (Algeria). His research interests Include Distributed Artificial Intelligence and Complex Systems.