Dynamic Road Navigation with Ant Algorithms

Sebastian Iwanowski[1] Thomas Walther

FH Wedel (University of Applied Sciences)

Feldstr. 143

D-22880 Wedel

Germany

e-mail: iw at fh-wedel.de

phone: +49 4103 80 48 63

fax: +49 4103 80 48 910 63

[1] corresponding author

Abstract

This paper discusses how to integrate ant algorithms into modern road navigation systems. We argue why traditional approaches used in existing road navigation tools are inherently too slow for the task to integrate all dynamic information necessary for a reasonable navigation advice considering the current traffic situation.

Ant algorithms belong to the AI discipline of swarm intelligence and emulate the behaviour of ants in nature. They rely on indirect communication between the participants which we show is very useful for dynamic road navigation. In particular, ant algorithms overcome the drawbacks of traditional navigation approaches by an intense application of concurrency and by the use of probabilistic information.

We show how to adapt existing ant algorithm approaches to the road traffic scenario and discuss and evaluate several algorithms. We present two own improvements for the use in maps of a size met in practice.

Keywords: pheromone-based navigation, dynamic adaptation, hierarchical routing

1. Introduction

The central task of a road navigation system is to compute an optimal path from a source to a destination. The definition of what is optimal may vary from driver to driver and also depend on the current situation. This can be considered in the assignment of individual and situation dependent cost values to the single road segments. Thus, the problem of practice can be considered an instance of the shortest path problem in graphs where the edge costs are dynamic.

While the kind of dynamics just described is dependent on the driver, the even stronger need to deal with dynamic data comes from the environment of the road network: The computation of the fastest or most convenient path requires the knowledge of the currently possible speed on all road segments that are candidates to be included in the resulting path. The currently possible speed on a road segment depends on highly dynamic data such as current traffic load or weather conditions.

Thus, the consideration of dynamic information is essential for modern road navigation systems. More precise and up-to-date information gives more useful navigation advice.

Current road navigation systems compute optimal results in a short time even on big maps. Although the underlying A*-like algorithms have quadratic behaviour, the short response time is due to the fact that the maps integrated in the cars have preprocessed information computed before installation. This is why modern road navigation systems are still not able to use real-time information for each road segment: This would require a new preprocessing of the data which is extremely costly. In the current state-of-the-art, dynamic data is only collected for a few road segments and broadcasted to all drivers. Such data is considered by the existing algorithms in a more or less crude way. In most cases, certain segments that are identified to be “congested” are just avoided. The practical usefulness of this kind of Boolean behaviour is very limited.

Another obstacle to prevent current road navigation systems from using better dynamic information is due to the fact that all current road navigation systems compute the path on an on-board-computer: If a driver wants to compute the currently shortest path from the current location to the desired destination by an on-board-computer, the car would need a transmittal of the current conditions of all road segments of the entire map, because theoretically none may be excluded. Even a confinement to road segments “in between” would still result in too much communication for each individual car.

This paper proposes to use an ant algorithm in order to overcome the problems just described. The ant algorithm should run on a central power computer and simulate all possible source/destination requests. Any ant algorithm uses so-called pheromones. In our application, pheromones are chunks of information assigned to each junction (i.e. a node) of the road network. For each possible destination, they are a probabilistic measure for the usefulness of each road segment (i.e. link) leaving the respective junction. Each car needs to load the current pheromone values only of the junction just ahead. This reduces the communication overhead dramatically and still gives very accurate results considering the latest information.

The reason for this can be seen by the following: In the traditional approach, each single car has to load all relevant information by itself in order to compute the current optimal path on its on-board-computer. The ant algorithm approach has the advantage that the current situation is continuously updated from the ant algorithm on the central power computer in an eager way, i.e. independent from any request. This makes it redundant for each car to load all dynamic information needed separately, but instead only requires retrieving the current pheromone values stored for the junction just ahead.

Ant algorithms have been invented by Marco Dorigo (1991) (cf. also Dorigo et al., 1992) for the application in computer networks. Since then, an active community has improved the underlying algorithms and adapted to other applications, e.g. also to road networks. We tested the currently known variations in a traffic simulator and identified further issues to be developed before ant algorithms can be applied in real road navigation tools.

In this paper, we first outline how natural and artificial ant algorithms work in general and discuss their particular advantages for the road navigation application. Then we outline the algorithmic varieties which we actually have tested. Afterwards, we show hierarchical approaches necessary to cope with the complexity of real maps. Finally, we sketch how to manage the actual communication between the individual cars and the central power computer.

2. Natural ant behaviour

The path finding problem real ants are involved in is to find the fastest (or most convenient) path from their hill to a food supply. In general, all ants swarm out from the hill in arbitrary directions and eventually find some food. After having collected enough food or other material, each ant returns to its hill.

In order to keep orientation, each ant drops some chemical markings, the pheromones, on the path it takes. Every ant gets biased by all pheromones it passes on its way: In principle, each ant tries to follow an intense pheromone path, i.e. a path that has been used by a lot of other ants before: The more intense a pheromone path, the more likely a succeeding ant would follow. So simple it is, this makes ant navigation rather efficient: If a bunch of ants finds a faster path to a food supply than others, these ants would return earlier (using the pheromone path of their way there), and, thus, double the pheromone path on their return path before the slower ants get their chance. This would automatically increase the pheromone intensity on the shorter path, and, by continuous feedback, steadily increase the probability that succeeding ants follow this path.

