A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems

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

***Abstract*—Advanced vehicle guidance systems use real-time traffic information to route traffic and to avoid congestion. Unfortunately, these systems can only react upon the presence of traffic jams and not to prevent the creation of unnecessary congestion. Anticipatory vehicle routing is promising in that respect, because this approach allows directing vehicle routing by accounting for traffic forecast information. This paper presents a decentralized approach for anticipatory vehicle routing that is particularly useful in large-scale dynamic environments. The approach is based on *delegate multiagent systems*, i.e., an environment-centric coordination mechanism that is, in part, inspired by ant behavior. Antlike agents explore the environment on behalf of vehicles and detect a congestion forecast, allowing vehicles to reroute. The approach is explained in depth and is evaluated by comparison with three alternative routing strategies. The experiments are done in simulation of a real-world traffic environment. The experiments indicate a considerable performance gain compared with the most advanced strategy under test, i.e., a traffic-message-channel-based routing strategy.**

***Index Terms*—Autonomous agents, distributed control, intelligent vehicles, mobile agents, multiagent systems, navigation, software architecture, traffic control.**

I. Introduction

PEOPLE use vehicles to go places using the road infrastructure. The large number of current vehicles 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, but it also is desired that the trip can be performed in a timely comfortable fashion. Aside from basic SatNav devices, which use static maps for the fastest path routing, more advanced devices exploit broadcast traffic information, e.g., through a traffic message channel (TMC). An accident that causes 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.

One disadvantage of state-of-the-art approaches is that they allow us only to react upon traffic jams after they have occurred and, hence, already propagate delays in a typically substantial part of the traffic network. Anticipatory vehicle routing aims at encompassing this approach by using forecast of traffic density. Forecast information can either be extracted from historical data or directly rely on the individual planned routes of the vehicles. Aside from obtaining and disseminating forecast information, major challenges are listed as follows: 1) to cope with the large scale of traffic, which consists of large numbers of vehicles that reside on large road networks; 2) to cope with dynamics, e.g., accidents, road blocks, and demand peaks, that have local effects with potentially far-reaching consequences; 3) stability, i.e., reactions of vehicles to traffic information, which must continually be managed to avoid unstable system behavior due to vehicle rerouting.

In this paper, a decentralized approach for anticipatory vehicle routing is defined and evaluated. The approach is defined as a situated multiagent system (MAS) 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 through actions. To coordinate 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 MAS coordination model is inspired by ant behavior—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, antlike agents explore the traffic environment on behalf of vehicles and drop relevant information in an information and communication technology (ICT) infrastructure that is coupled with the road infrastructure elements. This information can thereafter be used by other antlike agents that act 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 more thoroughly been evaluated using microsimulations 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 the delegate MAS approach. Next, we describe the experimental setup that we use to evaluate this proposed approach and analyze the results of the experiments. Finally, we draw our conclusions.

Manuscript received December 15, 2008; revised November 30, 2009 and September 27, 2010; accepted December 24, 2010. This work was supported in part by the Innovation by Science and Technology–Flemish Program for Strategic Basic Research project “A Model-Based Approach for Evaluating the Safety-End Environmental Effects of Traffic Policy Measures” (MASE; Project 060823) through the Research Foundation Flanders, by the Interuniversity Attraction Poles Program Belgian State, Belgian Science Policy, and by the Research Fund Katholieke Universiteit Leuven. The Associate Editor for this paper was R. J. F. Rossetti.

The authors are with the DistriNet Laboratory, Department of Computer Science, Katholieke Universiteit Leuven, 3001 Leuven, Belgium (e-mail: rutger.claes@cs.kuleuven.be; tom.holvoet@cs.kuleuven.be; danny.weyns@cs.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2105867

II. Problem Formulation

In our vehicle routing approach, we consider a traffic network that consists 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 that end in the junction. The throughput of a road is the 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 categorization and the precedence of traffic approaching from the right.

The traffic environment is a dynamic environment, and as such, vehicle routing is a dynamic problem. We can identify the following two important causes for the changes in traffic intensities: 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 individually be routed 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 considered.

Aside from 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 work, 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 that traverse them.

These causes are not unrelated. When demand rises, so does the chance of 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.

