University of Leuven (KUL)
Departement of Computer Science

Multi-Agent Systems

Project 2011-2012

By:
Karsten Daemen
Michiel Staessen

For:
prof. Tom Holvoet
dr. Robrecht Haesevoets

6 juni 2012

1 Introduction

This paper will outline a *Delegate Multi-Agents System (DMAS)* solution that uses *Ant Colony Optimalisation (ACO)* techniques to solve the dynamic *Pickup and Delivery Problem (PDP)*. The performance of this solution will be compared with two other classic *Multi Agent Systems (MAS)*, *ContractNet* and *Gradient Field*.

The considered PDP assumes that a truck can only transport one package at the time and that new packages can randomly be added to the system. The infrastructure will also remain the same and congestion on road segments will not occur.

All of the agents in the presented DMAS solution are simulated in a virtual environment (e.g. a private cloud). It is not very affordable or even practical to equip every pickup and delivery location (in the physical environment) with a communication device. These locations can change a lot and these devices are necessary because agents on these locations have to be able to communice with each other. Because the whole DMAS is run on the virtual environment, truck drivers (in the physical environment) will only need some kind of GPS device to receive the planned path from the simulation in the virtual environment.

The purpose of this paper is only to outline the implementation of an ACO based DMAS for PDP. Therefore, the reader is supposed to have a decent knowledge of *Multi-Agent Systems (MAS)* and the applications of *Ant Colony Optimalisation (ACO)* techniques in *Delegate Multi-Agent Systems (DMAS)*.

2 Approach

The DMAS solution we developped defines two agents, *Package Agents* and *Truck Agents*. All Package Agents have a corresponding package and are located on the pickup location of that package. Truck Agents control a corresponding truck and move along with the truck. Both agents also own a *pheromon table*. These tables can hold multiple paths (list of Package Agents in a certain order) and a corresponding pheromon (heuristic value).

Package Agents and Truck Agents communicate with each other by sending ants. These are a sort of smart messages.

There is also a *"forwarder"* placed in each delivery location of a package. These *"forwarders"* will send all received ants to the Package Agent corresponding with the package for the local delivery location. Because of their limited intelligence, forwarders are not considered to be agents.

Like most DMAS based on ACO, three type of ants will be used. They will be used in the following way:

Feasability Ants These ants will be periodically broadcasted over a certain radius from the Package Agents to the forwarders (delivery locations). Their main purpose is to inform other Package Agents wich Package Agents are in a certain radius near their delivery location.

Exploration Ants These ants will be sent from the Truck Agents to discover and evaluate a path of Package Agents. They will update the pheromons table in the Package Agents during their return over the discovered path.

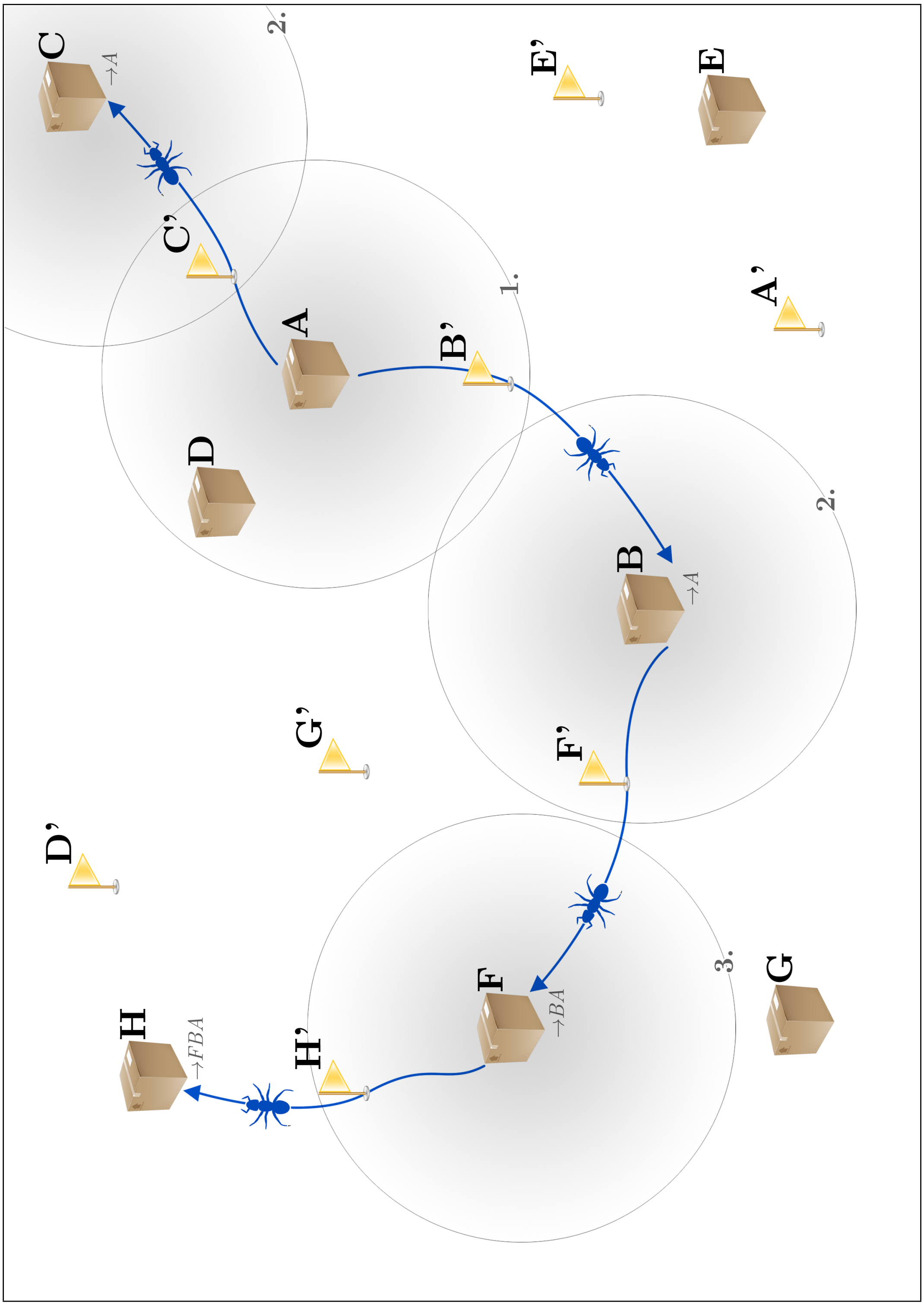
Intention Ants These ants will be sent from the Truck Agents over the path of Package Agents that they are planning to follow. These ants will decrease the pheromon values for that path in the visited Package Agents in order to discourage other agents to follow this path.