The reason why ants adapt to dynamic changes very well is due to the fact that ants are living animals and not deterministic machines: Some individuals still follow less intense paths or investigate even totally new paths just for curiosity. If now a traditionally good path deteriorates, the ants of the minority investigating “unusual” paths would eventually return faster than the ants of the majority following the traditional path. This will steadily improve the pheromone intensity of the new path and eventually switch the behaviour of the majority. Since the pheromones are set and updated continuously, the swarm reacts always to the latest development of the network.

Since pheromones are chemical matters, they evaporate with time. This makes newer pheromones more intense than older ones which automatically gives a preference to paths found more recently. This guarantees that a new path which better fits the current situation than the old one is found not only eventually, but also rather quickly.

As a consequence, although individual ants are rather simple animals, the swarm can find convenient paths very efficiently even in highly dynamic networks. The individual ants are steered by the following principles:

1) Each ant marks its path continuously by setting pheromones
2) At a junction, the probability that a succeeding ant would follow a certain alternative is proportional to the actual pheromone concentration.

Considering the whole community, this kind of swarm intelligence obeys the following rules:

1) Communication between individual ants is only indirect via pheromones, but never uses a direct ant-to-ant conversation.
2) Pheromones can only be read locally by the ants that actually pass the spot.
3) Pheromones deliver local information that is only relevant for ants that are in the direct neighbourhood.
4) Pheromones evaporate with time.

Rule 1) reduces the communication overhead that would result from a point-to-point conversation. Rule 2) guarantees that communication still works when the individual communicators only have a local scope. This suggests working with kind of local area networks. Rule 3) shows that it is possible to store pheromones in locally assigned regions. The transfer of this concept to the road navigation application makes it possible to dispense with the need of direct communication between each individual participant and the central power computer. Rule 4) guarantees that newer pheromones have got a higher priority than older ones which supports a better convergence to newer information.

3. Artificial ant behaviour

Artificial ant algorithms should acquire the advantages of natural ant navigation for arbitrary routing purposes. The original purpose of artificial ant algorithms was package routing in a computer network. Later, also car navigation in road networks had been considered. The crucial difference between these two applications is that packages are usually not driven by intelligent units, while cars are driven by intelligent drivers who always want to be in ultimate control. It is therefore not acceptable that the cars assume the function of the ants directly, because drivers are most likely not willing to be forced according a probability distribution. Instead, the natural ants are replaced by artificial software units in a central power computer which performs a continuous simulation of the network.

Nevertheless, all artificial ant algorithms share the following principles with natural ants:

1) Optimal paths are found by a colony of cooperating individuals.
2) Communication is only indirect via “pheromones”.
3) There is a continuous move between bunches of individuals just like in an ant hill.
4) The route finding decision of each individual is stochastic and locally confined.

For the sake of practicability and convergence accelaration, the following extensions have been made for all artificial ant algorithms:

5) The world of moves and state transitions is modelled in a discrete way.
6) Each artificial ant is allowed to have a memory (the natural ones do not!).
7) The modification of pheromone intensity is directly correlated to the quality of the solution (in natural systems the correlation is only indirect).
8) Pheromone dropping may be retarded or inhibited.
9) The ants may have additional problem specific skills.

The general scenario for the road navigation application works as follows:

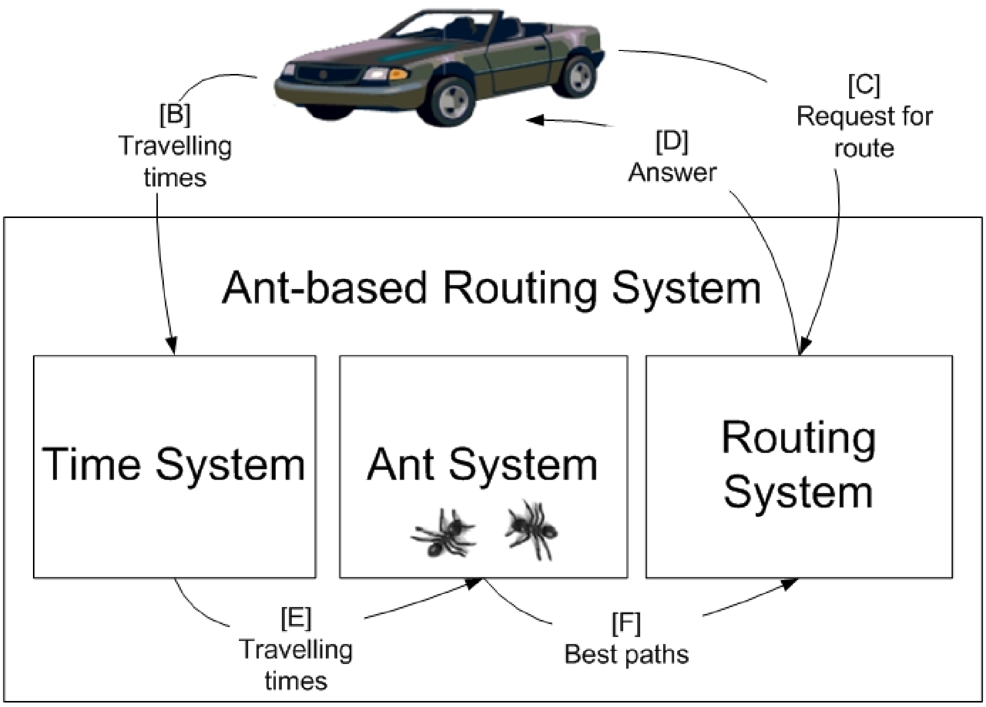


Figure 1: Components of the ant-based routing system

The modules of the ant-based routing system are shown in Figure 1. All of them are performed on a central power computer. Recall that the artificial ants are not identical with the individual cars that should be navigated, but rather implemented by simple identical locally distributed software units. This is performed by the module Ant System in Figure 1. The central power computer continuously receives all kind of info for each individual road segment which should always include the current throughput time. This info is collected by the module Time System in Figure 1. It is out of the scope of this paper how this info is gathered. It would be definitely a good idea to get this info from cars that are actually travelling on the road segments ("fleet net scenario") as it is suggested by Figure 1. But our system does not depend on this innovative approach which so far is only implemented in prototypic systems. The module Time System can be fed with any kind of state-of-the-art road segment information retrieval.