Making route choices must remain as the responsibility of the driver. This condition calls for a decentralized coordination mechanism in which driver preferences allow for fine-grained control over route characteristics. Although individual route calculations are demanding, advances in ICT in the traffic infrastructure allow for such a demanding approach.

III. Anticipatory Vehicle Routing Using Delegate Multiagent 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. Multiagent-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 MAS for modeling the entities that need coordination. A situated MAS consists of a number of autonomous entities, called agents, that are situated or embedded in an environment. The agents can locally observe and act in the environment. Coordination is decentralized—overall coordinated behavior 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 in [2] give an overview of agent-based applications in traffic and transportation systems, including the use of agent-based control mechanisms in intelligent transportation systems.

The MAS-based model of our approach is based on the following three basic types of entities: 1) *vehicle agent*; 2) *infrastructure agent*; and 3) *virtual environment*.

Every vehicle is represented by a situated *vehicle agent*, deployed on (a smart device within) the vehicle. A vehicle agent can 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 toward its destination and 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 and information (received through 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, for which the *infrastructure agents* in our approach, to a lesser extent, are also responsible.

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 toward 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 that 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 to allow other agents to incorporate this forecast occupancy in their own exploration.

The vehicle agent achieves this approach by informing all infrastructure agents that represent 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, whereas they explore viable routes, thereby improving the estimates of the vehicle that agents make on trip duration and closing the information loop.

B. Delegate MAS for Anticipatory Vehicle Routing

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

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

Delegate MASs are inspired by food foraging in ant colonies and their use of pheromones. When coordinating the search for food, ants do not directly communicate with each other. Instead, they use smelling substances called *pheromones* to communicate. An ant can 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, i.e., 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 directly communicating with other ants. 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 MASs, we use similar techniques. Instead of having vehicle and infrastructure agents directly communicate with each other, they send out lightweight agents that somewhat mimic the ants' behavior. To maintain a clear distinction between the main agents, i.e., the vehicle and infrastructure agents, and these lightweight agents, we refer to the latter agents as *ants*. Together, these lightweight agents, or ants, form delegate MASs that offer certain services to the main agents: the agents can delegate some of their responsibilities to these delegate MASs by using these services. Fig. 1 uses a Unified Modeling Language (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, which offer two distinct services to the vehicle agents. Both ants are shown in Fig. 2.

1) Exploration Ants: A vehicle agent sends out exploration ants at regular time intervals. Exploration ants explore various paths between the agent's 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).

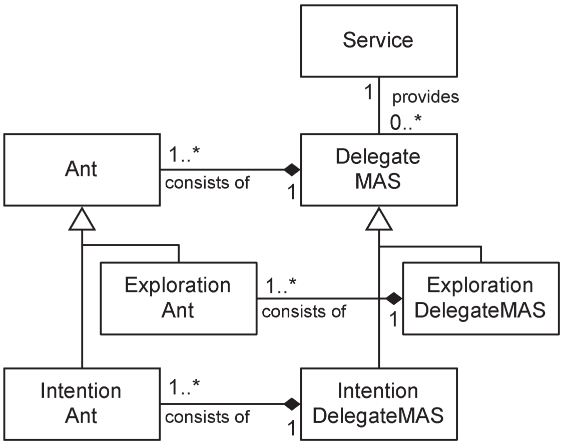


Fig. 1. Architectural pattern of delegate MASs. A delegate MAD is a swarm of antlike agents that can provide various services to a higher level agent.

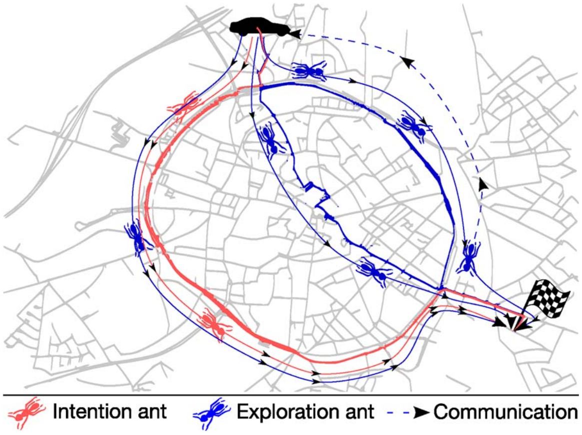


Fig. 2. Exploration and intention ants that traverse routes in a virtual representation of Leuven, Belgium. The exploration ants explore feasible routes (highlighted in dark gray) and return estimates on the routes duration. When the vehicle agent has chosen a route (highlighted in red) based on these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