How agents find an optimal path by using the pheromones and communication over ants is further explained in the following scenarios.

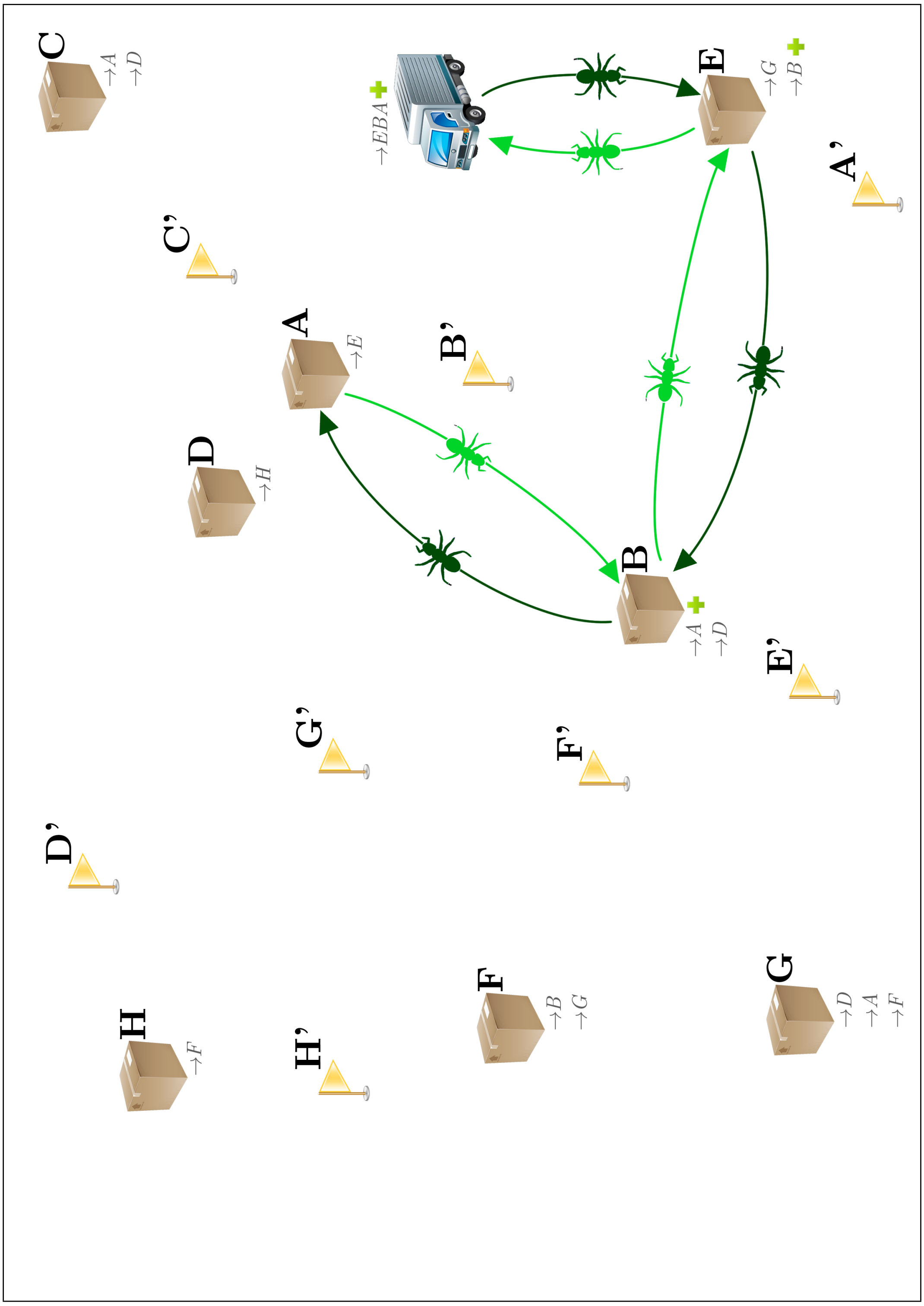
Scenario I: Broadcasting of Feasibility Ants