In module Ant System, for each relevant source/destination pair, a bunch of artificial ants is started continuously to simulate the moves of real cars. Each ant has a predefined starting time, source, and destination. The equivalent to the pheromones is the following: At each node and each outgoing road segment and each destination of the network, a probability is given for this segment to be the currently best one in order to reach the respective destination from the respective node. Thus, the pheromones are organized in a table similar to the routing tables of computer networks. Note that the huge number of all possible source/destinations pairs of the network is decreased by hierarchical approaches which will be discussed later.

The general navigation principle of module Ant System is as follows: There is only one pheromone value for each outgoing road segment and each destination, no matter how many ants have passed the node. Each ant leaving the node on a certain outgoing road segment heading towards a certain direction will update the respective pheromone value only after it has reached its destination. Thus, pheromone dropping is retarded. The updating is performed by so-called *backward ants* corresponding one-to-one to the original moving ants (the so-called *forward ants*). These backward ants have a memory of the travel times of the corresponding forward ants. This technique extending the capability of natural ant behaviour accelerates the convergence to good paths. The natural analogue would be that each ant returns always on the same way it took there disregarding the pheromone concentration on the return path. So far, such behaviour could not be proved in nature.

While all artificial ant algorithms follow the general principle just described, they differ in the way how exactly a pheromone value is updated. The details are given in the next chapter. All implementations have in common that the pheromone value of the road segment which was chosen by the forward ant is increased dependent on the travel time the forward ant needed to the final destination. The evaporation of natural pheromones is emulated in the following way: Each backward ant decreases the pheromone values of all road segments which were *not* chosen by the corresponding forward ant. Again, the details depend on the actual implementation of the ant algorithm which is elaborated in the next chapter.

Since the artificial ants are continuously generated in representative numbers on the central power computer and always take the time on the road segments that is currently valid, the so-generated pheromones reflect the current traffic situation towards the respective destination.

When a car navigating in the actual road system approaches a node, it reads the currently valid pheromone values from the central power computer and takes the direction of the currently most probable direction. The interface to the actual cars is provided by the module Routing System in Figure 1. In a first shot, Routing System provides exactly the current pheromone values for each junction, but it is not prohibited to refine this somehow. Note that the receiving car is totally independent of the Ant Routing System and may choose its own direction depending on an individualized strategy. But the current pheromone value will be a good base for that decision.

Even without showing the details, three advantages to current navigation systems can already be seen:

1) The pheromones of the alternative junction branches collect the info of many road segments ahead towards the destination in one value: This reduces the amount of data to be considered dramatically.

2) The computation of the current situation is performed in parallel on a central computer prior to any actual request: The central computer enables a much higher capacity than a local on-board computer. Furthermore, the information is already computed when it is requested.

3) Dynamic information distributed over the whole country is collected in one central computer, processed, and then forwarded to individual units again distributed over the whole country. Thus, the central computer plays the role of a middleware in a distributed system. This grants a decreased need for interfaces and communication compared to a direct communication of all dynamic information between all participants.

4. Realization techniques for the basic ant algorithm

The work for this chapter has been complemented and tested at FH Wedel by various prototypes in Java.

In scientific research, the general idea of ant algorithms has been adapted in various ways to solve optimization problems in different domains (cf. Dorigo et al. 1999, Dorigo and Stützle 2004). In the domain of packet routing for computer and telecommunication networks two different approaches – the AntNet algorithm (cf. Di Caro and Dorigo 1997, 1998) and the Ant Based Control algorithm (ABC, cf. Schoonderwoerd et al. 1996) – have been developed. Since road networks and computer networks can be modelled by a graph consisting of nodes and links, these algorithms can be easily adapted and used for the navigation of cars. This adaptation has already been performed for the ABC algorithm (cf. Kroon 2000, Tatomir et al. 2004). In the following chapter, we give a more detailed view on the AntNet algorithm and elaborate why we prefer the AntNet algorithm to the ABC algorithm.

