Intelligent dynamic route planning

G. Eggenkamp L.J.M. Rothkrantz

Delft University of Technology, Zuidplantsoen 4, 2628BZ Delft, the Netherlands

Abstract

In this paper the possibilities of artificial intelligence and especially of expert systems in the field of route planning using dynamic traffic data are explored. An expert system that has been built to perform dynamic routing and a dynamic route planner using a (traditional) shortest path algorithm are introduced. Using both implementations a comparison is made between the expert system approach and the shortest path approach. It is concluded that the expert system shows great potential. It outperforms the shortest path algorithm in computation time and the routes the expert system finds are indeed the shortest routes.

1. Introduction

Currently no dynamic route planners are available. Although the highway network in the Netherlands is flooded with cars and is subject to heavy congestion in both rush hours no planner is available that finds the shortest route in this (congested) network. The only option some route planners and car navigation systems offer is to ‘block’ roads that are congested and to find the best alternative route without using this road. Consequently a route is advised that might well take longer then the route along the congested road, since this road is not even considered anymore. Since dynamic data are available from the MONICA monitoring system (the detection loops under the highways) a study has been carried out to develop such a dynamic route planner.

When research was carried out to the performance of shortest path algorithms, like Dijkstra’s algorithm, to find the shortest route while using dynamic data, it showed that the computation time degrades significantly when dynamic data are incorporated. Consequently, other possibilities were investigated and since humans are quite well capable of finding alternative routes in the case of congestion it was decided to study the feasibility of an artificial intelligence approach. In this paper the feasibility of an expert system in a dynamic route planner is discussed and a comparison is made with a ‘regular’ shortest path algorithm.

This paper will start with a problem definition (section 2) and an introduction of a shortest path algorithm that can be used to find the shortest path in a dynamic network (section 3). In section 4 an introduction of the expert system that was constructed is given, while in section 5 the results of both approaches are presented. In section 6 some conclusions are given.

2. Problem definition

The highway network can be represented by a graph, as with static routing problems. The highway network that was considered in the research is the network that is being monitored by the MONICA system, since only dynamic data are available of these roads. In Figure 1 this network is shown. Somehow the dynamic aspect of the data has to be taken into account. The travel times between different edges (cities, junctions) change in time, and these changes have to be taken into account and incorporated in the graph. When, for example, travelling from Amsterdam to Delft in the morning rush hour, a departure of only 5 minutes later, can affect the travel time by more then 20 minutes, since major congestion may have occurred along the route during these 5 minutes. For example an accident might have happened or a sudden peak in cars that want to access the highway may have occurred.

A space time extended network (STEN) explicitly represents time by having a complete layer of all nodes of the physical network per time period. The first occurrence of STEN in literature can be found in [4]. Other applications of space-time expanded networks can be found in [1,3,6]. In all these publications time expanded networks were used to solve traffic assignment models, which are dynamic flow problems. In dynamic route planning only the shortest path has to be found, no dynamic flow problem has to be solved. Consequently, the same approach can be used, only with a flow of 1 for all links.

Concordant to these publication the space-time expanded network can be constructed as follows:

- for each period p create a complete layer of all nodes of the physical network,
- for each node in period p (all nodes with the same t), create links to the nodes it is connected with in the physical network in the corresponding period 'layer' p + d, with d the travel time when starting at period p,
- for each node in period p create a link to the same node in period p + 1 (it is also possible to stop in a node).

An example of a network constructed this way is shown in Figure 2. The original graph consisting of nodes A to G is repeated for each time interval. The edges between the nodes A to G (the lowest graph at t = 10:01) represent which nodes are connected to each other. For clarity these edges have been kept in the different layers of the graph to show these connections. The thicker links that intersect the different layers are the actual road connections. Their length (and thus the layer to which they go) represents the travel time when starting at the time of the layer in which they start. For each layer for all nodes all outgoing links are constructed and labelled according to the travel time at that moment. It should be noticed that in Figure 2 not all the links are shown, since that would have resulted in a cluttered figure.

