A Decentralized Approach for Anticipatory Vehicle Routing using Delegate Multi-Agent Systems

Rutger Claes, Tom Holvoet and Danny Weyns *Member, IEEE*

***Abstract*—Advanced vehicle guidance systems use real-time traffic information for routing traffic and avoiding congestion. Unfortunately, these systems are only able to react upon the presence of traffic jams, not to prevent the creation of unnecessary congestion. Anticipatory vehicle routing is promising in that respect, since this allows directing vehicle routing by accounting for traffic forecast information.**

This paper presents a decentralised approach for anticipatory vehicle routing that is particularly useful in large-scale dynamic environments. The approach is based on 'delegate MAS', an environment-centric coordination mechanism that is in part inspired by ant behaviour. Ant-like agents explore the environment on behalf of vehicles and detect congestion forecast, allowing vehicle to reroute. The approach is explained in depth, and evaluated by comparison against three alternative routing strategies. The experiments are done in simulation of a real-world traffic environment. The experiments indicate a considerable performance gain compared to the most advanced strategy under test, that is a TMC-based routing strategy.

I. Introduction

People use vehicles for making trips using the road infrastructure. The large number of vehicles today, and the limited capacity of the road networks make routing traffic a particularly challenging problem. Not only does a vehicle need to reach its destination, it is desired that the trip can be performed in a timely and comfortable fashion. Besides basic SatNav devices, which use static maps for fastest path routing, more advanced devices exploit broadcast traffic information (e.g. through Traffic Message Channel or TMC). An accident causing a traffic jam on the route of a vehicle can trigger the vehicle to reroute and bypass the traffic jam. This mechanism allows a substantial performance gain already.

One disadvantage of these state-of-the-art approaches lies in the fact that these allow only to react upon traffic jams after they occurred, and hence are already propagating delays in a typically substantial part of the traffic network. Anticipatory vehicle routing aims to encompass this by using forecast of traffic density. Forecast information can either be extracted from historical data, or can directly rely on the individual planned routes of the vehicles. Besides obtaining and disseminating forecast information, major challenges are: (1) to cope with the large scale of traffic - consisting of huge numbers of vehicles residing on large road networks, (2) to cope with dynamics - accidents, road blocks, demand peaks, have local effects with potentially far-reaching consequences, (3) stability - reactions of vehicles to traffic information must be managed to avoid unstable system behaviour due to vehicles rerouting continually.

In this paper, a decentralised approach for anticipatory vehicle routing is defined and evaluated. The approach is defined as a situated multi-agent system with environment-centric coordination. Situated agents are embedded, i.e. directly linked to the real-world environment, which they can observe and attempt to influence via actions. For coordinating such large numbers of entities (vehicles and road infrastructure elements), a coordination model that uses the environment as a shared space is appealing. The delegate Multi Agent Systems coordination model is inspired by ant behaviour - ants coordinate their activities such as food foraging not through direct ant-to-ant communication, but by dropping relevant information in the form of pheromones which are scented and interpreted by other ants. In our approach, ant-like agents explore the traffic environment on behalf of vehicles, and drop relevant information in ICT infrastructure that is coupled with the road infrastructure elements. This information can thereafter be used by other ant-like agents acting on behalf of other vehicles.

The approach presented in this paper was first outlined in the paper by Weyns et al. [1]. It has since been made more robust by removing the need for vehicle reservations and has been more thoroughly evaluated using micro-simulations of a real-world traffic environment. In this evaluation, the proposed approach is compared with three other approaches, including an advanced TMC-based route guidance system that broadcasts real-time information to vehicles.

The remainder of this paper is structured as follows. First we formulate the problem statement and describe our basic assumptions. Then we outline our proposed anticipatory vehicle routing using delegate multi-agent systems approach. Next we describe the experimental setup we use to evaluate this proposed approach and analyse the results of the experiments. Finally we draw conclusions.

II. Problem formulation

In our vehicle routing approach we consider a traffic network consisting of roads and junctions. The capacity of roads is determined by their length and the number of parallel lanes. Junctions have a capacity that is generally defined by the width of roads ending in the junction. The throughput of a road is the

R. Claes, T. Holvoet and D. Weyns are with DistriNet labs, Department of Computer Science, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium (email: rutger.claes@cs.kuleuven.be; tom.holvoet@cs.kuleuven.be; danny.weyns@cs.kuleuven.be).

This research was funded by the IWT - SBO project 'MASE' (project no. 060823), by the Research Foundation Flanders (FWO Vlaanderen) and by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund K.U.Leuven

sum of the throughput of the lanes in that road. The throughput of a lane is determined by the speed limit enforced on that lane. For junctions, the throughput is determined by a combination of factors such as the presence of traffic lights and the turning rules. The most important factors are the precedence of traffic based on road categorisation and the precedence of traffic approaching from right.

The traffic environment is a dynamic environment, and as such vehicle routing is a dynamic problem. We can identify two important causes for the changes in traffic intensities, namely (1) fluctuations in the demand and (2) fluctuations in the capacity. Fluctuations in the demand occur when vehicles enter the traffic network. Vehicles then need to be routed individually from their origin to their destination. As the amount of vehicles on roads increases, the speed at which they can travel generally declines and congestion is formed causing a reduction in road capacity. Traffic intensity thus affects the throughput of roads. When trying to minimize trip durations, traffic intensity should be taken into consideration.

Besides fluctuations in the amount of vehicles in the traffic network, there can also be fluctuations in the capacity of the traffic network. Events such as accidents, road blocks, road works or even bad weather can reduce the capacity of roads. These changes will have an effect on the throughput of roads and junctions, and thus on the duration of routes traversing over them.

These causes are not unrelated. When demand rises, so does the chance off accidents or other unforeseen road blockage.

The goal of our proposed coordination mechanism is to reduce trip durations. As a consequence, our mechanism also tries to reduce trip distance.

It must remain the responsibility of the driver to make route choices. This calls for a decentralized coordination mechanism in which driver preferences allow for fine-grained control over route characteristics. While individual route calculations are demanding, advances in information and communication technology in the traffic infrastructure allow for such a demanding approach.

III. Anticipatory vehicle routing using delegate multi-agent systems

In this section we outline the three main elements in our approach to anticipatory vehicle routing and describe how these elements interact to coordinate traffic. The approach makes realistic assumptions about the available infrastructure, which is briefly discussed.