When Package Agent A (depicted as a brown box 'A' in Figure 1) is added to the simulation it will broadcast a Feasibility Ant with origin *"A"* in a certain radius to all delivery locations (forwarders, depicted as a yellow flags in Figure 1). Forwarders C' and B' receive the feasibility ant and send it to their corresponding Package Agents C and B. These Package Agents now know that after their packages are delivered, package A is near. After C and B have added path *"A"* to their tables with a default pheromon value, they will change the origin of the Feasibility Ant to "CAänd "BA"respectively and broadcast it again. This time, on
This process goes on until the max number of hops (= number of times broadcasts

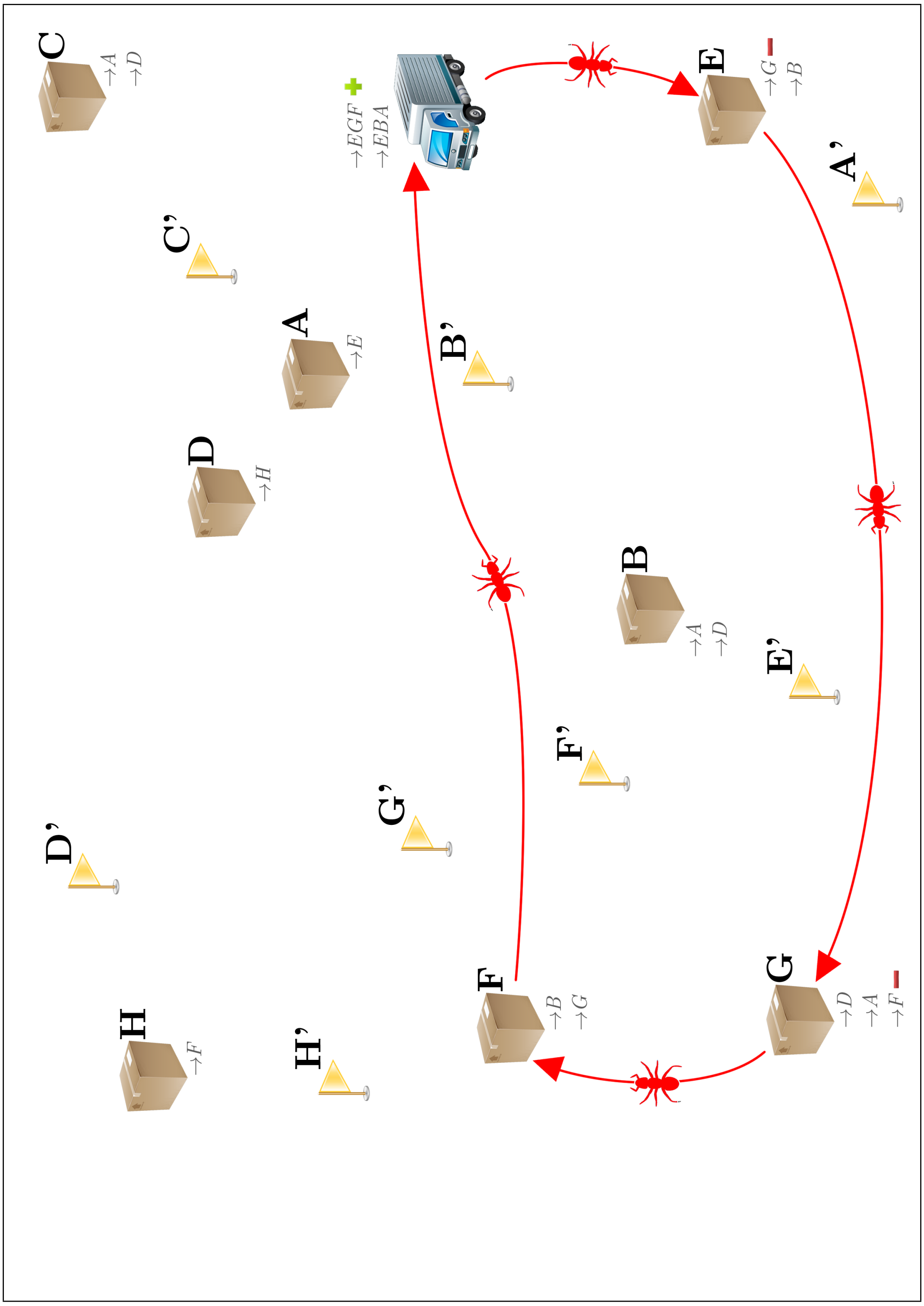
The max number of hops for Feasibility ants is in our application 1, i.e. only