Pheromone matrix

As already described in the previous chapter, forward ants and backward ants are used by both algorithms for searching continuously optimal paths for any pair of source and destination. The main difference between the two algorithms is the information that is stored for each node of the graph and is utilized by the forward ant to find a path from its source to its destination. The AntNet algorithm uses a pheromone matrix and a local statistical model for the computation of the probabilities which are the base for the ants' choice how to proceed towards the given destination. The pheromone matrix of a node n_i is two dimensional and consists of one column for each adjacent node n_j and one row for each node n_d of the entire road network. A pheromone value τ_{jd} stands for the plausibility to take the link to node n_j after node n_i when n_d is the final destination. By equation 4-1, the pheromone values are normalized for each row. N_i contains the indices of the node neighbors adjacent to n_i. All the pheromone matrices together are building the pheromone trails from any source to any destination. The pheromone matrices can be considered as the distributed memory of the whole ant colony.

$$\sum_{j \in N_i} \tau_{jd} = 1$$

equation 4-1

Statistical model

In addition to the pheromones, a statistical model of travelling times collected by the forward ants is computed for each node. The statistical model $M\ (t_{best}, \mu, \sigma^2, W)$ reflects the current traffic situation experienced by the forward ants. This model will be used later for updating the pheromones with respect to the quality of a path found by a forward ant. For the model, the fastest travelling time t_{best}, the average μ and variance σ^2 of all travelling times are computed. A sliding time window *W* is defined to gradually eliminate the influence of out-dated data. The size of the time window has an impact on the reactivity of the system to react on dynamic changes of the traffic situation. Again, these statistical values are collected for each pair of destination and outgoing road segment, i.e. a matrix similar to the pheromone matrix is build.

Pheromone-based path selection

A forward ant travelling through the network graph has to decide at each node which road to take next. This decision shall be based on the pheromones dropped by other ants. To reflect the behaviour of natural ants a probability value P_{ijd} is computed for the pheromone values for each link leaving the node (cf. equation 4-2).

$$P_{ijd} = \frac{\tau_{ijd} + \alpha\eta_{ij}}{1 + \alpha\left(|N_i| - 1\right)}$$

equation 4.2

The AntNet algorithm interprets this equation as follows: An ant at the node n_i with destination node n_d will select the node n_j with the probability of P_{ijd}. $|N_i|$ is the number of adjacent nodes. In addition to the pheromones, the AntNet-algorithm takes an additional problem-specific, heuristic quantity into consideration for the computation of the probability. The heuristic quantity η_{ij} is calculated as defined in equation 4-3. The influence of η_{ij} is weighted by a factor α between zero and one. This heuristic quantity can be computed of any dynamic or static information *q* known about the road segments, e.g. current state of congestion.

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum_{l=1}^{|N_i|} q_{il}}$$

equation 4-3

The somewhat complicated computation of P_{ijd} in equation 4-2 guarantees that the probability to take a certain road segment is always greater than zero, which is essential for the system to react on dynamic changes in the traffic system: If the probability for any road were zero, no ant would use this road anymore. And therefore, the biological behaviour of ants to discover new paths wouldn't be reflected by the artificial colony correctly.

In order to find best routes and to discover new routes, forward ants are travelling continuously through the road network and collecting information about their paths. In our implementation, we restricted this information to the actual travelling times on the road segments, i.e. each forward ant memorizes how long it needed to pass each road segment on its path.

Note that in the traffic scenario where the ants are not identical to the moving cars (cf. Figure 1), the computation time of the tour of an artificial ant does not take as long as the actual time for the jouney of the emulated car: The presently valid traveling times for the route segments are simply collected from the database and added which can be performed in milliseconds for each "travelling" ant through the whole country.

Updating the pheromone tables

When a forward ant has reached its destination, a backward ant is started to update the pheromone matrices and statistical model of each node visited by the forward ant. Based on the forward ant's memory the backward ant travels the same way back to the starting point of the forward ant. At each node, the statistical model is updated considering the total travelling time from the current node to the destination. The update should have two effects: If *f* was the index of the link taken by the forward ant heading do the direction with index *d*, then τ_{fd} should be inforced a little bit and the other values τ_{jd} (for $j \neq f$) should be diminished a little bit resembling the evaporation of the competing pheromones. AntNet performs this task using equation 4-4 for pheromone enforcement and equation 5-5 for pheromone evaporation.

$$\tau_{fd} \leftarrow \tau_{fd} + r \cdot (1 - \tau_{fd})$$

equation 4-4

$$\tau_{jd} \quad \leftarrow \quad \tau_{jd} - r \cdot \tau_{jd}$$

equation 4-5

r is an update factor which is based on the total travelling time and the current state of the statistical model (cf. equation 4-6):

$$\begin{aligned} r &= c_1 \left(\frac{T_{id_{best}}}{t_{id}} \right) + c_2 \left(\frac{I_{sup} - I_{inf}}{(I_{sup} - I_{inf}) + (t_{id} - I_{inf})} \right) \\ I_{sup} &= \mu_{id} + z \left(\frac{\sigma_{id}}{\sqrt{w}} \right) \\ I_{inf} &= T_{id_{best}} \end{aligned}$$

equation 4-6

Note that this sophisticated recalculation correlates the new pheromone values with the quality of the discovered path: The faster the destination was reached, the greater the pheromone value. The statistical model sets the requirement that good paths must be confirmed by several ants in a given time window laid upon the current travel period of each individual ant.

Compared to the AntNet algorithm, the ABC update mechanism is much simpler: ABC does not use a statistical model for computing pheromone values to be dropped. This is the major advantage of the AntNet-algorithm. Furthermore, in the ABC algorithm, backward ants consider the total travelling time experienced by the corresponding forward ant for the computation of the actual pheromone value to be dropped. But this travelling time is not compared to the best travelling time experienced before. Therefore, the pheromone value depends on the absolute quality of the travelling time and not on a time relative to the best travelling time. This is a drawback of the ABC algorithm, because long travelling times imply always small pheromone updates. But if we are looking on destinations far away from the source we will have always long travelling times. This is why the ABC algorithm has difficulties to form pheromone trails to long distance destinations. Due to its more sophisticated statistical model, the AntNet algorithm considers the relative improvements.