A. Multi-agent based vehicle routing

Traffic is by nature an open environment. Vehicles continually enter and leave the system and are dispersed over the spatially distributed road infrastructure. Our approach is based on a situated multi-agent system for modeling the entities that need coordination. A situated multi-agent system consists of a number of autonomous entities, called agents, that are situated or embedded in an environment. The agents can observe and act locally in the environment. Coordination is decentralized - overall coordinated behaviour results from the interaction between the different agents. The use of agent technology in various aspects of traffic and transportation systems is well documented. Chen and Cheng give an overview in [2] of agent based applications in traffic and transportation systems including the use of agents based control mechanisms in intelligent transportation systems.

The MAS-based model of our approach is based on three basic types of entities, namely *vehicle agent*, *infrastructure agent*, and *virtual environment*.

Every vehicle is represented by a situated *vehicle agent*, deployed on (a smart device within) the vehicle. A vehicle agent is able to access information about that vehicle's intended destination and state, including location and speed. A vehicle agent guides the driver by providing information on routing towards its destination - not unlike SatNav route guidance devices do today.

The core elements of the road infrastructure (such as roads and crossroads) are represented and managed by *infrastructure agents*. Infrastructure agents are deployed on computation and communication devices in the road infrastructure. Infrastructure agents maintain a view on the current status of their road elements as well as information (received via vehicle agents) on pending visits. The latter information will evaporate over time unless refreshed by vehicle agents. Cooperation between *vehicle* and *infrastructure agents* requires the presence of vehicle to infrastructure communication or Vehicle-Infrastructure-Integration (VII) as described in the work of Ma et al. [3]. The authors use VII to achieve a real-time assessment of highway conditions, something the *infrastructure agents* in our approach to a lesser extend are also responsible for.

The *virtual environment* is a software representation of the environment. The physical road network is mapped onto a graph representation. The nodes of the graph represent road elements such as lanes and intersections. This virtual environment is a distributed software entity that is deployed on the electronic devices provided by the road infrastructure. The virtual environment conceptually hosts the infrastructure and vehicle agents - the agents can observe and act through this environment. The use of a virtual environment is discussed by Weyns et al. in [4].

Vehicle and infrastructure agents are responsible for coordinating traffic. Vehicle agents have two responsibilities. First, they explore (through the virtual environment) and search viable routes towards their respective destinations. Exploring a route means assessing its quality (in terms of time that it would take to follow this route). From this set of alternative explored routes, every vehicle agent selects one route which it intends to follow. The selection is based on the objective of individual traffic users, which is assumed to be the travel time.

To allow for anticipatory vehicle routing the vehicle agent has a second responsibility. Every vehicle agent needs to inform other agents of its intended routes as to allow other agents to incorporate this forecast occupancy in their own exploration. The vehicle agent achieves this by informing all infrastructure agents representing elements that are part of its intention. By doing so, all vehicle agents cooperatively maintain information about their intentions in the infrastructure agents. Infrastructure agents can use this information to determine future traffic loads and provide this information back to the vehicle agents while they explore viable routes, thereby improving the estimates of the vehicle agents make on trip duration, and thereby closing the information loop.

B. Delegate MAS for anticipatory vehicle routing

Typical implementations of multi-agent systems would achieve the communication patterns described in the previous section by having all agents communicate via direct exchange of messages. The large scale of such systems, and the fact that communication bandwidth is not unlimited leads to an environment-centric approach.

We use delegate multi-agent systems to achieve both the exploration and intention propagation functionality. Delegate multi-agent systems are introduced in [5].

Delegate multi-agent systems are inspired by food foraging in ant colonies and their use of pheromones. When coordinating the search for food, ants do not communicate directly with each other. Instead, they use smelling substances called *pheromones* to communicate. An ant is able to notify other ants on its way back from a food source to the nest by dropping pheromone on its current location. Other ants receive this information by scenting the pheromone. Pheromone deposits convey information by their intensity, type and location. If the information is not reinforced it will disappear. Meaning that if an ant does not deposit fresh pheromone on the same location at regular intervals, the evaporating pheromone will no longer be detectable to other ants. By following the gradient of scent, other ants can reach the same food source, without having communicated with other ants directly. When these ants return from the food source to the nest, they will reinforce the pheromone trail between food and nest, thus maintaining the information. As soon as the food source is depleted, the reinforcement stops and the information will start to dissolve.

In delegate multi-agent systems we use similar techniques. Instead of having vehicle and infrastructure agents communicating directly with each other, they send out lightweight agents that somewhat mimic the ants behaviour. To maintain a clear distinction between the main agents, namely the vehicle and infrastructure agents, and these lightweight agents we refer to the latter as *ants*. Together, these lightweight agents, or ants, form delegate multi-agent systems that offer certain services to the main agents: the agents can delegate some of their responsibilities to these delegate multi-agent systems by using these services. Figure 1 uses a UML conceptual diagram to show and relate the different concepts of our approach.

In our anticipatory vehicle routing strategy we employ two different types of ants offering two distinct services to the vehicle agents. Both ants are shown in Figure 2.

1) Exploration ants: A vehicle agent sends out exploration ants at regular time intervals. Exploration ants explore various paths between the agents current location and its destination. To explore a path, an exploration ant follows it through the virtual environment. At every road element it asks the infrastructure agent what the departure time from its element would be *if* the vehicle would arrive at a certain arrival time. It then continues to the next element on its path and asks the same question, this time using the previously received departure time as its new estimated arrival time. The exploration ants assume basic, static routing information to be available (similar to routing functionality readily available in SatNav devices).

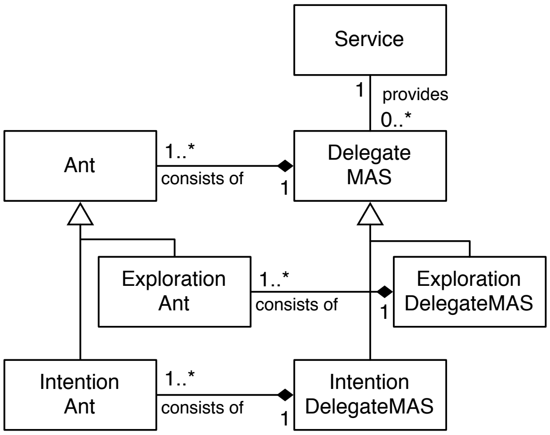


Fig. 1. The architectural pattern of delegate multi-agent systems. A delegate multi-agent system is a swarm of ant-like agents that can provide various services to a higher level agent.