Eventually, an exploration ant reaches the vehicle's 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 reversely following its path. 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 that it intends to follow, it must make this information available for other vehicle agents to take 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

```
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. Reasoning loop of a vehicle agent.

exploration ants. However, instead of simply 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 times. 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 that 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 aforementioned generic description of our approach leaves many design issues unanswered. This section offers further information on the details of our approach as it has been evaluated and tested.

1) Vehicle Agent Architecture: The vehicle agent 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 MASs. The basic algorithm that describes the inner working of the vehicle agents is given in Fig. 3.

When using anticipatory vehicle routing with delegate MASs, 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 sent 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 MAS approach in the previous section, infrastructure agents were 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 that pass over them and collect information on the number of notifications and the average time that it takes a vehicle to pass the road element. Using this information, the infrastructure can predict future traversal times based on the number of notifications that it has received. Infrastructure agents have a parameterized model that describes the relationship between the traversal time and notifications, and 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 that are not guided by the system that steals reserved slots [8] and should 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 that 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 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 that is currently planned in the Netherlands has similar requirements.

IV. Experiment Setup

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

A. Traffic Microsimulation

To evaluate our delegate multiagent routing strategy, we compare it with alternative routing strategies. We have developed a microsimulation that can simulate detailed traffic scenarios. In this microsimulation, every individual vehicle is modeled by its position on the road. The vehicles can move across the traffic network by accelerating and decelerating, changing lanes, and taking turns on junctions. The driving behavior of the vehicles is determined by the intelligent driver model [9] (IDM).

Simulation is given an origin–destination (OD) matrix that contains vehicles' start and destination locations that are annotated with the vehicles' departure time and IDM model parameters. This approach ensures that all simulation processes simulate the same vehicles that are operated by the same drivers. All experiments described here are initiated with the same OD matrix of 28 800 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 coordinates act as polar coordinates that originate in the city center and describe the vehicles starting location.

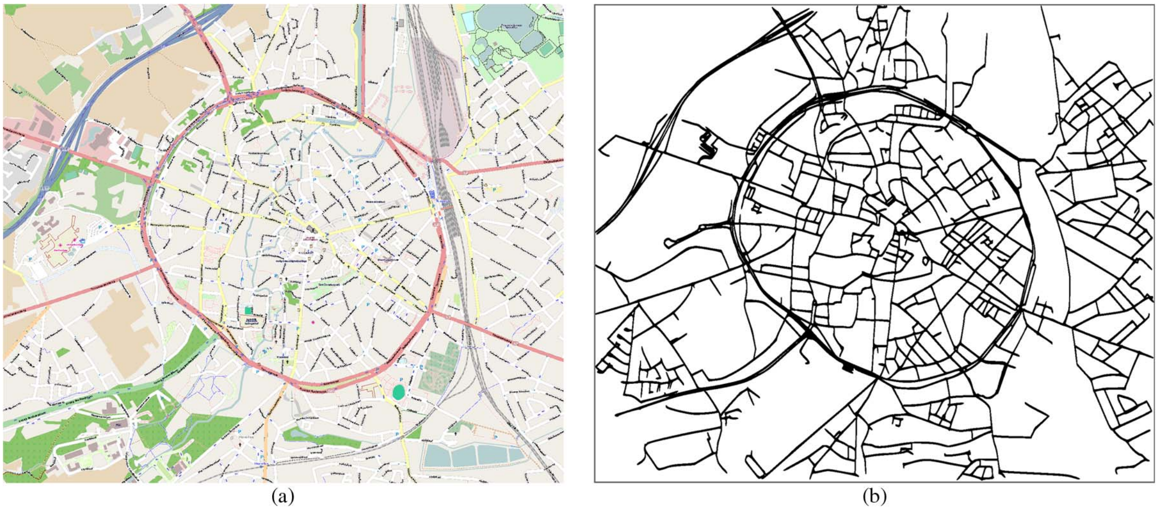


Fig. 4. Map of a simulated region. (a) Rendering of the OpenStreetMap data. (b) Graph with more than 1600 roads used in the microsimulation.

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 coordinates 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 microsimulation is that of the city of Leuven,[1] Belgium. It includes more than 1600 roads, mostly bidirectional, and 1250 junctions. The data are detailed and describe not only the location of most of the cities roads and junctions but their characteristics, such as their type, maximum speed, and capacity, as well. Fig. 4 shows the region modeled in our experiments.

