

# Practical Time-Dependent and Stochastic Routing with Historical Measurements of Travel Times

S. Demeyer, P. Audenaert, S. Logghe, M. Pickavet and P. Demeester

**Abstract**— We present a case study of an industrial-strength time-dependent and stochastic routing system, which makes use of predicted travel times for road networks. Next to a short theoretical consideration, the major focus of this research was on the practical implementation of the system, which aims at efficiently routing in a road network making use of stochastic travel time information. Using historical measurements collected by cell phone and GPS tracking techniques, we derive time-dependent travel time probability distributions for all links in our network. Making use of these distributions, one can accurately determine the travel time distributions for whole routes. In this way, the chance of arriving in time at the destination can be optimized. Moreover, a number of suggestions, not yet implemented due to various causes, for extending and improving the platform are presented. The results of this work are deployed by the industrial partners involved in this research.

## I. INTRODUCTION

This article presents a practical time-dependent and stochastic routing system which was developed for the IBBT ICON project MobiRoute [1]. The goal of this project is to optimize routing by taking into account historical, actual and predicted travel time data.

A time-dependent and stochastic routing algorithm has been developed, starting from well-known shortest path algorithms such as the algorithm of Dijkstra [2], which makes use of a single link cost (for example the financial cost, the travel time, the ecological cost, etc.). In reality, link costs, more specifically travel time, are dependent upon the time of the day. Traversing an important highway link in the rush hour (with a lot of traffic jams) will consume significantly more time than traversing that same link at night. One option to deal with this is replacing the single link costs with travel time functions. Algorithms, which make use of this time-dependent information, are called time-dependent routing algorithms. An overview of most commonly used time-dependent shortest path algorithms is given in [3]. Most of these algorithms originate from a modified Dijkstra algorithm [4], in which labels are updated according to the departure time in the origin. In order to calculate the time-dependent shortest routes for a whole day, a label-correcting algorithm [5] has been developed that

avoids iteratively applying this time-dependent path algorithm for every time of day.

While routing time-dependently provides more accurate results, it does not take into account that, especially in road networks, the travel times of the different links are influenced by random factors (individual driver's behavior, weather conditions, accidents). By making use of stochastic distributions, one can solve this uncertainty problem. Instead of a two-dimensional cost function, links now have assigned to them a three-dimensional function, which contains a travel time distribution for every time of day.

The stochastic shortest path problem first started to attract attention in the 70's, where a number of theoretical solutions have been proposed ([6] and [7]). Hall [8] was the first to address the time-dependent stochastic shortest path routing problem in a more practical way by proposing an algorithm which calculates the routes with the least expected travel time. He demonstrates that routing stochastically brings on a new spectrum of problems. Fu and Rilett [9] further investigated the problem of stochastic routing and proposed an expected travel time approximation making use of Taylor series. In [10], a promising stochastic shortest path problem is presented. Unfortunately the complexity of this algorithm is exponential, which makes it unusable for large transportation networks.

In this paper we present a case study of a routing system, which makes use of historical data, to predict the travel times. The routes are calculated combining time-dependent and stochastic aspects, which means that the user, providing an origin, destination and departure time, receives the best route at this time together with a distribution of the expected travel time. In the next section the context of this research is described, together with an overview of the available resources. Section 3 gives a global overview of the routing system, while a more detailed discussion can be found in section 4. In the following section, some experimental results are presented to demonstrate the operation and some characteristics of the system. Section 6 then focuses on some optimization measures for the future system. We end this article with a conclusion.

## II. CONTEXT AND RESOURCES

In order to better understand this paper, and appreciate the approach taken to tackle the problems, it is necessary to describe in some detail the context of our work. Since a few years, a major industrial player conquered the commercial field of traffic information by providing up-to-date and real-time information on the current status of traffic flow on the road network in Belgium. Next to that, they also provide a limited service of traffic forecasting. An interesting way to

Research supported by IBBT ICON project MobiRoute  
S. Demeyer is with Ghent University, Gaston Crommenlaan 8 bus 201, B9050 Ghent (Corresponding author: [sofie.demeyer@intec.ugent.be](mailto:sofie.demeyer@intec.ugent.be))

S. Logghe is with Be-Mobile, Technologiepark 12b, B9052 Ghent (e-mail: [steven.logghe@be-mobile.be](mailto:steven.logghe@be-mobile.be))

All other authors are with Ghent University, Gaston Crommenlaan 8 bus 201, B9050 Ghent (e-mail: [first.last@intec.ugent.be](mailto:first.last@intec.ugent.be))

achieve this has proven to be very successful in measuring the actual throughput on any given moment of the day: they have a business agreement with one of the major cellular phone providers in the country, from which they receive real-time handover timings of cellular phones. This means that all phone subscribers are continuously tracked (anonymously), allowing to trace their routes and to measure the time it took to drive each road segment. This is done by complex calculations involving triangulation and the mapping of the signals on roadmaps. In practice, there are lots of phone users driving, which do provide timings on how long it takes to drive a certain segment of our road network. This allows not only for knowledge about current traffic jams, but collecting all this data provides an enormous amount of historical travel time data, which enables accurate predictions for future jams.