Eventually an exploration ant reaches the vehicles destination with an estimate of how long it would take the vehicle to get there, taking into account the predicted delays along this route. The exploration ant then reports this aggregated data back to the vehicle agent by following its path reversely. The vehicle agent thus constantly receives alternative routes to its destination along with an estimate on the trip duration.

2) Intention ants: When a vehicle agent selects one of the explored routes as the route it intends to follow, it must make this information available for other vehicle agents to be able to take this into account. Vehicle agents do so by sending out intention ants over their intended route at regular intervals. These intention ants will follow the intended route through the virtual environment. While doing so they repeat the question also posed by the exploration ants. However, instead of just retrieving the departure time from the current road element, they also inform the infrastructure agent that the vehicle agent intends to make use of this road element between the arrival and departure time. Thus the intention ants provide the infrastructure agents with the information they need to predict future traffic intensities.

Vehicle agents are free to change their intentions when e.g. a newly explored path is considered preferable. If they would do so, they invalidate all notifications their intention ants have made. To prevent this incorrect information from building up in the virtual environment, the principle of evaporation is used. The notifications handed out by the intention ants will evaporate over time. If an agent changes its intention and no longer sends out intention ants over its old intention, the notifications on the old route will simply evaporate.

C. Design decisions in the implementation

The generic description above of our approach leaves many design issues unanswered. This section offers further informa-

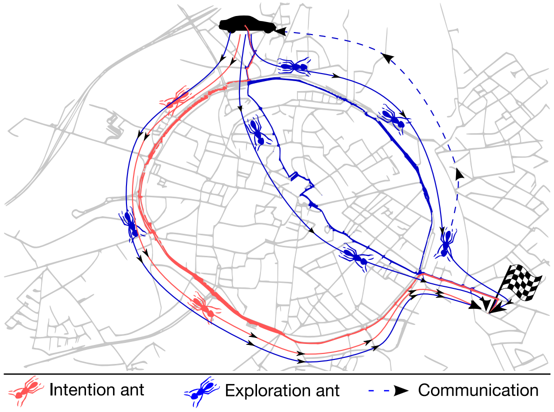


Fig. 2. Exploration and intention ants traversing routes in a virtual representation of Leuven. The exploration ants explore feasible routes (highlighted in dark grey) and return estimates on the routes duration. When the vehicle agent has chosen a route (highlighted in red) based on these estimates, it sends out an intention ant to notify the infrastructure agents of its pending visit.

tion on the details of our approach as it has been evaluated and tested.

1) Vehicle agent architecture: The vehicle agents architecture is based on the Belief-Desire-Intention (BDI) architecture [6] which has already been successfully used in vehicle agents [7]. The BDI architecture can easily be combined with the use of delegate multi-agent systems. The basic algorithm describing the inner working of the vehicle agents is given in Figure 3.

```
1:  loop
2:     // Update the vehicle agents beliefs with new
       feasible routes
3:     $routes \leftarrow getAlternatives(currentLocation, destination)$
4:     // Select the best route
5:     $selectedRoute \leftarrow choose(routes)$
6:     // Decide whether to revise intention
7:     if $reviseIntention(currentIntention, selectedRoute)$ then
8:        $currentIntention \leftarrow selectedRoute$
9:     end if
10:    // Propagate the current intention
11:    $propagateIntention(currentIntention)$
12:    // Instruct the driver
13:    $instructDriver(currentIntention)$
14: end loop
```

Fig. 3. The reasoning loop of a vehicle agent

When using anticipatory vehicle routing with delegate multi-agent systems line 3 will cause the vehicle agent to send out exploration ants. On line 5, the agent will select the route with the shortest trip duration based on the information send back by the exploration ants. On line 7, the vehicle agent will decide on whether to deviate to this new route. Whatever the agent decides, on line 11 the vehicle agent will send out an intention ant across its current intention.

2) Infrastructure agents: While explaining the delegate multi-agent systems approach in the previous section, infrastructure agents where given the functionality of collecting notifications from intention ants and using these notifications to provide predictive traffic intensity information. In this implementation the infrastructure agents do so by a simple learning algorithm. Infrastructure agents monitor the vehicles passing over them and collect information on the number of notifications and the average time it takes a vehicle to pass the road element. Using this information, the infrastructure is able to predict future traversal times based on the number of notifications it has received. Infrastructure agents have a parameterized model describing the relationship between traversal time and notifications, the parameters are continuously updated based on both historical and real-time data.

This learning algorithm is very simple, but appears to be sufficient. By using a learning algorithm and not a reservation based scheme, we are unaffected by the drawbacks of reservation schemes in traffic situations, such as problems with vehicles not guided by the system stealing reserved slots [8] and should be able to handle scenarios where only a portion of the drivers uses anticipatory vehicle routing.

The infrastructure agents need to be supported by the traffic infrastructure with the following infrastructure: (1) The road infrastructure is equipped with electronic devices which provide some computation power and are connected through a network. (2) The roadside computing devices need to communicate with the smart devices located in the vehicles. And finally, (3) The roadside computing devices can access sensor information on the current traffic intensity for learning purposes. These requirements are not unrealistic. The road pricing scheme currently being planned in The Netherlands has similar requirements.

IV. Experiment Setup

We have evaluated the delegate multi-agent system described in this paper by simulating it in a real-world setting, the city of Leuven, Belgium. In this section we discuss the setup of our experiments, namely the type of simulation we use, the map on which the model is based and the alternative routing strategies we will be comparing our delegate multi-agent system routing strategy with.

A. Traffic Micro-Simulation

To evaluate our delegate multi-agent routing strategy we compare it with alternative routing strategies. We have developed a micro-simulation capable of simulating detailed traffic scenarios. In this micro-simulation every individual vehicle is modeled by its position on the road. The vehicles are able to move across the traffic network by accelerating and decelerating, changing lanes and taking turns on junctions. The driving behavior of the vehicles are determined by the Intelligent Driver Model [9] (IDM).

The simulation process is given an Origin-Destination (OD) matrix containing vehicles start and destination locations annotated with the vehicles departure time and IDM model parameters. This ensures that all simulation processes simulate the same vehicles operated by the same drivers. All experiments described here are initiated with the same OD matrix of 28800 entries. The origin and destination are chosen at random with a distribution that favors trips cutting through the city. The entries are chosen as follows.