C. *Alternative Routing Strategies*

To evaluate the efficiency of our delegate MAS 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 routes—do not rely on communication. The third alternative is based on the real-world usage of TMC [12], i.e., 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}, as described in (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, i.e.,

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

Equation (1) results in the estimated time that a vehicle would need to traverse a road in the absence of other traffic. Using (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 approach is an unrealistic routing strategy but is useful as a reference strategy.

The city of Leuven has a beltway that surrounds it. Most drivers consider this beltway the preferred way of driving from one side of the city to the other side. This case is not because of the speed limit on the beltway, which is mostly 50 km/h, the same speed limit as in the inner city region, but because it has more lanes than the narrower streets in Leuven centrum. Routes that are calculated with C_{ofr} have a tendency to cut straight through the city center, which is a strategy that might work if the city center is desolated but 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, as described in (2), used by this strategy is an adaptation of (1). Here, the time needed to drive down a road is weighed by a factor w that is determined by the number of lanes of the road, i.e.,

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

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 four lanes.

This routing strategy results in what appears to be a much more realistic route choice. Simulation shows that vehicles now use the beltway around the city center, avoiding the smaller roads, only turning toward the city center in the proximity of

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

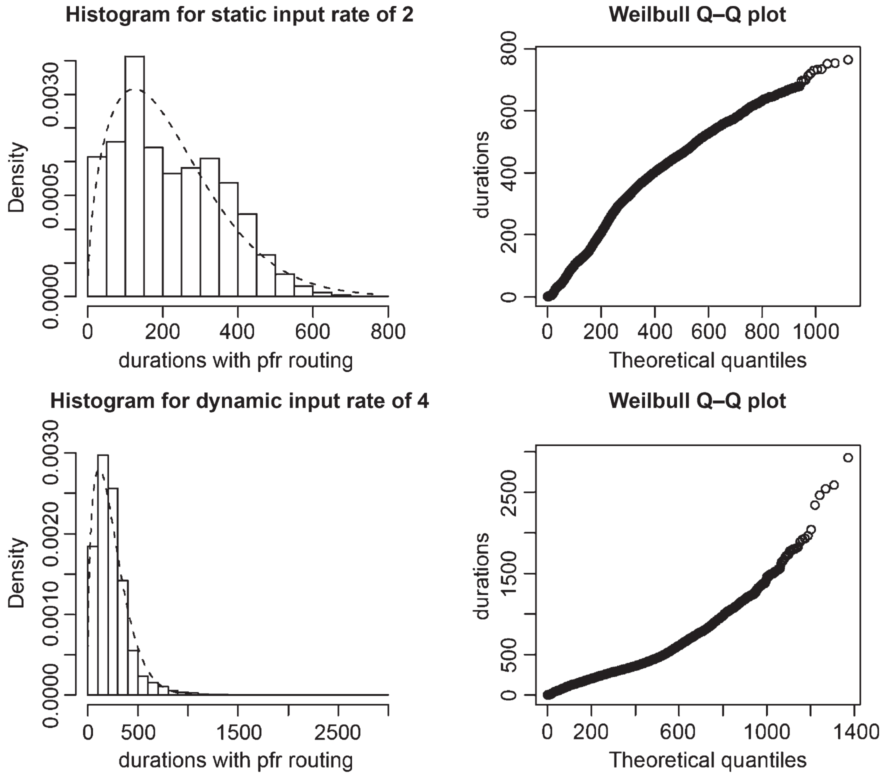


Fig. 5. Distribution of trip durations from two separate experiments. (Left) Histograms of two separate experiments with the fitted Weibull distribution density function. (Right) Fit of the Weibull distribution using a Q–Q plot. While not providing a perfect fit, the Weibull distribution describes the trip duration better than both the Gamma and log-normal distributions.

their destination. Although we can expect the routes generated by this routing strategy to be somewhat slower than the routes generated with the optimistic variant, they are likely to become the better alternative when traffic intensity increases, hence the term *pessimistic*.

TMC-Inspired Routing Strategy: The third and most important alternative for delegating MASs is a TMC-inspired routing strategy. Many modern SatNav devices receive regular traffic updates over radio frequencies. In Belgium, six such services currently exist. The information broadcast by TMC systems includes congestion, accidents, and other unforeseen circumstances that can affect routes calculated by in-vehicle SatNav devices. This information is generally broadcast with a small time delay as incoming information such as floating car data or incident reports that have to be processed and mapped in a traffic information center before it can be broadcast 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 for the following three reasons: 1) Information is continuously broadcast; 2) it reports on all roads in the network and not just the major traffic arteries; and 3) it includes average speeds of noncongested 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. Although the TMC-inspired strategy is not realistic, it makes a good reference model, because aside from the imposed delay, it comes close to the ideal use of real-time data.

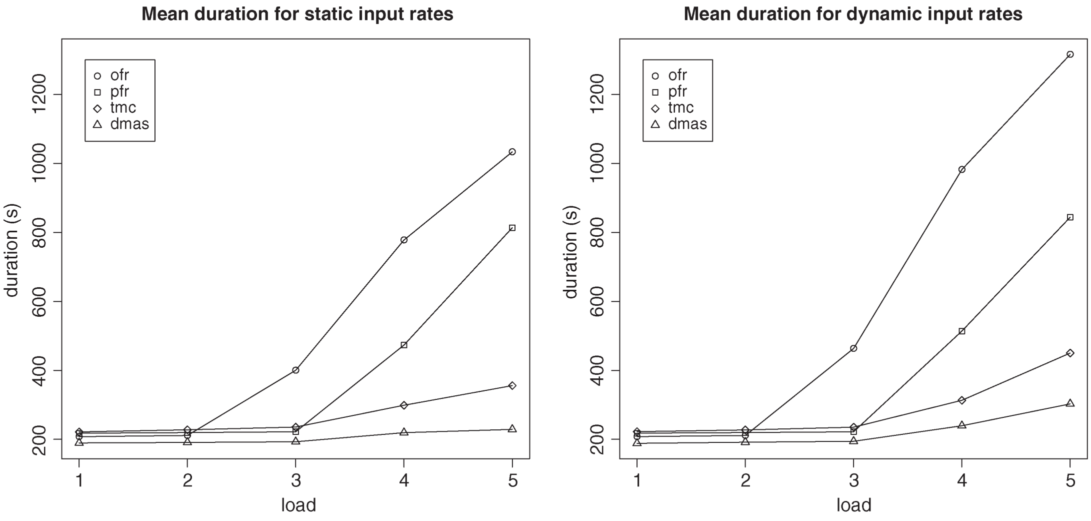