3. The extended Dijkstra algorithm

The most secure way to find an optimal route from an origin to a destination at a specific time of the day would be to find the optimal route in the graph that was constructed in the previous section. In [4] a proof is given that a dynamic routing problem that is expanded in the way described can be solved using static shortest path algorithms.

In practice the graph that is proposed in section 2 better can not be constructed, since this would require a lot of computation time. It would be far more efficient if travel times only were estimated if they are really needed. Consequently an algorithm was constructed that finds the shortest path in this 3-dimensional graph and which only estimates travel times if necessary. When the algorithm was constructed and was reviewed thoroughly, it was discovered it differs with Dijkstra's algorithm only slightly. Consequently, it was called the 'extended Dijkstra algorithm'. Since the Dijkstra algorithm is widely known, no explanation is given here. It can be found in [3].

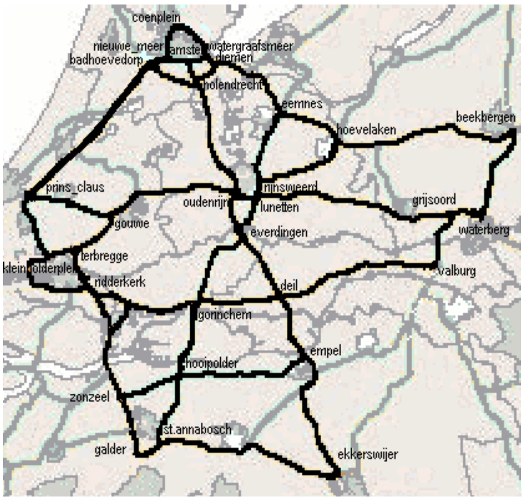

Figure 1. The freeway network that is monitored by the MONICA system.

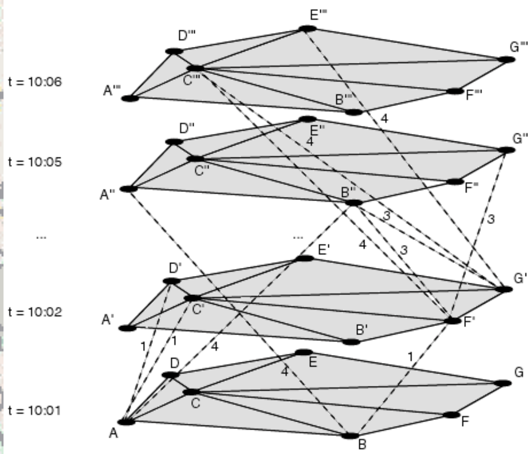

Figure 2. A space-time expanded network.

It should be noticed that no research was done into the estimation of travel times, which is a very complex process. During this project we focused on one main aspect: route planning. Consequently, it was assumed travel time estimates were available and a set of historical data was used as 'dummy' data.

4. Expert system

In this section the expert system that has been constructed is introduced. As was stated in section 2 the problem domain of the expert system is given by the road network that is given in Figure 1. The expert system should find the fastest route in this network.

4.1 Knowledge elicitation

The knowledge the expert system should possess consists of alternative routes in the case of congestion along a part of a road. This knowledge can be made explicit in two ways. Firstly, experts can be interviewed. These experts should be experienced 'traffic jam travelers' that often have tried alternative routes in the case of congestion along a part of the freeway they normally use. Secondly, the map of the Netherlands combined with historical traffic data can be investigated, to see which alternatives are reasonable in the case of a traffic jam. In this project, the second approach was chosen. The reason to do this was as follows. Travelers are not able to monitor the routes they did not choose. When they have chosen an alternative, afterwards they do not know if it was faster then the original route or other alternatives (unless they know another traveler, who tried the alternative at the same time). Consequently, the perception the traveler has of the quality of alternatives he tried can be wrong, since it may also be influenced by other incentives then the shortest travel time.