5. How to cope with the complexity of real maps

For small networks as implied in prototypic settings, experiments and scientific research have proved that ant algorithms are very useful to find shortest and fastest ways. But it can be shown that the performance of ant algorithms degrades with increasing complexity of the underlying network (cf. Kroon 2000). This chapter presents two approaches to manage the complexity of real road networks. First, a hierarchical ant system introduced by Dibowski 2003 and advanced by Tatomir et al. 2004 is examined. This approach resembles very much the current state-of-the-art of the OSPF internet protocol for package routing in large computer networks (RFC 2328 (1998), for an easier reference, cf. Kurose and Ross 2008)).

Then we propose an alternative approach using dynamic sectors for remote destinations.

5.1 Hierarchical approach like in large computer networks

In Dibowski 2003 and Tatomir et al. 2004, a hierarchical model was introduced. The authors applied this model to the ABC algorithm.

The road network is separated into sectors and two classes of roads. Minor roads are connecting nodes within a sector and major roads are building connections between sectors. This means, if you are travelling from one sector to any another sector you must use a major road. Furthermore, if you are driving on a major road you cannot use any minor road until you have reached the sector enclosing your destination. A sector can be left only at dedicated junctions connecting minor roads and major roads. Such junctions are connecting the two levels of the hierarchical model. This simple model can be easily extended to more than two levels. The basic idea of the hierarchical model is shown in Figure 2.

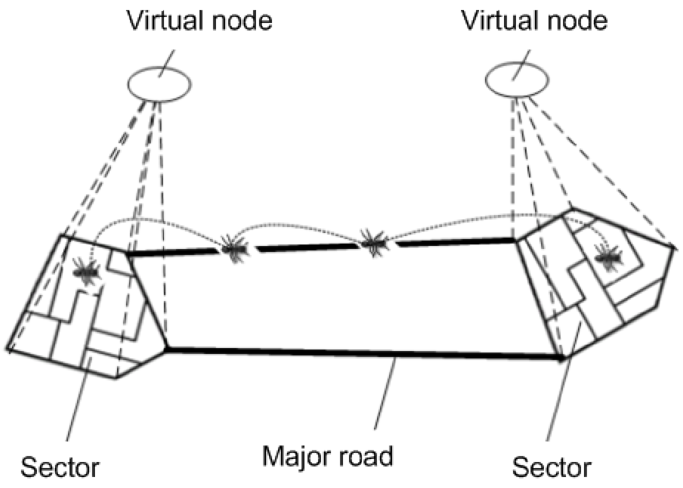


Figure 2: Ants moving through a hierarchical model of the road network

The underlying ant-algorithm (AntNet or ABC) needs to be adapted to support the hierarchical level. The concept how forward ants and backward ants are discovering and establishing best ways is not changed. But the information stored at each node is adapted to reflect the hierarchical model. Therefore, virtual nodes *V* are introduced, where each sector of the model is represented by one virtual node *V*. Furthermore, three different types of nodes are defined to control the movement of ants:

1) Local nodes: Nodes within a sector. At this node an ant can see all local nodes of the current sector and all virtual nodes.
2) Top-level nodes: Nodes that can only be reached via major roads. At this node an ant can see all other top-level nodes and also all virtual nodes.
3) Border nodes: Nodes that are connecting sectors with the top-level of the model. At border node connecting sector SA with a major road an ant can see all local nodes of SA, all top-level nodes and all virtual nodes.

Note that these types directly correspond to the different router types of the OSPF standard (RFC2328, 1998).

Since ants are using information only stored locally at each node, these three types of nodes can be implemented by modifying the pheromone tables stored at each node as shown in Figure 3. The table of a top-level node contains only rows – representing any possible destination – for virtual nodes *V* and top-level nodes *T*. A local node consists only of rows for all local node *s* belonging to the same sector and for all virtual nodes. Border nodes are essential for the hierarchical model. Their table contain a row for each top-level node, a row for each virtual node, and a row for each local node of the current sector, that can be left or entered via this node. Some links are blocked (indicated by “-“) to force the ants to use only major roads if they want to travel to a top-level node or virtual node. For example, in Figure 3 it is not allowed to take the link to *s3* if the ant’s destination is *T1*.

Toplevel node

	T3	T4
T1	0.3	0.7
T2	0.5	0.5
...		
Tn	0.4	0.6
V1	0.1	0.9
...		
Vm	0.65	0.35

Local sector node

	s3	s4
s1	0.3	0.7
s2	0.5	0.5
...		
sn	0.4	0.6
V1	0.1	0.9
...		
Vm	0.65	0.35

Border node

	s3	T3	T4	G1	G2
s1	0.7	-	-	0.3	-
s2	0.8	-	-	0.2	-
...					
sn	0.2	-	-	0.8	-
T1	-	0.1	0.7	0.1	0.1
...					
Tk	-	0.2	0.6	0.1	0.1
V1	-	0.6	0.2	0.1	0.1
...					
Vm	-	0.6	0.2	0.1	0.1

Figure 3: Routing tables for the different type of nodes defined in the hierarchical model

Ants are still started continuously at each node. But the ant’s task to find good routes is slightly different for each type of node.

1) An ant started at a local node can find best routes to other local nodes of the same sector or to any virtual node.
2) An ant started at a top-level node can discover routes to other top-level nodes and to virtual nodes.
3) An ant started at a border node is responsible for establishing best ways to local nodes of the connected sector, virtual nodes, and top-level nodes.

