

# Hierarchical routing in traffic networks

Bogdan Tatomir <sup>ab</sup>      Henrik Dibowski <sup>c</sup>      Leon Rothkrantz <sup>ab</sup>

<sup>a</sup> Delft University of Tehnology, Mekelweg 4, 2628 CD Delft

<sup>b</sup> DECIS Lab, Delftechpark 24, 2628 XH Delft, The Netherlands

<sup>c</sup> Dresden University of Technology, Germany

## Abstract

In this paper the design of a dynamic Routing system, called Hierarchical routing system, has been developed. It splits traffic networks into several smaller and less complex networks by introducing a hierarchy. The Hierarchical routing system therefore consists of several distributed Routing systems where each of it is responsible for one network of the hierarchical network. The route optimization is done with an adapted version of the AntNet-algorithm, a decentralized routing algorithm, which uses intelligent agents that explore the network and find the shortest routes in time. This algorithm was derived from the behaviour of ants in nature.

## 1 Introduction

Nowadays in many countries the car traffic is a very big problem. One of the reasons for congestions is that the road capacity is not optimally used. This leads to the development of Routing systems which use dynamic information about the current state of the road network. By avoiding congested roads and using alternative ways instead the network capacity is exploited in a more optimal way. Cars that use such a Routing system are routed around the congestions. And if many cars use alternative roads the congestions will be reduced. Such Routing systems route the cars along the path with shortest time and not along the path with shortest distance like nearly all today's navigation systems do. Conventional navigation systems are nowadays widely used and are offered by many companies. But they are not able to avoid routing a car into congestions because they do not use dynamic data. So if there are congestions on the road network, these navigation systems can only be considered as help for finding the way to a destination but not to find the shortest way in time.

## 2 Distributed routing

In [5] a dynamic vehicle Routing system was introduced which uses a promising approach, namely the Ant Based Control algorithm (ABC-algorithm). The Routing system is able to guide individual car drivers through a city. The algorithm is very suitable for decentralized, dynamic route optimization in networks where

the dynamic data changes very fast. So the Routing system is expected to have a great potential compared with conventional route planners.

As it was found out in [5] the size of the network is a limiting factor for the ABC-algorithm. The algorithm does not perform as well for big networks as for small networks. For developing a dynamic Routing system for a large area with several cities and motorways between the cities it is not possible just to use a larger traffic network for the existing Routing system. It is necessary to split up the map in local/global sectors. In a Hierarchical routing system vehicles are routed using a hierarchical network. The approach is that a separate network exists for every city. And the motorways that connect the cities are part of another network. In that way the whole network is split up into several smaller networks which can be handled by different distributed computer systems (Figure 1).

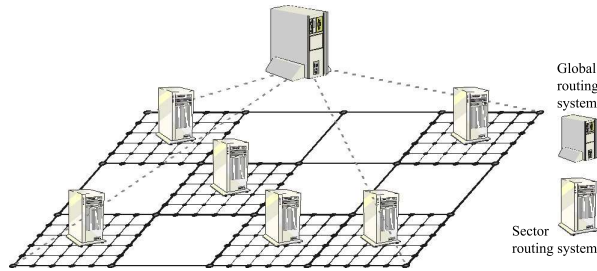


Figure 1: Distribution of the Hierarchical routing system

The Global routing system and the Sector routing systems each can run on a separate computer system. These computer systems must be mutually connected via a network to cooperate as a common system. The vehicles use a wireless connection to connect with the Hierarchical routing system. Compared with a Routing system that runs on one computer system the distribution has several advantages. Because now several computer systems fulfil the task that one computer system did, the speed and the available memory space increases. Besides, the failure of one of the computer systems does not have to imply a total breakdown of the whole Routing system. The drawback of the distribution is that always some communication between the systems is necessary.

### 3 Network model

The kind of network that is considered for the development of a Hierarchical routing system is a simplified traffic network which can be considered as a model of a geographical map of cities connected by highways. An example of such a network is shown in Figure 2.