Figuur 1: Example Scenario I (2)

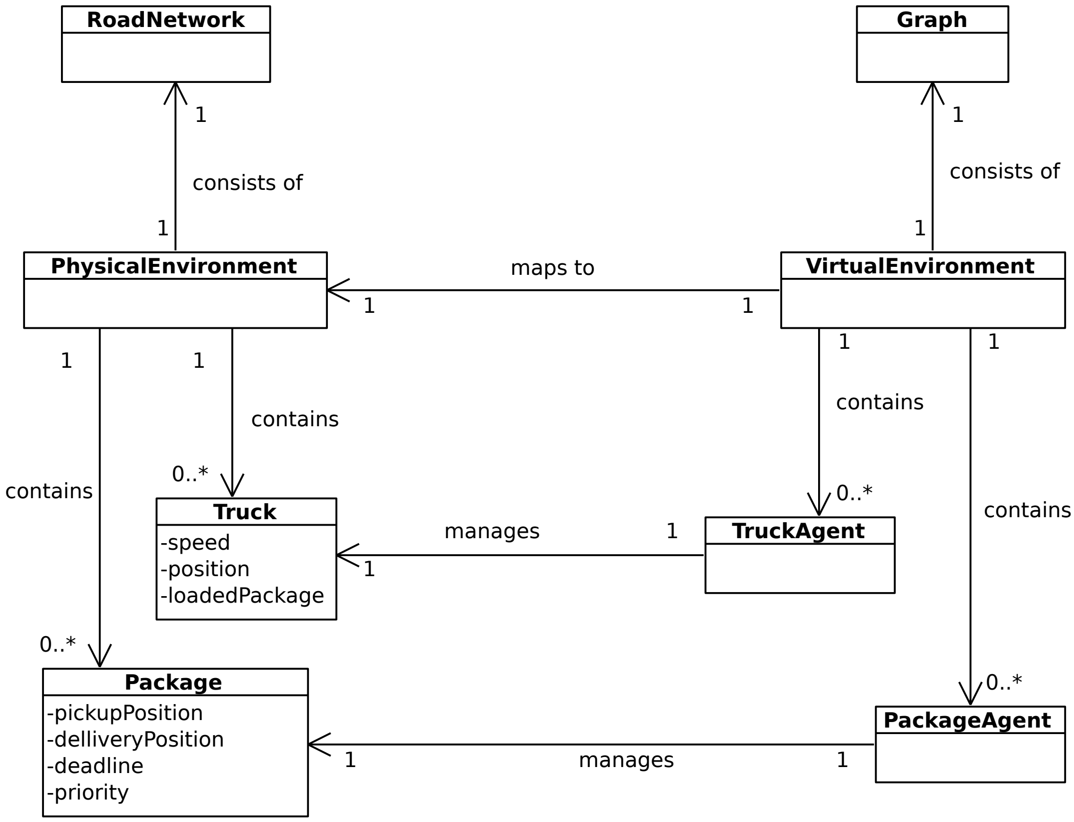


Figuur 2: Exploration Ants (example scenario 2)