1) An angle θ_o is chosen from a uniform distribution. A radius ρ_o is chosen from an log-normal distribution with

the average just outside the beltway. Together, these act as polar coordinates originating from the city center and describe the vehicles starting location.

2) An angle θ_d is chosen from a normal distribution with a mean opposite to the $\theta_o - \pi$. A second radius ρ_d is taken from the same distribution as ρ_o. Together, these form the vehicles destination.
3) Both coordinate pairs are mapped on the closest traffic element - road or junction - in the simulated environment.

B. Traffic network of Leuven

The traffic network modeled in our micro simulation is that of the city of Leuven[1], Belgium. It includes over 1600, mostly bidirectional, roads and 1250 junctions. The data is detailed and not only describes the location of most of the cities roads and junctions, but also their characteristics such as their type, maximum speed and capacity. Figure 4 shows the region modeled in our experiments.

C. Alternative routing strategies

To evaluate the efficiency of our delegate multi-agent systems approach we have implemented three alternative routing strategies for comparison. These alternatives are all based on the A* algorithm [11] often used in traffic routing applications.

The first two alternative routing strategies, optimistic and pessimistic fastest route do not rely on communication. The third alternative is based on real-world usage of Traffic Message Channel [12], or TMC, a service that is commonly used in Belgium.

1) Optimistic fastest route strategy: In the optimistic fastest route strategy every vehicle relies on the A* algorithm combined with a cost function, C_{ofr}, described in equation 1 to calculate its individual route. C_{ofr} calculates the estimate travel time of a vehicle by iterating over all segments s_i in a road r and uses the length l and speed limit v_{max} to determine the traversal time, or cost, of r.

$$C_{ofr}\left(r\right) = \sum_{s_i \in r} \left(\frac{l(s_i)}{v_{max}\left(s_i\right)} \right) \tag{1}$$

Equation 1 results in the estimated time a vehicle would need to traverse a road in the absence of other traffic. Using equation 1 as the cost function in an A* algorithm results in routes that are the fastest as long as vehicles are solitary. Hence the term optimistic. Early experiments with optimistic shortest path and the Leuven street map indicated that this is a unrealistic routing strategy, yet useful as a reference strategy.

The city of Leuven has a beltway surrounding it. Most drivers consider this beltway the preferred way of driving from one side of the city to the other. This is not because of the speed limit on the beltway which is mostly 50 km/h, the same as in the inner-city region, but because it has more lanes than the narrower streets in Leuven centrum. Routes calculated with C_{ofr} have a tendency to cut straight through the city center, a strategy that might work if the city center is desolated, but that is likely to result in long unforeseen waiting periods otherwise.

2) Pessimistic fastest route strategy: A more realistic routing strategy is the pessimistic fastest route strategy. The cost function, described in equation 2, used by this strategy is an adaptation of equation 1. Here the time needed to drive down a road is weighed by a factor w determined by the number of lanes of the road.

$$C_{pfr}\left(r\right) = \sum_{s_i \in r} \left(w(s_i) \frac{l(s_i)}{v_{max}\left(s_i\right)} \right) \tag{2}$$

Where $w(s_i)$ decreases as the number of lanes in segment s_i increases. In our experiments we take

$$w(s_i) = \frac{1}{\sqrt{min\left(4, lanes(s_i)\right)}}$$

cutting the effect of the weighing factor of at a width of 4 lanes.

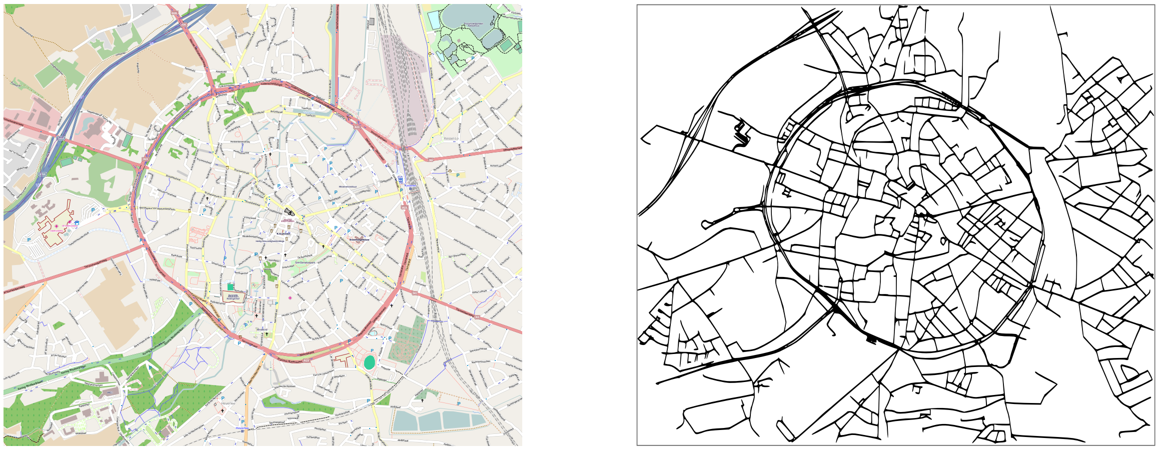
This routing strategy results in what appears to be much more realistic route choice. Simulation shows that vehicles now use the beltway around the city center avoiding the smaller roads, only turning towards the city center in the proximity of their destination. While we can expect the routes generated by this routing strategy to be somewhat slower than those generated with the optimistic variant, they are likely to become the better alternative when traffic intensity increases. Hence the term pessimistic.

3) TMC inspired routing strategy: The third and most important alternative to delegate multi-agent systems is a TMC inspired routing strategy. Many modern SatNav devices receive regular traffic updates over radio frequencies. In Belgium currently six such services exists. The information broadcasted by TMC systems includes congestion, accidents and other unforeseen circumstances that can affect routes calculated by in-vehicle SatNav devices. This information is generally broadcasted with a small time delay as incoming information such as floating car data or incident reports have to be processed and mapped in a Traffic Information Center before it can be broadcasted by radio stations. The number of locations on which TMC information can report is limited to a set of predefined locations already included in the digital maps of the major map vendors.