As a consequence, best routes between two nodes of two different sectors are composed of two results. First, the best path to the virtual node is searched. This means, the best path to any border-node connected to the destination sector is computed first. Then starting from this border-node, the best route to the final destination is calculated. Together, these two parts build up the best route.

But composing these two parts does not imply that the resulting path is the best to get from the starting point to the destination. This problem is due to the fact that the final destination is not considered at the starting point (cf. Hock and Srikanthan 2001). This problem was also mentioned in Dibowski 2003. While in computer networks this problem is not a real practical obstacle for non-intelligent packages, we consider this a problem when we apply this to intelligent drivers.

In the following, we present an adaptation of the hierarchical model just described that overcomes the problem.

It is necessary to optimize the selection of the border node which is taken to leave the starting sector. The hierarchical model and the behaviour of the ants must be changed.

	s3	s4
s1	0.3	0.7
s2	0.5	0.5
...		
sn	0.4	0.6
(n1,V1)	0.1	0.9
...		
(|V1|,V1)	0.65	0.35
...		
(|Vm|,Vm)	0.4	0.6

Figure 4: Modified routing table of local sector nodes

The structure of the border nodes and top-level nodes need not be changed but the local nodes have to be adapted. For each local node *n* assigned to a virtual node *V* one row linked to the pair *(n,V)* is added to the routing table as shown in Figure 4. Additionally to this new representation, each local node of the same sector is represented by a simple entry as defined before. For the AntNet algorithm (which we want to apply instead of the ABC algorithm), the statistic tables must be changed similarly. The functional behaviour of ants starting at a local node and travelling to a local node of a distant sector is modified as follows:

1) The forward ant memorizes the destination pair *(n,V)* and starts to find a path to *n*.
2) At each local node of the starting sector the ant makes its decision based on the pheromone table's row describing the destination *(n,V)*.
3) At a border node, the ant verifies if a row for the node *n* exists.
4) If yes, it will take this row for its next step.
5) If not, the ant looks up the row linked to the virtual node *V*.
6) At a top-level node, the ant bases its next step on the row linked to the virtual node *V*.
7) At each local node belonging to the destination sector, the ant makes its decision based on the row linked to the destination node *n*.
8) At the destination node, a backward ant is created. The backward ant travels the same way back and updates the pheromone table at each node. It updates only table rows selected by the forward ant.

Now, the optimization of the routes is not composed of two intermediate results anymore. The destination node is taken into account at each node on the ant's way. The changes described here can be adapted easily to models with more than two levels.

Note that this quite simple improvement makes use of the fact that in ant systems also suboptimal paths are explored automatically. By the feedback of the backward ants, a path that did not appear optimal from a local sector's view will automatically have the better pheromone trail if this is really better to achieve the respective location in the destination sector.

A drawback of the described approach is the increased complexity: At local nodes, the number of rows has been increased for each table. Therefore, the solution space is enlarged, more ants are needed to discover and establish best routes. But this trade-off may be acceptable.

5.2 Dynamic Sector Routing

In the following, we give a conceptual alternative to the hierarchical model described before. The concept is based on a rough view over the complete map as this would be done by an intelligent driver. While the artificial ant (in place of the driver) should have a very detailed view on all the nodes located next to its current position, it should see distant nodes only very vaguely. The static sectors of the hierarchical model described before are replaced by dynamic sectors dependent on the current location of the respective ant.

This concept also solves another drawback of the hierarchical model explained in the previous section: In the hierarchical model, an ant cannot leave a road of the top-level until it has reached its destination sector. But on these roads traffic jam or congestion can also occur. An experienced driver knowing the current condition of the traffic network would often use minor roads to avoid such congested roads. Furthermore, our modified version of the hierarchical model has a feature that may be a drawback in real applications: The selected route is based on the current traffic situation in the neighbourhood of the destination. But during a long lasting ride the traffic situation might completely change at the destination. In the following, we present a model to trade-off between the hierarchical model introduced by Dibowski 2003 and our modified version thereof presented in the previous section.

A human driver would consider his directions for a long distance trip as follows: At start, the driver selects a route towards a known landmark (e.g. a big town) in the neighbourhood of his accurate destination. For example, a driver starting at Hamburg and heading to a small village located 50 km south of Berlin (which is 300 km south east of Hamburg) will first drive towards Berlin. But coming closer to Berlin, he will have a closer look onto the map in order to select his final route. In case of congested roads ahead to his accurate destination, he would look for a bypass to avoid the congested part. But being far away from the congestion, he would not consider it as long as the road towards the rough goal Berlin is free.

This behaviour is mapped to artificial ants by the following:

1) At any node, an ant can always select any link branching from this node (i.e. it is not restricted to a certain level).
2) An ant has to know the accurate map with all nodes and links in the neighbourhood of its current position.

3) For parts of the map that are far away, the ant knows reference nodes only and major links connecting them.

The first requirement is already met by the simple model without any hierarchical level. For the other requirements, the model of the nodes must be redefined. The road network will be separated dynamically into sectors depending on the current position of an ant.