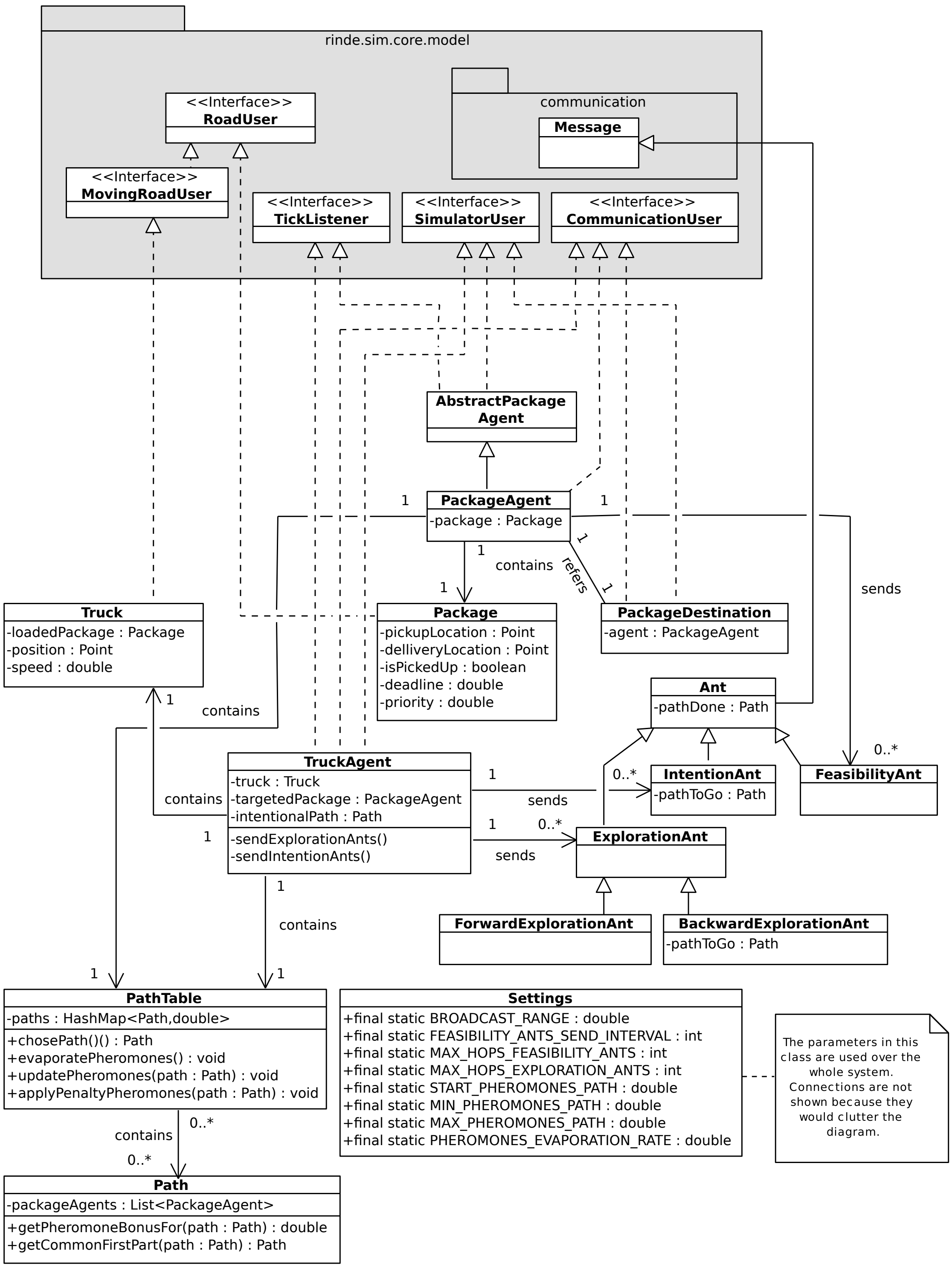


Figuur 3: Intention Ants (example scenario 3)