Our implementation is in many ways an improvement to existing TMC implementations because (1) information is broadcasted continuously, (2) it reports on all roads in the network and not just the major traffic arteries and finally (3) it includes average speeds of non or slightly congested roads and not only information about blocked roads. The improvements of our TMC implementation are not feasible in the real world because of the limited bandwidth available to the TMC system. While the TMC inspired strategy is not realistic, it makes a good reference model because, besides the imposed delay, it comes close to the ideal use of real-time data.

In this routing strategy, the average speed of all vehicles on a given road in a 5 minute interval is calculated. This information is gathered for all roads in the network and is broadcasted continuously to all vehicles with a 5 minute delay.

[1] For this region between latitudes 50.8612 and 50.8958, longitude 4.6665 and 4.7294 we accessed OpenStreetMap data from [10]



(a) Rendering of the OpenStreetMap data

(b) Graph model based on OpenStreetMap data

Fig. 4. Map of simulated region. Figure (a) on the left shows a rendering of the OpenStreetMap data. Figure (b) shows the graph with the over 1600 roads used in the micro-simulation.

Thus, at 10:15 all vehicles would receive the average speeds on all roads in the 10:05 - 10:10 time interval. The vehicles can use this historical data to replace v_{max} in equation 1. The in-vehicle routing calculations then use the following cost function for their A* algorithm.

$$\sum_{s_i \in r} \left(\frac{l(s_i)}{v_{tmc}(s_i)} \right) \tag{3}$$

Where v_{tmc} is the latest average speed for segment s_i the vehicle received. The number of lanes is no longer included in this cost function because it already influences the value of v_{tmc} if traffic is sufficiently dense.

V. Experiment results

In this section we will compare results obtained using the delegate multi-agent approach against those obtained with the three alternatives. All results where aquired using the simulator described in IV-A initiated with the Leuven traffic layout described in IV-B. For these experiments we generated a list of 28800 origins and destinations, during the simulation this list is used to instantiate 28800 vehicles. The rate at which this vehicles enter the network is dependent on the parameters of the experiment.

The experiments can be divided into two major clusters, namely those with static input rates and those with dynamic input rates. A static input rate means every second n vehicles are inserted into the network and that this n remains constant throughout the duration of the experiment. The origin and destination of the n vehicles will be taken from the OD matrix.

A dynamic input rate means n will evolve over time, allowing us to temporarily increase the traffic intensity in the network. The experiments with dynamic input rates described in this paper all have n evolving in a block wave fashion. The input rate will remain n for half a period and will be zero during the next half. The period is chosen to be 5 minutes as smaller periods have little impact on the traffic intensities and larger periods mean less drivers are affected by change in the traffic intensity during their trip.

By using the generated OD matrix, we guarantee the same demand in all experiments. Because of the different input rates, the conditions vehicles face in fulfilling these trips will differ.

A. Trip duration

In this section we will compare the different routing strategies based on the trip duration. All the route guidance algorithms use this as their primary heuristic in selecting the route they intent to follow. Therefore it becomes the most important metric when evaluating their performance. The trip distance, which is examined in the next section, is only a consequence of the route guidance algorithms efforts to minimize trip durations.

A standard normal distribution does not model the trip durations very well, the Weibull distribution appears to be a better fit for the observed durations. The distribution of trip duration for two separate experiments is shown in Figure 5. The Weibull distributions definition of the mean will be used to evaluate the trip durations.

Experiments show that only the TMC based routing strategy and the delegate multi-agent routing strategy are able to handle increasing traffic intensity under both static (Figure 6a) and dynamic (Figure 6b) input rates. Both of these routing strategies use information about traffic intensity on the vehicles intended route.

Examining the gain, $100\% \times (1 - mean_{ant}/mean_{tmc})$, that the use of our proposed approach, and thus the use of forecast data, has over the TMC based approach results in Figure 7. As the traffic intensity, and thus the likelihood of congestion, rises, the benefit of forecast data increases.

Taking these results into account, it appears that the use of forecast data results in shorter trip durations. The benefit of forecast data increases as traffic intensity increases and vehicles are confronted with dynamic traffic intensities. The use of routing strategies that use external data, namely the TMC inspired and the anticipatory routing strategies, drastically outperform stand-alone route guidance strategies.

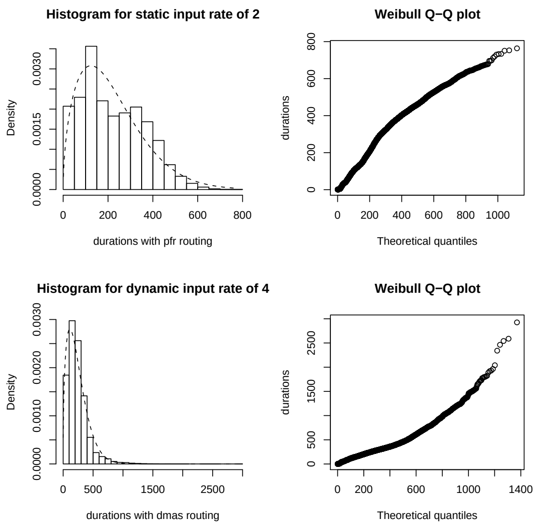


Fig. 5. Distribution of trip durations from two separate experiments. On the left are the histograms of two separate experiments with the fitted Weibull distribution density function. On the right the fit of the Weibull distribution is shown using a Q-Q plot. While not providing a perfect fit, the Weibull distribution describes the trip duration better then both the Gamma and log-normal distributions.

B. Trip distance

While trip duration is often the main criterium for route selection. The length of the route also plays an important role. The driver often has to consider the tradeoff between distance and duration. Always choosing a route resulting in an earlier time of arrival completely disregards fuel, maintenance and environmental costs.

The average trip distances shown in Figure 8a and 8b indicate that using anticipatory routing decreases trip distances. While this may seem counter intuitive, it is explained by the intention update strategy of the Vehicle agents. Vehicle agents using `ofr` or `pfr` never reconsider their route since they never receive new information. An A* algorithm using cost functions described in Equations 1 and 2 will not necessarily generate the *shortest* route, it will try to generate the *fastest* route. The anticipatory route guidance algorithm also generates, initially, the fastest route. But as congestion starts to form, the anticipatory route guidance algorithm starts looking for alternatives. These alternatives will often be the shorter, less fast route.