The sector is connected to the motorways by special nodes (exit points). Inside each sector there can be a further kind of network, a city network, also consisting of nodes and links. These links represent streets in the city and the nodes represent intersections, with or without traffic lights. The motorway intersections are the

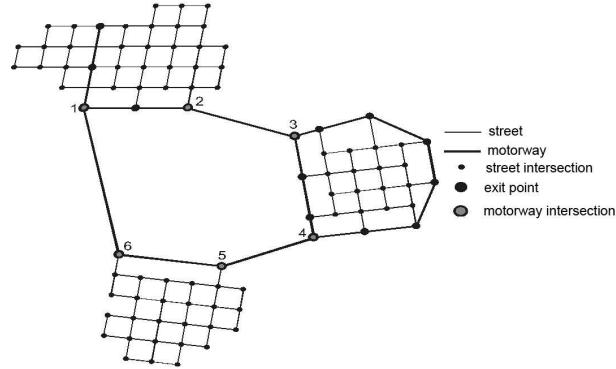


Figure 2: Traffic network with three cities

nodes on the motorways that form the edges of a sector. The traffic network forms a directed graph.

In this network model a hierarchy can be introduced by splitting up the network into several less complex networks. The resulting hierarchical network consists of two detail levels, the abstract level and the detailed level. The lower detailed level is the abstract level. On the abstract level there is only one network, the motorway network. To it belong all motorways and motorway intersections (Figure 3). The detailed level consists of several independent city networks, one for each sector that contains a city network. To a city network belong all streets and intersections of the sector and also the surrounding motorways with their intersections (Figure 4). By splitting up the network in this way a hierarchical traffic network based on the human's behaviour to plan a route is created.

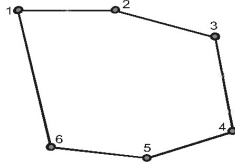


Figure 3: Global network

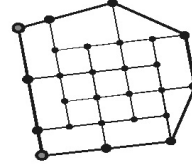


Figure 4: A sector network

## 4 Design

The Hierarchical routing system consists of several parts. These are the Global routing system and a Sector routing system for each sector that contains a city network. So if there exist  $N$  sectors with a city network then the Hierarchical routing system will consist of  $N$  Sector route finding systems. Every part of the Hierarchical routing system consists of the same components as shown in Figure 5, namely a Timetable updating system (TUS) and a Route finding system (RFS).

Besides all Routing systems have their own timetable which stores the dynamic data of the traffic network.

## 4.1 Dynamic data

To route the traffic dynamically through the streets network, we need dynamic data about the state of the traffic on the roads. As source for information about the traffic situation, are the vehicles themselves. Vehicles can provide information about the path they followed and the time they needed to cover it. To locate the current position vehicles can use the GPS-technology (Global Positioning System). At regular intervals vehicles send the position where they sent the last route information, the current position and the followed path between both positions via a wireless connection to a server. With that information the time estimates to cover each road can be computed on the server. In that way the travel times for every link of the network can be determined from the information provided by the cars.

Vehicles connect directly with the the Routing system responsible for the part of network where these are (Figure 6). The following rules define with which part of the Routing system vehicles must connect:

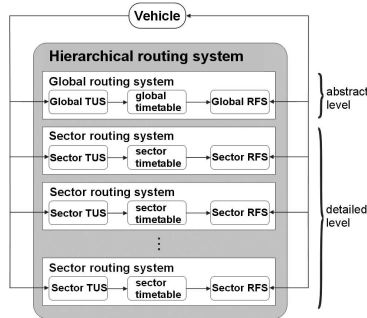


Figure 5: Design

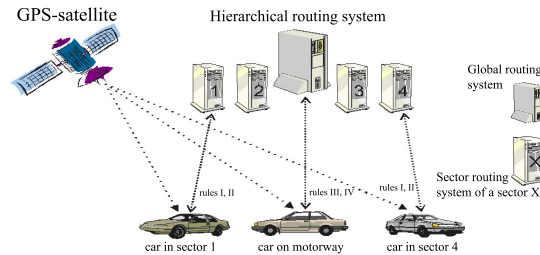


Figure 6: Vehicle connection

1. Request a route inside a sector: connect with the Sector RFS.
2. Send update information inside a sector: connect with the Sector TUS.
3. Request a route on a motorway: connect with the Global RFS.
4. Send update information on a motorway: connect with the Global TUS.

## 5 Timetable updating system

The reason why we need the TUS is the following: the RFS needs information about the state of the network (dynamic data). Those data are provided by the TUS. That information can be for example the load of the parts of the network but a more direct and therefore more practical type of information is the time

it takes to cover a road. Vehicles send information about their covered route to the TUS. From that information this system computes the travelling-times for all roads and stores it in the timetable in the memory  $M$ . Every link  $l$  in the network graph is weighted with a value  $M_l$ . This value represents the average estimated time needed to cover the link, from the source node to the destination node.