Sector definition

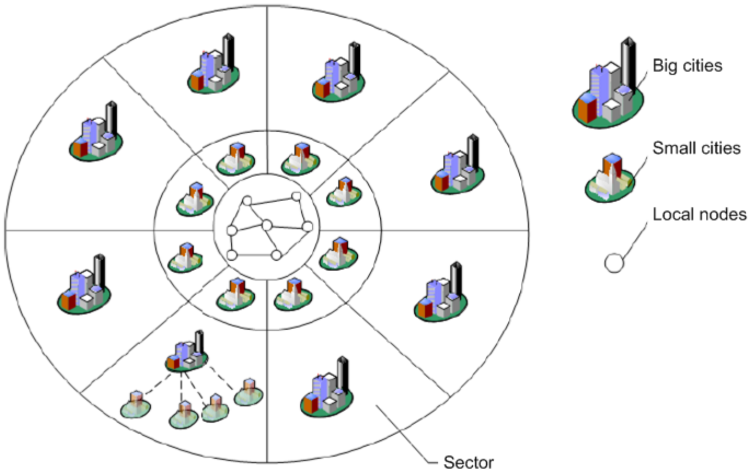


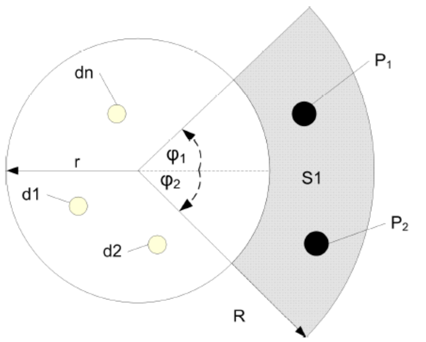
Figure 5: Model of the road network for the dynamic-sector-based ant system

The decomposition of the road network into sectors is schematically shown in Figure 5. The road network is partitioned into circular segments with the respective ant being in the centre. The network is decomposed into three levels of detail. Each level is confined by a circle. Within the inner circle, all nodes are visible. Additionally, the circles of the second and third level are decomposed into directional parts and only small (level 2) and big cities (levels 2 and 3) are visible within these sectors. The cities just symbolize reference nodes; junctions of major roads can be used as well. These nodes are similar to the virtual nodes of the hierarchical model described in the previous section, but this time the role of a node is not static, but changes depending on the centre of view.

There is only one more information, we need to add to the representation of a node compared to a non-hierarchic network: Each node has a flag defining in which sector level this node may still be visible. This does not yet define the real visibility, but only a potential for visibility. The real visibility is dynamically assigned depending on the position of the respective ant.

Adaptation of routing tables

The functionality of the routing table is not changed. It still defines which node should be visited next considering a given destination. But the definition of the rows must be redefined for the new model. In Figure 6, a routing table of a node is shown. Each column defines an adjacent node and each row defines a potential destination. There are rows for nearby nodes *{d1, d2 … dn}* and there are rows for distant sectors *{S1, S2 … Sm}.* Distant sectors are given by pairs (S, P) where S defines the sector and P defines a reference point assigned to the sector. In this representation, sectors of level 2 and 3 behave the same way. They only differ in the respective radius. A sector may contain several or no reference points. Figure 6 shows one sector with two reference points assigned. Therefore the routing table contains one row for *(S1, P1)* and one row for *(S1, P2)*. A sector is defined by the delimiting angles and the delimiting radii. In the example of Figure 6, $S1 = \{\varphi_1, \varphi_2, r, R\}$ where the φ_i describe the delimiting angles, r the lower and R the upper delimiting radius. In the example of Figure 6, for a better visibility φ_1 is positive and φ_2 negative. But in general they may be arbitrary polar coordinate angles.



	s3	s4
d1	0.3	0.7
d2	0.5	0.5
...		
dn	0.4	0.6
...		
S1,P1	0.65	0.35
S1,P2	0.4	0.6
...		

$S1 = \{\varphi_1, \varphi_2, r, R\}$

Figure 6: Node routing table and sector definition in the dynamic-sector-based ant system

For the AntNet algorithm, the pheromone table and statistic table are adapted accordingly.

Making the Ant-based routing system work for dynamic sectors

Forward ants and backward ants are still created continuously at each node. But only for the inner circle of the creation node, all containing nodes serve as randomly assigned destinations for the ants. For the sectors of level 2 and 3, only the reference points serve as destinations.

The functionality of the Ant System module as described in Figure 1 may remain unchanged to the basic algorithm. Our generation policy will automatically reduce the number of ants compared to a nonhierarchical approach.

But the Routing System module needs to be changed: When the pheromone value of a node belonging to a higher level sector is requested, no entry exists for this node. Instead, the entry of the corresponding reference point is returned. If there are several reference points available, the reference point closest to the respective node is taken. If no reference point is available, the closest reference point in an adjacent sector is taken.

Following this algorithm, we have reduced the complexity compared to the basic AntNet algorithm. The number of ants to be started at each node has been decreased, due to the definition of sectors. Compared to the hierarchical model presented before, the complexity has been increased, because the number of roads is not restricted by different levels. But we consider this as an advantage because now we can avoid congested roads at any time. Due to the sliding sector definition we have always good information about the traffic situation in the neighbourhood of a car's current location. Therefore it is possible to discover and establish bypasses in case of congestion or traffic jams that are really relevant for the current position.

6. Communication issues

In the road navigation scenario described at the beginning of this paper (cf. Figure 1), the communication between the individual cars and the central power computer (in particular parts [C] and [D]) may be a bottleneck: In order to achieve the latest relevant information for an optimal path, each car has to retrieve the pheromones of the next junction ahead. If this is performed by all cars travelling through a country simultaneously, it has the same effect as a denial-of-service attack and will definitely knock out the communication interface. Another issue is the load and resulting cost of the mobile communication network.