Figure 8 shows a steady upward trend in the trip distance when using the TMC based routing strategy. A possible explanation for the upward trend that is noticeable in both static and dynamic input profiles is the staleness of the information the TMC based algorithm uses. This could cause a buildup of rerouting actions where the TMC guided vehicle agent deviates from its original route, starts a detour only to find that this detour has also become congested. Such instability and inefficient decision making is predicted by [13].

VI. Related work

Dynamic vehicle routing is an extensively studied field of research [14]. In this paper we focus the discussion of related work on a number of representative agent based approaches used for vehicle routing. First we describe other work involving ‘anticipatory vehicle routing’ and see whether it describes the same problem we do. The discussion focusses somewhat on the work on anticipatory vehicle routing done in [15] because of its resemblance to the approach we describe here. Then, as a comparison of the delegate multi-agent systems approach, we look at a number of routing approaches that are biologically inspired, including the use swarm techniques and stigmergy. Finally we discuss reservation based mechanisms and the hints at using machine learning found in [16].

A. Anticipatory vehicle routing

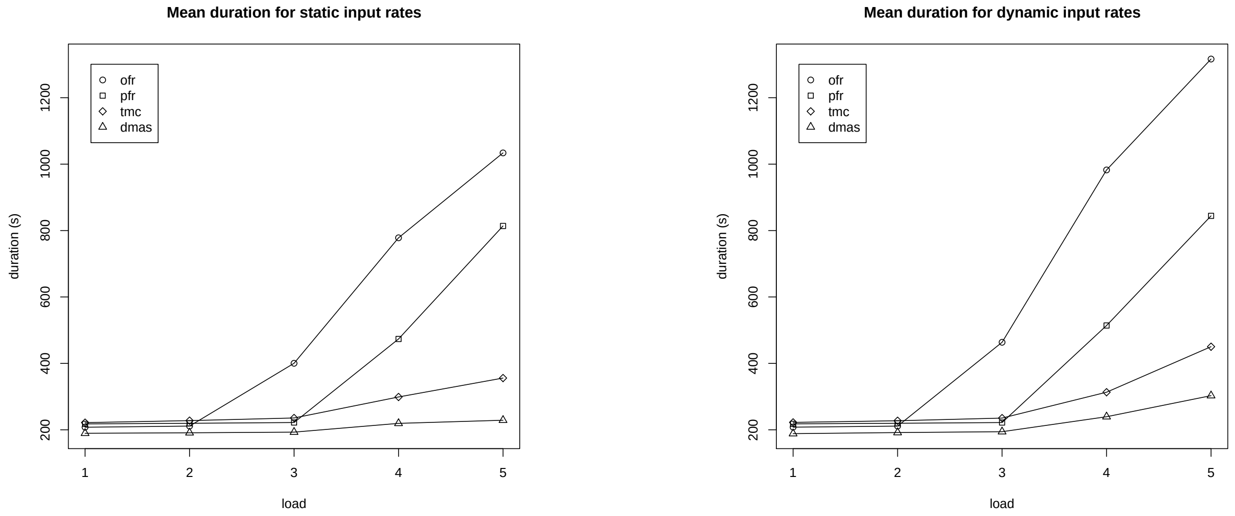
There is already an extensive body of work involving the term ‘anticipatory vehicle routing’. Some of this related work, such as that of Wunderlich, Kaufman and Smith deals with a similar problem as this paper, though their focus differs. Other work, such as [17], [18], focusses on different problems such as the dispatching of pickup vehicles to meet anticipated future customer demands in a pickup and delivery problem.

The US National ITS Architecture [19] outlines an evolution in route guidance architectures. The first step of this evolution is an autonomous architecture in which all vehicles make isolated decisions based on static link data, corresponding with the optimistic and pessimistic route guidance algorithms we described in Section IV-C. This autonomous architecture is followed by a decentralized architecture in which real-time information is broadcasted to vehicles allowing them to adjust their routing to current traffic densities. The third and final step would be a centralized architecture in which vehicles send routing requests to an Independent Service Provider. This ISP will then provide the vehicle with an individualised route taking into account all other issued routes to predict future traffic states. Such a centralized architecture is expected to solve the problems predicted by [13]. In [13] the authors argue that providing information to vehicles can lead to instability and inefficient decision making as all vehicles take part in a minority game.

The need for this last evolution, the transition from a decentralized to centralized architecture, can be avoided by providing the vehicles with predictive instead of real-time information according to [15].

A comparison between a decentralized and centralized architecture for traffic route guidance systems is discussed in [15]. The distinction between a decentralized and centralized architecture is made based on the location of the route choice. If the vehicle makes a routing decision on its own based on the information broadcasted by an Independent Service Provider, the architecture is said to be decentralized. If, on the other hand, the routing decision is made centrally by some Independent Service Provide and is only then broadcasted to the vehicle, the architecture is considered as centralized.

The route guidance system described in [15] uses forecast, instead of real-time, data is a way that is similar to ours. The method of collecting and distributing this forecast data, however, differs from the approach we describe in this paper. The architecture described in [15] can only be called decentralized when disregarding the role of the Independent Service



(a) Static input rates

(b) Dynamic input rates

Fig. 6. Trip durations for all route guidance strategies for both static and dynamic input rates at various levels. Only the TMC inspired (`tmc`) and delegate multi-agent (`dmas`) based strategies perform consistently under various loads. The optimistic (`ofr`) and pessimistic (`pfr`) fastest route algorithms fail to handle the increase in vehicles.

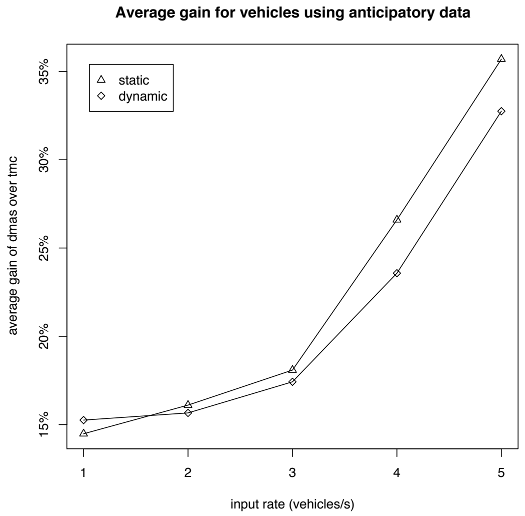


Fig. 7. Gain in average trip duration obtained by using forecast instead of real-time data.

Provider. In this paper we describe a truly decentralized architecture where the functionality attributed to the ISP in [15] is distributed across the traffic infrastructure.