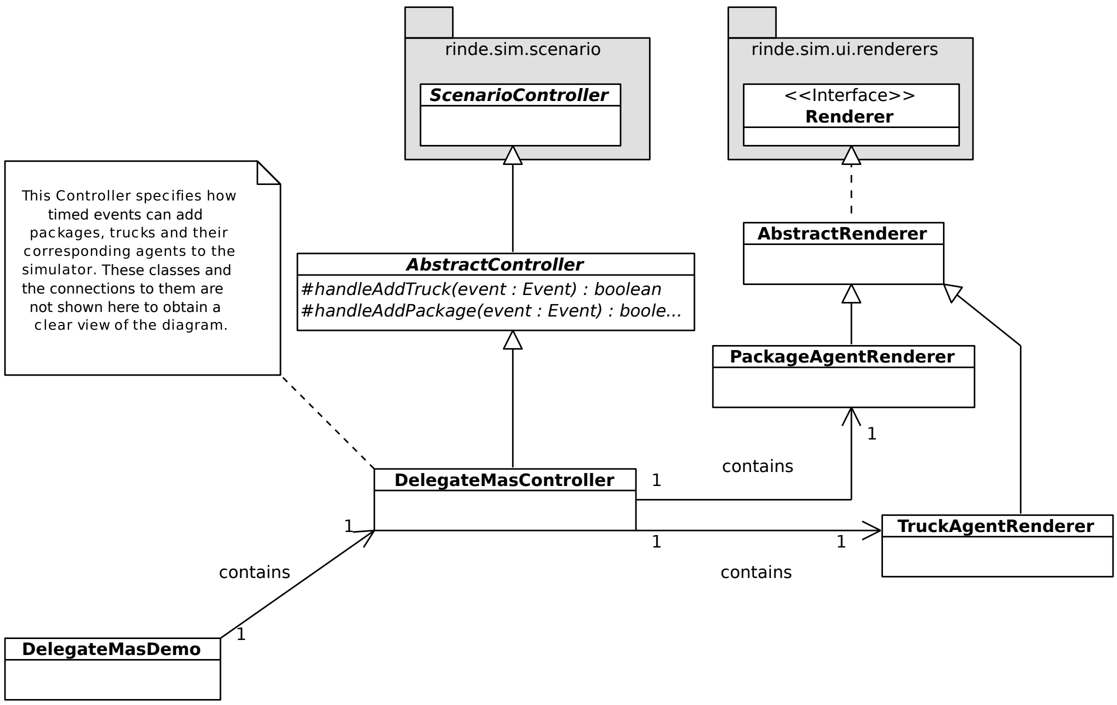
3 Development



Figuur 4: Class Diagram (part I)



Figuur 5: Class Diagram (part I)



Figuur 6: Class Diagram (part II)

4 Evaluation

In order to evaluate the DMAS solution, it is compared to two reference solutions: Gradient Field and Contract Net. We will describe them in a nutshell.

Gradient Fields are based on electrical fields: packages and trucks have opposite charges and will thus attract each other. However, trucks will repel other trucks because they are able to move and have equal charges. This will make sure that the work is divided among all trucks.

In a Contract Net, package agents will broadcast the position of their packages. (Some) Truck agents will receive the broadcasted message and repond with an offer to pick up the package. The package agent will then evaluate all the offers it received and inform the senders whether they won the offer or not. When a truck receives an accepted proposal, it can still decide that there is a better alternative. In this case, the package agent is informed of a failure. The contract is now broken and the package will again broadcast the position of its package.

4.1 Experiments

As explained in 2, the DMAS solution will try to plan an optimal route. As a consequence, we are particularly interested in the performance of this system. We test this in a number of ways.

First, the overall performance is measured by counting the number of delivered packages in a fixed time window.

Second, we track the time that passes between the creation of a package and the pickup. This can serve as an indication for the responsiveness to dynamism.

Third, all packages should be delivered within a given time window. Therefore, the lateness for every delivery is logged.

Finally, to evaluate the efficiency of the chosen routes, the total distance travelled by trucks is also measured.

All this information is gathered during a fixed-time scenario in which a number of trucks start with an initial amount of packages. In order to increase the statistical relevance, this scenario is executed 10 time with different random seeds. The result is the average of the 10 subresults. In the scenario, there are initially 3 trucks and 12 packages. Every time a package is delivered, a new one is added. This will prevent the trucks from running out of packages and will not flood the system with too much packages either. This scenario is used for every experiment.

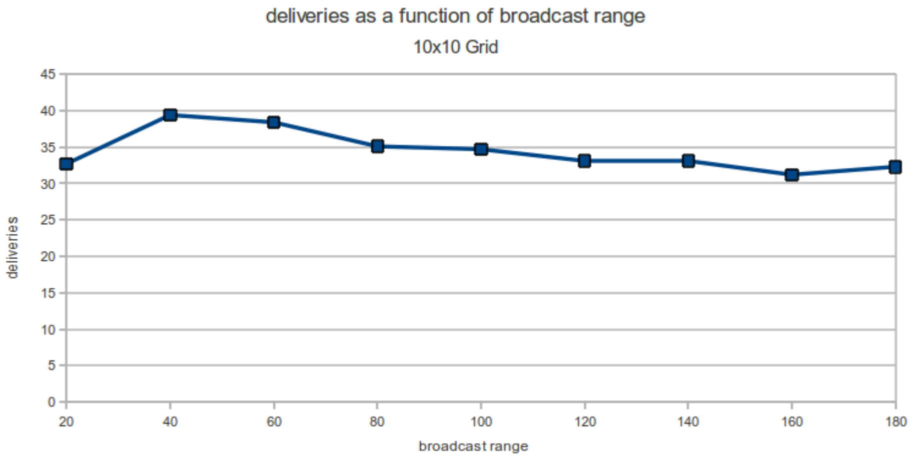
We will test this on the map of Leuven, which is a realistic environment, but we also test it on a 10×10 grid, which is a more ideal environment but not very realistic.

Because of the optimal path planning, we expect to see a high thoughput (delivered packages) and a low value for total distance travelled.