4.2 Level of detail

For the level of detail in which congestion in the road network is monitored route sections between junctions where one can change freeways were chosen. Since routes are only optimised in the freeway network (and not considering secondary roads), only at junctions the route can be changed. Since secondary roads are not taken into account, it is not interesting to construct rules on the basis of congestion between two ramps: for all ramps that are between two junctions, the same rules would be constructed, since only at the first junction after the ramp it is possible to change the route. In Figure 1 the different road parts between junctions can be found (the junctions are identified by their names).

4.3 Route representation

The expert system is provided with a number of route parts (trajectories), that each have a predicate, which can be ‘route’, ‘file’, ‘entrance’ or ‘exit’. These route parts are delimited by two junctions. An example of such a route can be found in Figure 4. The route on the map was generated by a static route planner and was translated to a route for the expert system. Right of the figure the translation of the route can be found.

(entrance coenplein nieuwe_meer)
(route nieuwe_meer badhoevedorp)
(route badhoevedorp burgerveen)
(route burgerveen prins_claus)
(route prins_claus ypenburg)
(exit ypenburg kleinpolderplein)

Figure 4. The route from Amsterdam to Delft and the format that is given to the expert system.

It was chosen to keep the choice whether there is congestion along a trajectory, outside the domain of the expert system. A separate module was constructed in which it is decided if the delay is significant enough to consider the trajectory congested. This module provides the expert system the route with its appropriate predicates.

4.4 Construction of the rule base

For each trajectory along the road network rules were made stating which alternative to take if the trajectory was congested. Since there are 92 edges in the network that is monitored by the MONICA system (Figure 2) the construction of these rules was a time consuming task. The alternative route that can be taken when a route part is congested depends on the direction one is coming from and the direction one is going to: the rules for alternative roads depend on the previous and following route parts. The different steps, which are needed to construct the different rules for each trajectory are given in the following action list.

1. Determine the possible directions where one can be coming from
2. Determine the possible directions where one can be going to
3. Determine the different alternative routes for each possible route, by investigating the map and historical data.
4. Calculate the 'detour time' of each alternative: the time needed to make the detour in the best case (no congestion along the alternative route).
5. Order the different routes according to this 'detour-time'.
6. When the 'detour-time' is too large, do not use the alternative.
7. Check with reports of car drivers if no routes are missing

The first two steps are very straightforward, the possible directions can be found by having a look at the map. The third step requires by far the most time: in this step the different alternative routes have to be chosen. When these routes are known, their travel times can be computed. The fifth step is important to find the best route as quickly as possible. This property can be very useful if the dynamic route planner is used in a real-time environment and there is a time constraint. When alternatives become available very fast, while searching is continued for better alternatives, the best route found so far can be used if the time constraint has to be met. Of course, it would be optimal if the first route found also is the best route and this is examined using this parameter. Consequently, the order in which the alternatives are searched should depend on the travel times and the chance of congestion along these alternatives, when there is congestion along the trajectory for which alternatives are searched.

5. Results

To be able to compare both methods the following four parameters were chosen: 1) the number of travel time estimates, 2) overall computation time, 3) shortest route found and 4) order of found routes.

The number of travel time estimates parameter was chosen since it gives an indication of the performance of the algorithm. Since the travel time estimation is the process that needs the most computational time the number of travel time estimates will strongly indicate the total computational time needed.

The overall computation time parameter is included to be able to judge the performance of the expert system. It could be possible, the expert system approach needs only few travel time estimates, but is very slow itself, since the rule base is very large.

The third variable on which the methods will be compared is the shortest route that is found. Since it is mathematically proven that the extended Dijkstra algorithm will return the shortest route this parameter is only applicable to the expert system.