In this routing strategy, the average speed of all vehicles on a given road in a 5-min interval is calculated. This information is gathered for all roads in the network and is continuously broadcast to all vehicles with a 5-min delay. Thus, at 10:15, all vehicles would receive the average speeds on all roads in the 10:05–10:10 time interval. The vehicles can use this historical data to replace $v_{\max}$ in (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 that 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 the results obtained using the delegate MAS approach with the results obtained with the three alternatives. All results were acquired using the simulator described in Section IV-A, initiated with the Leuven traffic layout described in Section IV-B. For these experiments, we generated a list of 28 800 origins and destinations, and during the simulation, this list is used to instantiate 28 800 vehicles. The rate at which these vehicles enter the network is dependent on the parameters of the experiment.



(a)

(b)

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 MAS (`dmas`)-based strategies consistently perform under various loads. The optimistic (`ofr`) and pessimistic (`pfr`) fastest route algorithms fail to handle the increase in vehicles. (a) Static input rates. (b) Dynamic input rates.

The experiments can be divided into the following two major clusters: 1) experiments with static input rates and 2) experiments with dynamic input rates. A static input rate means that every second, n vehicles are inserted into the network and 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 that 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 that evolves 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 min, because smaller periods have little impact on the traffic intensities, and larger periods mean that fewer 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 that 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 trip duration as their primary heuristic in selecting the route that they intend 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' effort to minimize trip durations.

A standard normal distribution does not model the trip durations very well, and 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 Fig. 5. The Weibull distributions definition of the mean will be used to evaluate the trip durations.

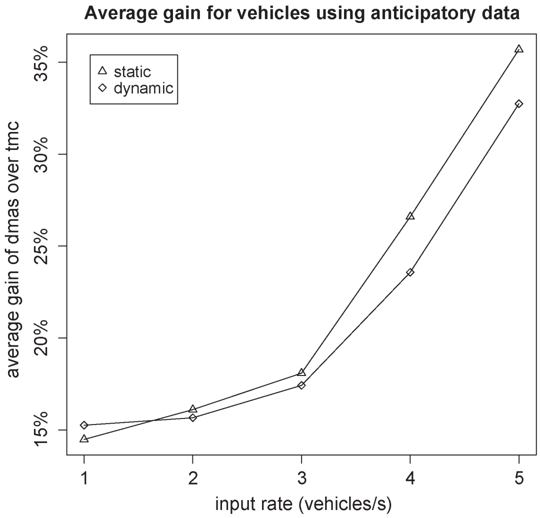
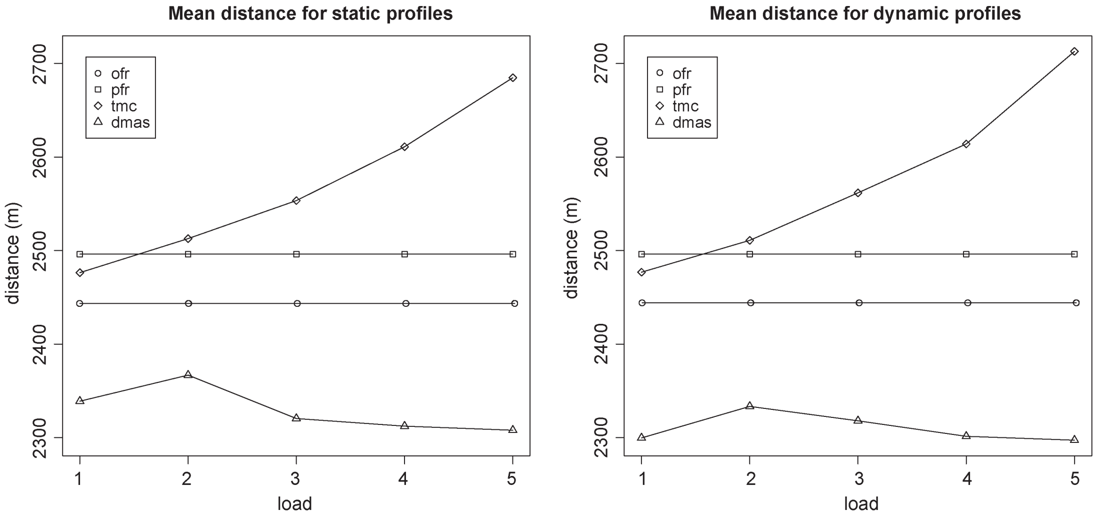


Fig. 7. Gain in average trip duration obtained by using forecast instead of real-time data.

Experiments show that only the TMC-based routing strategy and the delegate MAS routing strategy can handle increasing traffic intensity under both static [see Fig. 6(a)] and dynamic [see Fig. 6(b)] input rates. Both 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 Fig. 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, i.e., the TMC-inspired and the anticipatory routing strategies, drastically outperform stand-alone route guidance strategies.



(a) (b)

Fig. 8. Trip distances for all route guidance strategies for both static and dynamic input rates at various levels. (a) Static input rates. (b) Dynamic input rates.

B. *Trip Distance*

Although trip duration is often the main criterion 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 that results in an earlier time of arrival completely disregards fuel, maintenance, and environmental costs.

The average trip distances in Fig. 8(a) and (b) indicate that using anticipatory routing decreases trip distances. Although this approach may seem counterintuitive, it is explained by the intention update strategy of the vehicle agents. Vehicle agents that use ofr or pfr never reconsider their route, because they never receive new information. The A* algorithm using cost functions in (1) and (2) will not necessarily generate the *shortest* route but will try to generate the *fastest* route. The anticipatory route guidance algorithm also initially generates the fastest route. However, as congestion starts to form, the anticipatory route guidance algorithm starts looking for alternatives. These alternatives will often be the shorter less fast route.