4.2 Comparison

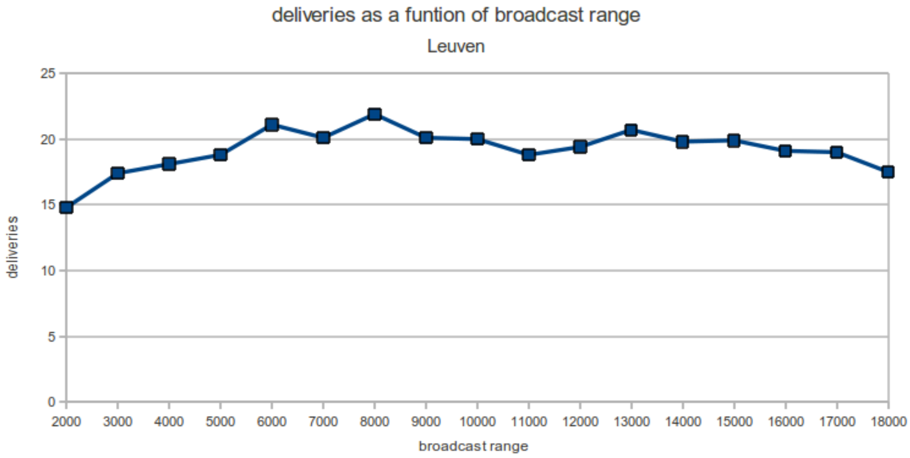
Before starting the actual comparison, the optimal parameters for the Delegate MAS are determined. In particular, the broadcast radius should be optimized. When a truck has no current path, it moves through the map randomly. If a package never receives a broadcast, it will never answer that broadcast. As a result, the truck will keep on moving randomly and travel many useless miles...

Nodes in the grid are 20 virtual distances apart and a broadcast range between 20 and 180 is tested. The results are shown in 4.2. The best broadcast value seems to be 40 but it does not seem very significant.



Figuur 7: Average number of deliveries in function of broadcast range in a 10×10 grid

A similar setup is used for Leuven. The "diameter" of Leuven is a lot larger than that of the 10×10 grid, though. Therefore, increments of 1000 distance units are used. The results are shown in figure 4.2. Here, the best value seems to be 8000 but as before, this is not a very significant result.

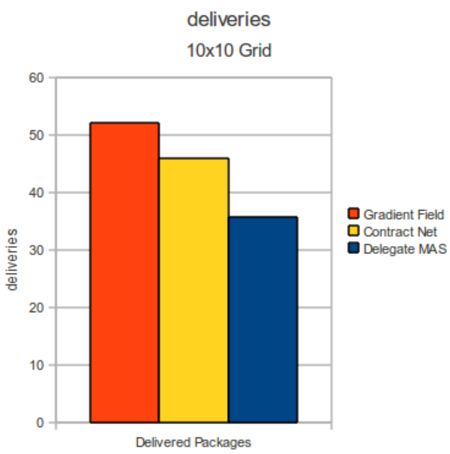


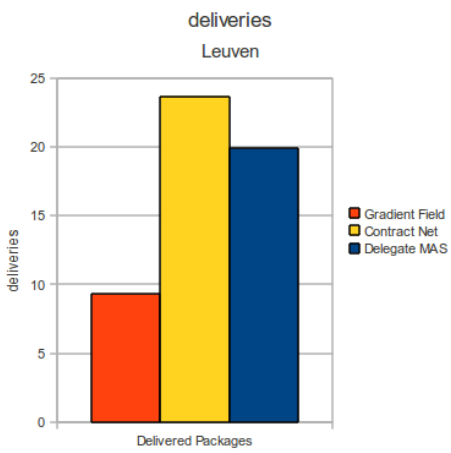
Figuur 8: Average number of deliveries in function of broadcast range in Leuven

Now that the "best" parameters are determined, the actual comparison can take place. As before, the same scenario is run 10 times for each strategy (Gradient Field, Contract Net and Delegate MAS). From these 10 runs, the average result is calculated.

For the grid, we see that the Gradient Field offers the best solution, followed by the Contract Net strategy and the Delegate MAS solution is the least performing (see figure 4.2). However, in a real world environment like Leuven, the Gradient Field solution performs the worst. The reason for this is that the trucks easily enter a deadlock state. This can however still be tweaked by adapting the field strength weights. It is also possible to remember the last visited nodes and determine whether a truck is going back and forth in a (dead end) street.

It is remarkable that the Delegate MAS cannot outperform the Contract Net implementation as we would expect it to do so.