The last variable which will be carefully examined is the order in which alternative routes are given. This aspect is also only applicable to the expert system, since Dijkstra's algorithm does not return any other route then the best one. It should be examined how many alternatives have to be computed to find the shortest one. As was stated in section 4.4 this can be important in a real-time environment.

5.1 Testing protocol

To test both approaches several departure and destination addresses were chosen between which the shortest route had to be found. Firstly the routes were searched on a free-flow network, without congestion. Congestion was created along one of the trajectories in the shortest route that was found and both algorithms were applied again. Again congestion was created along one of the trajectories of the newly found route and both algorithm were applied. This process was repeated until all trajectories were delayed. In Figures 5 and 6 the first two iterations of this process are illustrated. In Figure 5 the free flow route that was found is shown. Congestion was created between the Prins Claus and Badhoevedorp junctions and both algorithm were applied. The shortest route found now is shown in Figure 6. Now congestion was created between the Holendrecht and Diemen junctions and the same process was repeated. In Table 2 an overview of the results of the first three steps of this testing procedure for the route between Zoetermeer and Muiden is given.

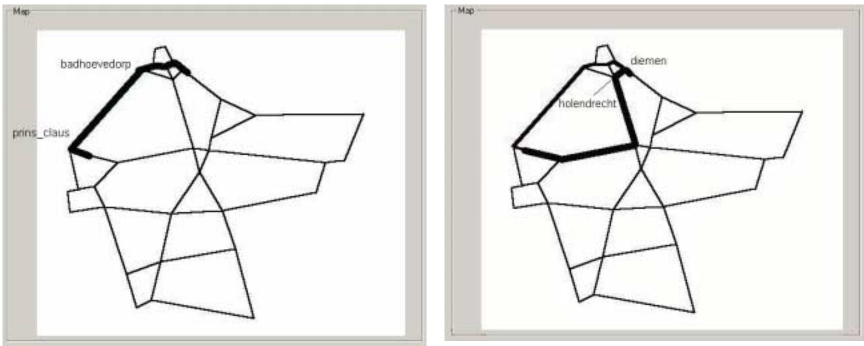

Figure 5 and 6. Free flow route and route if there is congestion between the Prins Claus and Badhoevedorp junctions.

Table 2. Test results of route between Zoetermeer and Muiden.

	exp. syst. trav. est.	exp. syst. comp.time	graph alg. trav. est.	graph alg. comp.time	order found	routes are the same
Original route, A12-A4-A10-A1 (48’)	0	9500	722	17470	0/0	Yes
Delay between prins claus and badhoevedorp (A4), +20’,A12-A2-A9-A1 (53’)	93	10000	1286	34270	1/3	Yes
Delay between holendrecht and diemen (A9), +15’, A12-A2-A10-A1 (54’)	94	10490	1285	34880	3/3	Yes

5.2 Results

In Table 3 the results of this testing procedure for different routes are shown. In the most left column the routes are denoted together with the number of iterations with congested trajectories that was carried out. Columns 2 to 5 show the average number of travel time estimations and the average computation time of both methods. In column 6 some kind of indication is given which alternative of the expert system was the best route. In Table 2 this was indicated as x/6 for each route, which means that 6 alternatives were generated and the first one of these was the fastest one. The indication that is given in Table 3 is simply the sum of the total routes found (right number) and the sum of the numbers that indicate when the route was found (left number). A value of 4/13 indicates that overall measured the fourth alternative was the right one, given thirteen routes. The last column indicates the number of correct routes out of the number of total routes found.

Table 3. Average values of testing procedures.

	expert syst. travtime est.	expert syst. comp. time	graph alg. travtime est.	graph alg. comp.time	order found	same routes
Muiden-Amerongen (6)	96	9403	1268	33958	4/13	5/6*
Amerongen-Delft (7)	142	10511	1570	46656	8/15	7/7
Amsterdam-Apeldoorn (7)	203	8457	1436	36744	13/42	7/7
Deventer-Gouda (8)	130	10011	1015	32034	12/29	7/8*
Weesp-Moordrecht (10)	466	14664	1302	33238	21/65	9/10*
Total average	**229**	**10899**	**1310**	**36216**		

* The routes were different, although the travel time was the same.