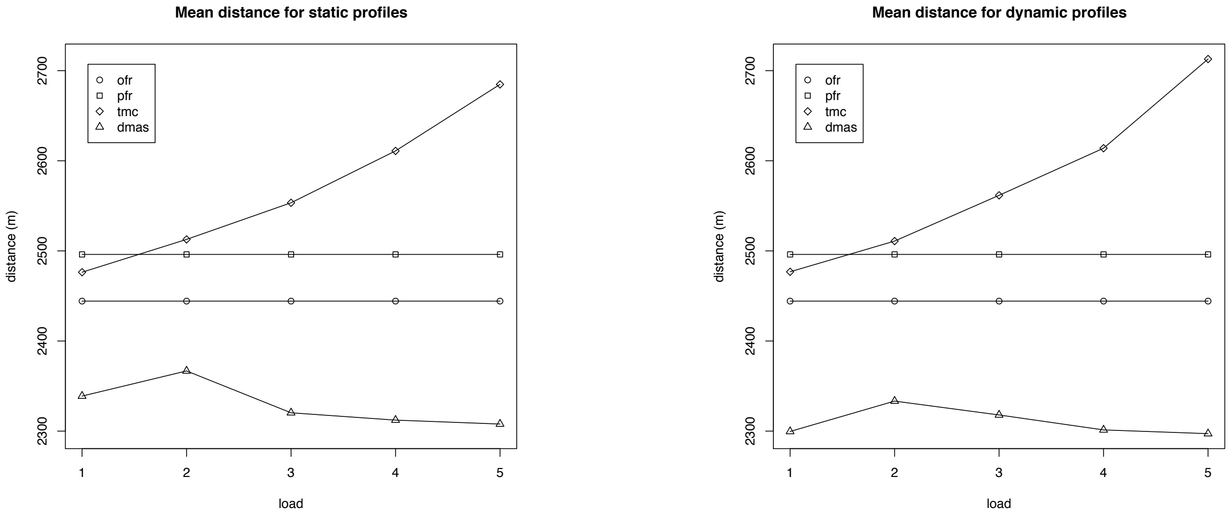
B. Propagation of information

One of the contributions of this paper compared to the work discussed in the previous section is the decentralized propagation of vehicles intentions using swarms. The lightweight agents in delegate multi-agent systems have some resemblance to an approach known as Polyagents introduced by Brueckner and Parunak [20], [21]. Propagation of information through traffic networks using biologically inspired mechanisms such as pheromones or swarms have been studied extensively before by Ando in [22] and Tatomir in [23].

1) The use of pheromones: In [22] pheromones are used to aggregate and propagate traffic densities in traffic networks. Vehicles in [22] drop virtual pheromones at their current location. These pheromones will spread locally before evaporating over time. The pheromone intensity at any given location in the network is used as an indication of traffic density at that location. As vehicles drive slower, they will deposit more pheromones in the same region. Information about traffic density and congestions are thus propagated through the environment. The information is kept up to date by the evaporation process. Vehicles can look ahead and sense the pheromone levels on their intended route and compare these pheromone levels with those of alternative routes.

Digital pheromones are also used in [24]. Here the pheromones are deposited onto the roads and represent information about the number of vehicles and their link entry time. The deposited information can be used to label roads as congested and this information can be transmitted back to vehicles. The use of pheromones in [24] resembles that of [25] in that it uses pheromones to represent the current traffic situation in a virtual environment. How the pheromones are applied in [24] differs from [22], but the view they provide on the traffic network is similar.

The main difference between the route guidance approach in [22] and [24] and the approach described in this paper are the nature of the data. Both approaches store information about traffic densities in the environment and thus use a process called stigmergy to propagate information to other agents. The information stored in the environment by [22] and [24] represents current or past traffic information, while the information stored in the environment by delegate multi-agent systems describes future traffic densities. The use of



(a) static input rates

(b) dynamic input rates

Fig. 8. Trip distances for all route guidance strategies for both static and dynamic input rates at various levels.

the environment to store pheromones and the process of propagation and evaporation are based on the work on Ant Colony Optimisation [26].

A second difference between the approach presented in this paper and that of [22] is the entity that deposits the pheromone trails. In our approach, since pheromones represent future information, the pheromones are deposited by ant-like agents operating in a delegate-multi agent system. The approach described in [22] has the vehicle agents depositing the pheromones as they represent real-time traffic information.

2) The use of swarm computing: The use of ants in the domain of traffic are described in [27] and [23]. In the first paper, [27], the authors focus more on pickup and delivery problems. The second paper, [23], describes a hierarchical routing system using three different types of ants, namely: local ants, backward ants and exploration ants. The function of the exploration ants in [23] differs from those described in this paper. In [23] exploration ants maintain information about routes between different sectors in the hierarchy. Local ants are dispatched by nodes in the network to prepare for arriving vehicles. These local ants will explore the route the vehicle intents to follow and updates the information using backward ants in all the nodes involved in this route.

C. Reservation based mechanisms

In our delegate multi-agent approach vehicles send out intention ants to notify road agents of pending visits. While this is not a reservation based mechanism, it does resemble one. Reservation based intersection control has been described by Dresner and Stone in [28] and expanded to a market-inspired approach in [29]. The authors of [28] later experimented with traffic scenarios where not all vehicles use the reservation based mechanism in [8] and identify a number of difficulties with reservation based mechanisms in such settings.

In [16] the authors identify a number of learning opportunities for both their agent types, namely ‘driver agents’ and ‘intersection agents’. Here the authors hint at the use of a learning approach similar to the one we use to replace the need for reservations, by saying that ‘intersection agents’ could learn the characteristics of traffic as a response to a number of inputs, including recent history.

VII. Conclusion

In this paper we described a routing strategy for anticipatory vehicle routing using delegate multi-agent systems. This routing strategy is able to route vehicles more efficiently by using forecast information. This anticipatory vehicle information is collected and distributed in a decentralized fashion, unlike other approaches involving forecast information where collection and distribution of information is performed as a central service. The distributed nature of this approach fits the distributed nature of the traffic domain and ensures scalability requirements can be met more easily then in centralized systems.

Experiments show that the use of forecast data, even when gathered in a decentralized manner, helps drivers reach their destination up to 35% faster compared to drivers using no data or real-time data made available by TMC services. The forecast data not only allows drivers to avoid existing congestions, but also prevents them from forming congestion.