Now let's consider how the computation of the time for every single link of the covered route can be done: First of all, the total time to cover the route without delay at intersections or on the links needs to be calculated. The formula for this is:

$$T = \sum \frac{d_l}{S_l} \quad (1)$$

With the optimal time calculated with formula (1) the travel time for every single link can be computed by using the following formula:

$$x_l = \frac{L_l}{S_l \cdot T} \cdot t \quad (2)$$

$$M_l = M_l + a \cdot (x_l - M_l) \quad (3)$$

1.  $d_l$  is the covered distance on link  $l$  in meters
2.  $L_l$  is the length of link  $l$  in meter
3.  $S_l$  is the allowed speed on link  $l$  in meter per second
4.  $T$  is the total time in seconds that is needed to cover the route without delay
5.  $t$  is the total time of the covered route in seconds.
6.  $x_l$  is the actual measurement of the travel time for link  $l$  in seconds
7.  $M_l$  is the mean of the travel time measurements for the link  $l$ .

The computed travel times computed by the formula (2) are only average values. The delay of the congestion on one of the links is distributed over the computed travel times of all links of the covered route although only one street is congested. This can be avoided if the vehicles send update information at every intersection. But a drawback is the amount of data and communication required between the vehicles and the TUS. Therefore a compromise should be made for a suitable update interval to avoid too much communication on the one hand, and to ensure accurate and actual travel times on the other hand.

## 6 Route finding system

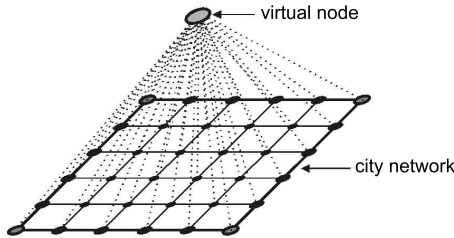
This Route finding system uses a variant of the AntNet algorithm described in [3]. Routing is determined through complex interactions of network exploration agents. These agents (ants) are divided into two classes, the forward ants and

the backward ants. The idea behind this sub-division of agents is to allow the backward ants to utilize the useful information gathered by the forward ants on their trip from source to destination. Based on this principle, no node routing updates are performed by the forward ants, whose only purpose in life is to report network delay conditions to the backward ants. This information appears in the form of trip times between each network node. The backward ants inherit this raw data and use it to update the routing tables of the nodes.

Each RTS has a model of the traffic network of his sector and differs from the packet switch network for which AntNet was designed. No packets are running here. There are no buffers in nodes, and an infinite bandwidth is available on the links. The links still have a virtual delay provided from the time tables. The delay represents the necessary time for a car to cross the link.

## 6.1 Virtual nodes

For every sector that contains a city network a virtual node is introduced. A virtual node is the union of all nodes that belong to the same sector. So it can be understood as abstraction from all the nodes of the sector (see Figure 7). Each virtual node will have an entry in the data structures of every node at both levels of the hierarchy (abstract and detailed). Considering a map with  $m$  sectors, in Figure 8 is an example of a routing table in a one of the  $n$  nodes of a sector. The abstract level contains  $m$  nodes. Whenever a car is routed from an intersection of one sector to an intersection of a distant sector, this destination sector is represented by its virtual node.



| Next node →<br>Destination ↓ | n1   | n2   |
|------------------------------|------|------|
| d1                           | 0.68 | 0.32 |
| d2                           | 0.71 | 0.29 |
| ...                          |      |      |
| dn                           | 0.98 | 0.02 |
| V1                           | 0.34 | 0.66 |
| V2                           | 0.81 | 0.19 |
| ...                          |      |      |
| Vm                           | 0.54 | 0.46 |

Figure 7: Virtual node of a sector

Figure 8: Sector node probability table

## 6.2 The Ant Based Control algorithm

We adapted AntNet for our network model. The algorithm for a sector is different then the one for the global system.

In a sector, as showed before, in the routing table of a node we have entries for each destination in the sector, but also for each virtual node. The algorithm works as follows:

1. The mobile agents  $F_{s \rightarrow d}$  are launched at regular time intervals from every sector node  $s$  and with random destination (another node or a virtual one).