Table 3 shows the expert system requires significantly less computation time then the shortest path algorithm. The overall computation time is a factor 3.5 less, while the difference of the number of travel time estimates is almost a factor 6. Since it is expected the estimation of the travel time will take (by far) the most computation time in a real-time estimation it can be expected the expert system will perform even better when used together with MONICA data: the data the detection loops generate have to be combined with historical data using some kind of prediction algorithm, which will take a lot more computation time then currently was needed, since dummy data were used.

In Table 3 it can be seen that the few times the routes were different (three times out of approx. 60 routes), the travel times were the same. Since different algorithms were used to find the shortest route both approaches returned a different one, although the other alternatives were also returned. As a result it can be stated that the quality of the routes found by the expert system is very good. On the other hand it should also be remarked that the testing procedure influenced the results a little bit. During the testing procedure no scenario's were tried to frustrate the expert system. It would be possible to create such congestion along all reasonable alternatives that a very strange alternative would become the best one. When for example travelling from Amsterdam to Utrecht one could create severe congestion along all 'normal' alternative roads such that one would have to travel via Apeldoorn and Arnhem, back to Utrecht to have the fastest route. Of course such situations are very rare in reality.

With respect to the order in which the expert system generates alternative routes it can be remarked the results are quite well. Most of the time one of the first routes that is generated actually is the fastest route. On the other hand it can be noticed that sometimes the best route is one of the last routes found. This is a consequence of the unpredictable behaviour of congestion. As was stated in section 4.4 it was tried to rank the different alternative routes in such a way that the alternative with the highest chance of being the

best one was tried first. Since a chance guarantees nothing sometimes other routes are better. Especially when more then one trajectory is congested along a route the best alternative can be one of the last ones tried. Two or more trajectories are congested, so two or more file predicates will instantiate different rules. The order in which these rules fire cannot be regulated in a way the rule with the 'best' alternative fires first, since more then one trajectory is congested and it can not be stated beforehand which alternative will be most promising in that case. Consequently the alternative that are fired by one rule might all be tried first after which the second rule fires which contains the best alternative.
The last remark that can be made is concerned with the implementation. In section 4 it was stated the travel times of each alternative should be computed to prevent the travel times of alternatives being computed that will not make a chance since their detour time is larger then the delay due to the congestion. Since the construction of the rule base took much more time then expected this implementation was not made. Subsequently sometimes alternatives were tried that should not be tried at all.

6. Conclusion

In this paper the possibilities of an expert system in the field of dynamic route planning were discussed and a comparison was made between a shortest path algorithm and the expert system. The expert system showed great potential. Not only performs the expert system much better with respect to computation time, the routes the expert system returns are as good as the routes the conventional shortest path algorithm computes and the expert system shows great possibilities when real time constraints are placed. The expert first generates all possible solutions and then computes their travel time one by one. As soon as the travel time of a solution has been computed the solution becomes available.

The most important drawback of the expert system approach is the construction of the rules. This is a very intensive process and requires a lot of time.

References

[1] H.K. Chen. *Dynamic travel choice models: a variational inequality approach*. Springer, Heidelberg, 1999.

[2] G. Eggenkamp. *KRIS: Knowledge based routing information system*. Graduation thesis, TU Delft, 2001.

[3] J.R. Evans and E. Minieka. *Optimization algorithms for networks and graphs.* Marcel Dekker Inc., New York, 2nd edition, 1991.

[4] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, New Yersey, 1962.

[5] B. Ran and D.E. Boyce. *Modeling dynamic transportation networks: an intelligent transportation system oriented approach.* Springer-Verlag, Berlin, 2nd edition, 1996.