Further research on the subject of anticipatory vehicle routing and the use of delegate multi-agent systems remains necessary. Providing efficient and stable routes to vehicles will be challenging, even with the use of forecast data, as we apply our approach to larger traffic scenarios involving more dynamics. Further experimentation with the learning algorithms used by the infrastructure agents could improve the quality of the predictive information by including information such as time of day, day of week.

The use of forecast information obtained by delegate multi-agent systems does come at a cost. Instead of the one-way broadcasting of information needed for TMC based systems, our approach would require two-way communication between vehicles and the road infrastructure. However, road side pricing schemes currently being deployed share these communication and computation requirements.

References

[1] D. Weyns, T. Holvoet, and A. Helleboogh, "Anticipatory vehicle routing using delegate multi-agent systems," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, sep. 2007, pp. 87 –93.

[2] B. Chen and H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 485 –497, jun. 2010.

[3] Y. Ma, M. Chowdhury, A. Sadek, and M. Jeihani, "Real-time highway traffic condition assessment framework using vehicle-infrastructure integration (vii) with artificial intelligence (ai)," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 4, pp. 615 –627, dec. 2009.

[4] D. Weyns, A. Omicini, and J. Odell, "Environment as a first class abstraction in multiagent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 5–30, 2007.

[5] T. Holvoet and P. Valckenaers, "Exploiting the environment for coordinating agent intentions," *Environments for Multi-Agent Systems III*, pp. 51–66, 2007.

[6] A. Rao and M. Georgeff, "Deliberation and intentions," in *Proc. of 7th Conference on Uncertainity in Artificial Intelligence*, 1991.

[7] R. J. F. Rossetti, R. H. Bordini, A. L. C. Bazzan, S. Bampi, R. Liu, and D. V. Vliet, "Using bdi agents to improve driver modelling in a commuter scenario," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5-6, pp. 373 – 398, 2002.

[8] K. Dresner and P. Stone, "Sharing the road: autonomous vehicles meet human drivers," in *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 1263–1268.

[9] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805–1824, Aug 2000.

[10] (2009, 08) Openstreetmap. [Online]. Available: http://www.openstreetmap.org

[11] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100 –107, jul. 1968.

[12] P. Davies, "The radio system-traffic channel," in *Vehicle Navigation and Information Systems Conference, 1989. Conference Record*, Sep 1989, pp. A44–A48.

[13] J. Wahle, A. L. C. Bazzan, F. Klügl, and M. Schreckenberg, "Decision dynamics in a traffic scenario," *Physica A: Statistical Mechanics and its Applications*, vol. 287, no. 3-4, pp. 669 – 681, 2000.

[14] E. Schmitt and H. Jula, "Vehicle route guidance systems: Classification and comparison," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 242–247, Sept. 2006.

[15] K. Wunderlich, D. Kaufman, and R. Smith, "Link travel time prediction for decentralized route guidance architectures," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 1, pp. 4–14, 2000.

[16] K. Dresner and P. Stone, "Multiagent traffic management: Opportunities for multiagent learning," *Learning and Adaption in Multi-Agent Systems*, pp. 129–138, 2006.

[17] B. Thomas and C. White Iii, "Anticipatory route selection," *Transportation Science*, vol. 38, no. 4, p. 473, 2004.

[18] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Exploiting knowledge about future demands for real-time vehicle dispatching," *Transportation Science*, vol. 40, no. 2, pp. 211–225, 2006.

[19] "National intelligent transportation systems architecture, theory of operations," Washington, DC, USA, 1998.

[20] H. Van Dyke Parunak, S. Brueckner, D. Weyns, T. Holvoet, P. Verstraete, and P. Valckenaers, "E pluribus unum: Polyagent and delegate mas architectures," in *Multi-Agent-Based Simulation VIII*, ser. Lecture Notes in Computer Science, L. Antunes, M. Paolucci, and E. Norling, Eds. Springer Berlin / Heidelberg, 2008, vol. 5003, pp. 36–51.

[21] H. V. D. Parunak and S. Brueckner, "Concurrent modeling of alternative worlds with polyagents," *Multi-Agent-Based Simulation VII*, pp. 128–141, 2007.

[22] Y. Ando, O. Masutani, H. Sasaki, H. Iwasaki, Y. Fukazawa, and S. Honiden, "Pheromone model: Application to traffic congestion prediction," *Engineering Self-Organising Systems*, pp. 182–196, 2006.

[23] B. Tatomir and L. Rothkrantz, "Hierarchical routing in traffic using swarm-intelligence," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 230–235, Sept. 2006.

[24] W. Narzt, G. Pomberger, U. Wilflingseder, O. Seimel, D. Kolb, J. Wieghardt, H. Hortner, and R. Haring, "Self-organization in traffic networks by digital pheromones," *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 490–495, 30 2007-Oct. 3 2007.

[25] Y. Ando, Y. Fukazawa, O. Masutani, H. Iwasaki, and S. Honiden, "Performance of pheromone model for predicting traffic congestion," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2006, pp. 73–80.

[26] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[27] R. Montemanni, L. Gambardella, A. Rizzoli, and A. Donati, "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 10, no. 4, pp. 327–343, 12 2005.

[28] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 530–537.

[29] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 617–624.

Rutger Claes is a PhD student and research assistant at DistriNet Labs of the Computer Science department at the Katholieke Universiteit Leuven. He received his Master of Applied Science and Engineering in Computer Science from the Katholieke Universiteit Leuven. Rutger's research interests are the application of multi-agent systems, online simulation and swarm based coordination in intelligent traffic systems.

Tom Holvoet is a professor in the Department of Computer Science, Katholieke Universiteit Leuven. His research interests include software engineering of decentralized and multi-agent systems, coordination, software architecture, and autonomic computing. He received his PhD in computer science from the Katholieke Universiteit Leuven in 1997 on open distributed software development. Tom heads a research team with three post-doctoral and 6 PhD students within the DistriNet Lab.

Danny Weyns is a post-doctoral researcher at the Katholieke Universiteit Leuven funded by the Research Foundation Flanders (FWO). He received a Ph.D in Computer Science in 2006 from the Katholieke Universiteit Leuven for work on multi-agent systems and software architecture. Danny's main research interests are in software architecture, self-adaptive systems, multi-agent systems, and middleware for decentralized systems.