Fig. 8 shows a steady upward trend in the trip distance when using the TMC-based routing strategy. One possible explanation for the upward trend that is noticeable in both static and dynamic input profiles is the staleness of the information that the TMC-based algorithm uses. This condition could cause a buildup of rerouting actions, where the TMC-guided vehicle agent deviates from its original route and starts a detour, only to find that this detour has also become congested. Such instability and inefficient decision making is predicted in [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 that involves "anticipatory vehicle routing" and see whether it describes the same problem as we do. The discussion focuses somewhat on the work on anticipatory vehicle routing done in [15] because of its resemblance to the approach that we describe here. Then, as a comparison of the delegate MAS approach, we look at a number of routing approaches that are biologically inspired, including the use of 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 that involves the term *anticipatory vehicle routing*. Some of this related work, e.g., the work of Wunderlich *et al.*, deals with a similar problem as in this paper, although their focus differs. Other work, e.g., [17] and [18], focuses on different problems such as the dispatching of pickup vehicles to meet anticipated future customer demands in a pickup and delivery problem.

The U.S. National Intelligent Transportation Systems (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, which correspond with the optimistic and pessimistic route guidance algorithms that we described in Section IV-C. This autonomous architecture is followed by a decentralized architecture in which real-time information is broadcast 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 (ISP). This ISP will then provide the vehicle with an individualized route, taking into account all other issued routes to predict future traffic states. Such a centralized architecture is expected to solve the problems predicted in [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, i.e., 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 broadcast by an ISP, the architecture is said to be decentralized. If, on the other hand, the routing decision is centrally made by some ISP and is only then broadcast to the vehicle, the architecture is considered centralized.

The route guidance system described in [15] uses forecast, instead of real-time, data in a way that is similar to our approach. The method of collecting and distributing these forecast data, however, differs from the approach that we describe in this paper. The architecture described in [15] can only be called decentralized when disregarding the role of the ISP. 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 with the work discussed in the previous section, is the decentralized propagation of vehicles' intentions using swarms. The lightweight agents in delegate MASs 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 extensively been studied by Ando in [22] and by Tatomir in [23].

1) 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 locally spread 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 is thus propagated through the environment. The information is kept up to date through evaporation. Vehicles can look ahead and sense the pheromone levels on their intended route and compare these pheromone levels with the pheromone levels 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 the approach in [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 that 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 is on 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 in [22] and [24] represents current or past traffic information, whereas the information stored in the environment by delegate MASs describes future traffic densities. The use of the environment to store pheromones and the process of propagation and evaporation are based on the work on ant colony optimization [26].