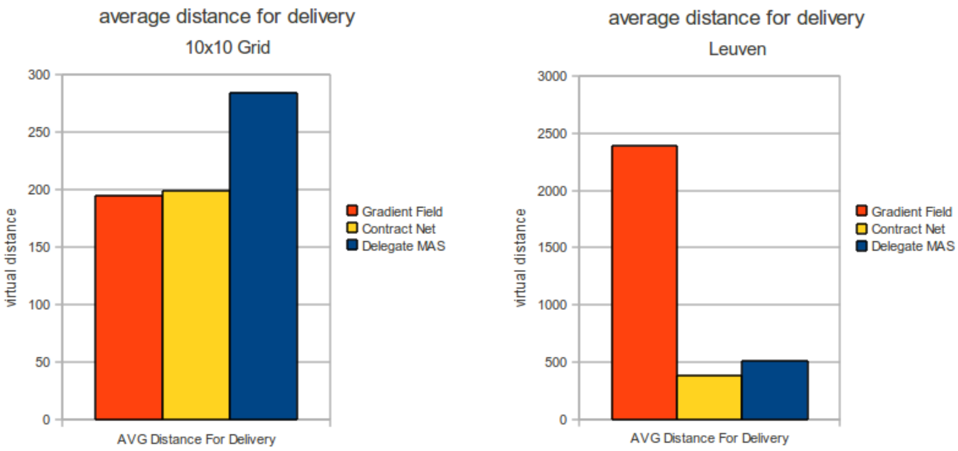




Figuur 9: Average number of deliveries for every tested strategy

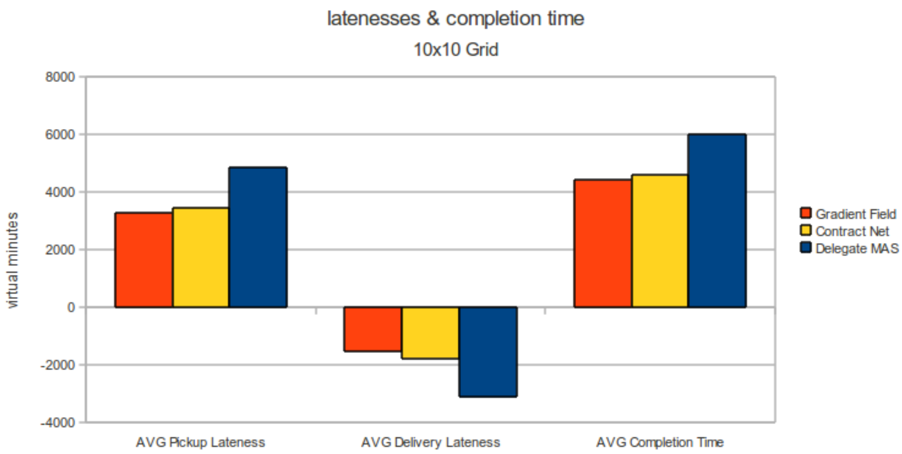
When comparing average distance for a delivery job (see figure 4.2), there are some obvious remarks. In the grid, a deadlock will never occur, the travelled distance is thus optimal. In a real world environment though, deadlocks do occur and the truck travels lots of miles...

When comparing Contract Net to Delegate MAS, the average distances are similar. With a little more parameter tweaking, the Delegate MAS solution can possibly outperform the Contract Net solution. However, this solution can also be tweaked.



Figuur 10: Average distance for a delivery job for every tested strategy

Finally, the average latenesses and completion times are compared. The results for the grid are show in figure 4.2, the results for Leuven are shown in figure 4.2. These graphs only show statistics for the packages that are actually delivered. For instance, it would be impossible for the Gradient Field to have the lowest delivery lateness while being deadlocked. Other packages would be waiting too.



Figuur 11: Average latenesses and completion time for every tested scenario for the grid

Figuur 12: Average latenesses and completion time for every tested scenario for Leuven

As you can see, all packages are delivered too late. This is due to the fact that there are too few trucks to handle this many packages in the used time window. We could increase the deadline, but it won't change the ranking of the strategies. You can see that if the trucks manage to deliver packages in a gradient field, they are delivered the fastest. Next in line is the contract Net implementation and last is the Delegate MAS solution.

4.3 Critical Reflection

First of all, the system manages to get the task done. However, it still needs a lot of tweaking.

A huge advantage to this system should be that it can planq ahead, whereas the (naive) Contract Net and the Gradient Field solution in particular do not/cannot plan ahead. Using heuristics, the delegate MAS solution can find an optimal path throughout the system.

One of the disadvantages about this system is that it will not always find packages at a given moment. Sometimes, it is required to navigate randomly until packages nearbycan "hear" the truck agent. One way to solve this problem would be to periodically increase the broadcast radius of truck agents (up to a maximal range) when no packages are found in a certain time window.

Another way to solve this problem could be to combine multiple solutions into one hybrid solution. When no packages are found, a Gradient Field could be used to spread the agents accross the physical environment. This will increase the coverage of the map and hence the chance for packages to be heard by trucks.

As we have seen, traditional systems outperform our delegate MAS solution. This can probably be optimized by using a better heuristic. To obtain such a heuristic, more work need to be done, obviously.

There is however potential and it will scale better than a Contract Net solution. At this moment, this implementation is pretty naive and broadcasting means that all agents will receive the message.

Also, in terms of robustness, we think that Delegate MAS solutions perform better because they evaporate their pheromones slowly. A missed message will not result in a missed opportunity.

5 Conclusion