2. Each ant keeps a memory stack  $S_{s \rightarrow d}(i)$  with the visited nodes. When an ant arrives in a node  $i$ , coming from node  $j$ , it memorizes the identifier of the visited node  $i$  and the virtual delay of the link ( the trip time necessary for a car to travel from intersection  $j$  to intersection  $i$ ).
3. When ant comes in the node  $i$ , it has to select a next node  $n$  to move to. The selection is done according with the probabilities  $P_d$ .
4. When the destination node  $d$  is reached, the agent  $F_{s \rightarrow d}$  generates another agent (backward ant)  $B_{d \rightarrow s}$ , transfers to it all of its memory, and dies. If the destination is a virtual node  $v$ , the ant stops when it reaches one of the motorway intersection nodes  $d$ . Here it pushes onto the stack also the virtual node  $v$  identifier and  $\mu_v$ , the average time to go from  $d$  to the sector  $v$ .
5. The backward ant takes the same path as that of its corresponding forward ant, but in the opposite direction. At each node  $i$  along the path it pops its stack  $S_{s \rightarrow d}(i)$  to know the next hop node. It also updates the probability table with a reinforcement value  $r$ . This is a function of the time  $T_{i \rightarrow d}$  the ant computed and the local stochastic model of traffic in the node.

$$P'_{dj} = P_{dj} + r(1 - P_{dj}) \quad \text{when } j = n \text{ the next node chosen by the ant} \quad (4)$$

$$P'_{nd} = P_{nd} - rP_{nd}, \text{ for } j \neq n \quad (5)$$

6. When the source node  $s$  is reached again, the agent  $B_{d \rightarrow s}$  dies.

The motorway intersection nodes are present on both networks, local and global. The global system copies from the node at the abstract level, the values  $\mu_v$  for the virtual nodes  $v$ , and send them to the sector copy of the node. To avoid extra communication, this is done just in case a value  $\mu_v$  changes with more than 5% in the node at the abstract level.

For the Global routing system the algorithm is slightly different. The routing tables of the nodes have entries only for the virtual nodes. Most of the steps are similar with the ones in the first case:

1. The mobile agents  $F_{s \rightarrow v}$  are launched at regular time intervals from every network node  $s$  and with random virtual node as destination.
2. The virtual destination  $v$  is reached when the ant arrives in a node  $d$ , which is also part of the sector  $v$ .
3. In every node  $i$ , the backward ant  $B_{d \rightarrow s}$  updates the routing table just for the virtual node  $v$ .

The cars start on the links at the sector level. They have as destination another node which can be in the same sector or in a distant one. The car routing has the following steps:

1. If the destination node is in the same sector then the car is routed by the Sector routing system to the node following the maximum values in the probability tables.

2. If the destination is in another sector, the car is routed by the local system according with the virtual node entries in the probability tables. This is done until the car arrives at one of the motorways intersections which belong to the actual sector.
3. Once the car reaches a motorway intersection, it is routed by the Global Routing system. The routing is done following the probabilities entries for the corresponding virtual node of the destination sector.
4. When the car arrives at a node which belongs to the destination sector, the routing is switched to the destination's Sector routing system. This one will guide the car to the final destination node in the sector.

## 7 Conclusions

We designed and developed a first prototype of a Hierarchical routing system as modelled in Figure 2. The simulator allowed us to test the performance of the algorithm in different situations. It showed a significant improvement compared with a non Hierarchical routing system, where the whole network was considered as a sector. The concept of Hierarchical routing combined with the AntNet-algorithm ensures an excellent scalability of the Routing system. The system is capable of routing vehicles in very complex networks. Besides, the design of the Hierarchical routing system ensures a high robustness against failures of parts of the distributed system. His disadvantage is that, in some cases, the cars are sometimes routed along a sub-optimal path.

## References

- [1] G. Di Caro, M. Dorigo. AntNet: A Mobile Agents Approach to Adaptive Routing, Technical Report 97-12, IRIDIA, Universit Libre de Bruxelles (1997)
- [2] R. Schoonderwoerd, O. Holland, J. Bruten, L.J.M. Rothkrantz. Load Balancing in Telecommunication Networks, Adaptive Behaviour, vol. 5 no. 2, (1997)
- [3] G. Di Caro and M. Dorigo. AntNet: distributed stigmergetic control for communication networks. Journal of Artificial Intelligence Research (JAIR), 9, (1998), 317–365
- [4] L.J.M.Rothkrantz, J.C. Wojdel, A. Wojdel, H. Knibbe. Ant based routing algorithms, Neural Network World, Vol. 10, No.3, July (2000)
- [5] R. Kroon, L.J.M Rothkrantz. Dynamic vehicle routing using an ABC-algorithm, Transportation and Telecommunication in the 3rd Millenium, Prague, May 26–27, (2003)