The second difference between the approach presented in this paper and the approach in [22] is the entity that deposits the pheromone trails. In our approach, because pheromones represent future information, the pheromones are deposited by antlike agents that operate in a delegate MAS. In the approach described in [22], vehicle agents deposit the pheromones, because they represent real-time traffic information.

2) Use of Swarm Computing: The use of ants in the domain of traffic are described in [23] and [27]. In [27], the authors focus more on pickup and delivery problems. The second paper [23] describes a hierarchical routing system that uses the following three different types of ants: 1) local ants; 2) backward ants; and 3) exploration ants. The function of the exploration ants in [23] differs from the function 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 MAS approach, vehicles send out intention ants to notify road agents of pending visits. Although this approach is not a reservation-based mechanism, it resembles 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, i.e., "driver agents" and "intersection agents." Here, the authors hint at the use of a learning approach similar to the approach that 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 have described a routing strategy for anticipatory vehicle routing using delegate MASs. This routing strategy can more efficiently route vehicles by using forecast information. This anticipatory vehicle information is collected and distributed in a decentralized fashion, unlike other approaches that involve 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 that scalability requirements can more easily be met than 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 with drivers who use no data or real-time data made available by TMC services.

The forecast data not only allow drivers to avoid existing congestions but prevents them from forming congestion as well.

Further research on the subject of anticipatory vehicle routing and the use of delegate MASs 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 that involve 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 the time of day and day of the week.

The use of forecast information obtained by delegate MASs comes 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 that are deployed share these communication and computation requirements.

REFERENCES

[1] D. Weyns, T. Holvoet, and A. Helleboogh, "Anticipatory vehicle routing using delegate multiagent systems," in *Proc. IEEE ITSC*, Sep. 2007, pp. 87–93.

[2] B. Chen and H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, 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)," *IEEE Trans. Intell. Transp. Syst.*, 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," *Auton. Agents Multiagent Syst.*, vol. 14, no. 1, pp. 5–30, Feb. 2007.

[5] T. Holvoet and P. Valckenaers, "Exploiting the environment for coordinating agent intentions," in *Proc. Environ. Multiagent Syst. III*, 2007, pp. 51–66.

[6] A. Rao and M. Georgeff, "Deliberation and intentions," in *Proc. 7th Conf. Uncertainty Artif. Intell.*, 1991, pp. 300–307.

[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 modeling in a commuter scenario," *Transp. Res. Part C: Emerging Technol.*, vol. 10, no. 5/6, pp. 373–398, Oct.–Dec. 2002.

[8] K. Dresner and P. Stone, "Sharing the road: Autonomous vehicles meet human drivers," in *Proc. 20th IJCAI*, 2007, pp. 1263–1268.