Since the pheromones needed by a car refer to the local area in which the car is presently situated, it makes sense to store them also physically in that area and not only in the central power computer. The smartest way as suggested by Iwanowski and Matylis 2002 is not to install a separate infrastructure for the pheromone storage in the regions but rather store them in the cars participating in the dynamic navigation system. In his master thesis elaborating details of this idea, Suthe 2007 showed that the size of all pheromones belonging to a cell of a mobile cell phone provider easily fits into the existing on-board units of a regular navigation system and that the transfer of such data can be performed within seconds with modern UMTS technology. In the same thesis, it was also shown that the pheromones belonging to all of Germany would definitely not fit into such on-board units. Following the general idea of Iwanowski and Matylis 2002, Suthe 2007 proposed that the whole map is partitioned into local zones (that may be identical to the cell phone areas). For each local zone, at least one car is elected to retrieve the current pheromones of that zone polling the central power computer in regular intervals (e.g. one minute). The pheromones are then distributed to all other cars of the zone by a broadcast technique. For the broadcast, it is possible to apply local distribution technology such as ad-hoc networks. The election of the car requesting to the central computer can be performed by a distributed technique such as the bully algorithm. Details are beyond the scope of this paper. They are elaborated in Suthe 2007.

Summarizing, the communication load required by the ant algorithm navigation proposed in this paper, can be managed by the distributed election of single cars for each zone which should poll all pheromones relevant to that zone from the central power computer. The pheromones are then distributed to all cars of that zone by a distributed forwarding technique.

7. Conclusion

In chapter 1, we argued that dynamic road navigation considering current disturbances as congestions cannot be performed in a satisfying way when traditional techniques are applied known from static road navigation.

The proposed ant net scenario described in the subsequent chapters with increasing level of detail is an intelligent alternative to the navigational techniques existing in practice. The main advantage is that ant systems are self adapting to the respective current situation. The way we proposed that ant algorithms should be applied yield also the advantage that information is readily available at the time requested. This is achieved by the use of a central power computer. In the previous section, we outlined how to cope with resulting communication problems.

The main contribution of this paper is the elaboration how ant algorithms should be actually performed in the road navigation scenario. We evaluated results given in the literature and showed how to improve approaches for a hierarchical modelling which is essential for the application in large maps. We gave two alternative improvements: The first alternative improved an idea developed at the University of Delft (Dibowski 2003 and Tatomir et al. 2004) which reflects up-to-date protocols for computer networks. The second alternative, dynamic sector routing, reflects the natural behaviour of an intelligent driver.

Parts of this work have been implemented in Java and showed promising results in a prototype setting. However, we have not yet applied our ideas in a field test. Furthermore, the ideas given by dynamic sector routing need to be further elaborated in detail. A more detailed version of this paper (but in German) is given in Walther 2006.

Ant algorithms belong to the AI branch of swarm intelligence and emulate the behaviour of ants. Other techniques of swarm intelligence emulate the behaviour of bird or fish swarms. These are quite different from the algorithms described here, because ant algorithms rely on indirect communication between the participants via pheromones. Conversely, bird or fish algorithms rely on direct communication such as ad-hoc-networks. Their application is not for the solution of a shortest path problem as we intended to do, but rather for the problem how participants can stick together for various reasons. This may also have interesting applications in road navigation, but this is beyond the scope of this paper.

References

Dibowski: Hierarchical routing system using ant based control, Master thesis, Delft University of Technology / Dresden University of Technology, 2003

Di Caro, Dorigo: AntNet: A mobile agents approach to adaptive routing, Research Report, Université Libre de Bruxelles, 1997

Di Caro, Dorigo: AntNet: Distributed Stigmergetic Control for Communication networks, *Journal of Artificial Intelligence Research 9* (1998), pp. 317-365

Dorigo: Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992

Dorigo, Maniezzo, Colorni: Possitive feedback as a search strategy, Research Report, Politecnico di Milano, Italy, 1991

Dorigo; di Caro, Gambardella: Ant algorithms for discrete optimization, *Artificial Life vol. 5 no. 2*, 1999

Dorigo, Stützle: Ant Colony Optimization, MIT Press, 2004, ISBN 0-262-04219-3

Hock, Srikanthan: Improving hierarchical route computations for roadway networks, *Proceedings of IEEE Intelligent Transportation Systems,* 2001, pp. 112-117

Iwanowski, Matylis: Verkehrskoordination mit mobiler Datenhaltung (in German), Patent application DE 10050675, 2002

Kroon: Dynamic vehicle routing using Ant Based Control, Master thesis, Delft University of Technology, 2000

Kurose, Ross: Computer Networks, Pearson Education, 2008 (4th ed.), ISBN 0-321-49770-8

RFC2328: IETF standard for the OSPF-2 internet protocol,1998, download: http://www.ietf.org/rfc/rfc2328.txt

Schoonderwoerd, Bruten, Holland, Rothkrantz: Ant-based load balancing in telecommunication networks. *Adapt. Behav.* 5 No 2, 1996, pp. 169-207

Suthe: Mobile Nutzung von Pheromontabellen für dynamische Fahrzeugnavigation (in German), 2007, download: http://www.fh-wedel.de/fileadmin/mitarbeiter/iw/Abschlussarbeiten/MasterarbeitSuthe.pdf

Tatomir, Dibowski, Rothkrantz: Hierarchical routing in traffic networks, Research Report, Delft University of Technology, 2004

Walther: Dynamische Fahrzeugnavigation auf Basis von Ameisenkolonien (in German), Master thesis, FH Wedel University of Applied Sciences, 2006, download: http://www.fh-wedel.de/fileadmin/mitarbeiter/iw/Abschlussarbeiten/MasterarbeitWalther.pdf