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

[10] Openstreetmap, Aug. 2009. [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," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[12] P. Davies, "The radio system-traffic channel," in *Conf. Rec. Vehicle Navig. Inf. Syst. Conf.*, Sep. 1989, pp. A44–A48.

[13] J. Wahle, A. L. C. Bazzan, F. Klügl, and M. Schreckenberg, "Decision dynamics in a traffic scenario," *Phys. A: Stat. Mech. Appl.*, vol. 287, no. 3/4, pp. 669–681, Dec. 2000.

[14] E. Schmitt and H. Jula, "Vehicle route guidance systems: Classification and comparison," in *Proc. IEEE ITSC*, Sep. 2006, pp. 242–247.

[15] K. Wunderlich, D. Kaufman, and R. Smith, "Link travel time prediction for decentralized route guidance architectures," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 4–14, Mar. 2000.

[16] K. Dresner and P. Stone, "Multiagent traffic management: Opportunities for multiagent learning," in *Proc. Learn. Adaption Multiagent Syst.*, 2006, pp. 129–138.

[17] B. Thomas and C. White, III, "Anticipatory route selection," *Transp. Sci.*, vol. 38, no. 4, pp. 473–487, Nov. 2004.

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

[19] Nat. Intell. Transp. Syst. Architecture, Theory of Operations, Washington, DC, 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 *Multiagent-Based Simulation VIII*, vol. 5003, *Lecture Notes in Computer Science*, L. Antunes, M. Paolucci, and E. Norling, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 36–51.

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

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

[23] B. Tatomir and L. Rothkrantz, "Hierarchical routing in traffic using swarm intelligence," in *Proc. IEEE ITSC*, Sep. 2006, pp. 230–235.

[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," in *Proc. IEEE ITSC*, Sep. 30–Oct. 3, 2007, pp. 490–495.

[25] Y. Ando, Y. Fukazawa, O. Masutani, H. Iwasaki, and S. Honiden, "Performance of pheromone model for predicting traffic congestion," in *Proc. 5th Int. Joint Conf. AAMAS*, 2006, pp. 73–80.

[26] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

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

[28] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proc. 3rd Int. Joint Conf. AAMAS*, 2004, pp. 530–537.

[29] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *Proc. 8th Int. Conf. AAMAS*, 2009, pp. 617–624.

Rutger Claes received the M.A.Sc. degree in computer science from the Katholieke Universiteit Leuven, Leuven, Belgium, where he is currently working toward the Ph.D. degree.

He is currently a Research Assistant with the DistriNet Laboratory, Department of Computer Science, Katholieke Universiteit Leuven. His research interests include the application of multiagent systems, online simulation, and swarm-based coordination in intelligent traffic systems.

Tom Holvoet received the Ph.D. degree in computer science (major in open distributed software development) from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1997.

He is currently a Professor with the Department of Computer Science, Katholieke Universiteit Leuven, where he also heads a research team with three postdoctoral and six Ph.D. students within the DistriNet Laboratory. His research interests include software engineering of decentralized and multiagent systems, coordination, software architecture, and autonomic computing.

Danny Weyns (M'08) received the Ph.D. degree in computer science from the Katholieke Universiteit Leuven, Leuven, Belgium, in 2006 for his work on multiagent systems and software architecture.

He is currently a Postdoctoral Researcher with the DistriNet Laboratory, Department of Computer Science, Katholieke Universiteit Leuven, which is funded by the Research Foundation Flanders. His research interests include software architecture, self-adaptive systems, multiagent systems, and middleware for decentralized systems.