The company contacted our research group, in order to develop a route planner which does take into account the historical data, and provides accurate estimates for the travel times together with uncertainty intervals. Indeed, it is perceived that a large number of people using route planners are not interested in the shortest route from A to B, but the fastest route, together with an uncertainty interval which indicates the chances of delay. The company provided us with a road network consisting of 53010 nodes and 96286 directed links between these nodes. Together with this, a database of timing measurements was provided in the form of data from 10 Tuesdays throughout the year. For every 15 minutes an average of the measurements was calculated. Together this made up for  $96286 * 10 * 4 * 24$  data-items, each representing the travel time measured in seconds. Taking everything together, this made up about 2GB of unprocessed data files. Sometimes measurements were missing (due to external measurement system failures) and in such cases the values were interpolated.

The aim was to develop an industrial-strength (i.e. realistic and robust) commercial platform for probabilistic routing. To that aim, we had to make sure that the data structures and algorithms are both robust and scalable. Where necessary, a trade-off was made between theoretical issues and practical feasibility. We developed the system using a Java platform running on a stock PC with an Intel Dual Core 2.4GHz and 3.5GB RAM.

### III. GLOBAL SYSTEM OVERVIEW

In the literature, mathematical solutions to the problem at hand have been proposed. Often, these are technically entirely correct, but suffer from the fact that they are not really practical: typically the computation times or memory constraints do not allow an industrial-strength deployment. Therefore, in this paper, we develop a theory which trades off mathematical rigor to practical usability.

Figure 1 gives an overview of the system. Three major operations can be distinguished: data processing, graph modeling and stochastic routing. In this section we will shortly address each of them separately, while in the next section a more detailed discussion will be provided.

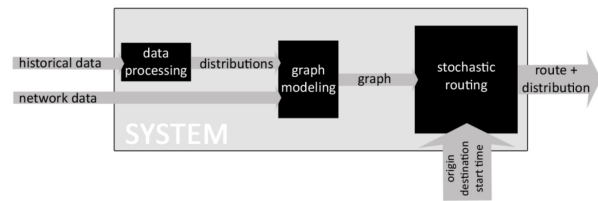


Figure 1 System Overview

#### A. Data Processing

In the data processing step, raw historical data is transformed into distributions. A distribution needs to be determined for every link for every predefined timeslot of the day. It should be noted that in order to accurately predict these data points not only the measurement data of the current (predicted) timeslot should be taken into account but also the data of the neighboring timeslots. Nevertheless, this data should have a lower impact on the prediction than the data of the predicted timeslot. In this way, the predicted distribution of the travel time for timeslot  $p$  can be modeled as

$$v(p) = \frac{\iint_{-\infty}^{\infty} G^p(x, y) D(x, y) dx dy}{\iint_{-\infty}^{\infty} G^p(x, y) dx dy}$$

where  $G^p(x, y)$  represents a weight function which states that the values of the specific timeslot are more important than the values of the neighboring timeslots. The function  $D(x, y)$  is a density function which represents the measured data points. The denominator normalizes the predicted distribution. An example of a good weight function is depicted in Figure 2, where the original data points are shown in the horizontal plane. To predict the distribution of a different timeslot, the weight function  $G^p(x, y)$  is shifted parallel with the time axis. It should be noted that a distribution needs to be predicted for every link and for every timeslot. That means that if  $V$  is the number of links and  $T$  is the number of timeslots,  $V * T$  distributions need to be calculated.

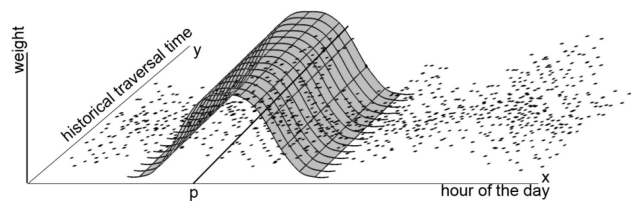


Figure 2 Example Weight Function

#### B. Graph Modeling

Following most common graph models, the graph contains a set of nodes, representing geographic locations, together with a set of links connecting them. As the routing, presented in this paper, is both time-dependent and stochastic, special attention should be paid to link cost modeling. In our graph model every link has assigned to it a number of distributions, namely one for every timeslot. The

amount of disk space needed to store the link costs is both dependent upon the number of timeslots, i.e. the granularity, and the size needed to store one distribution. It should be noted that the storage space needed for the complete network is consumed almost entirely by these link costs.

### C. Stochastic Routing

This is the main operation of the system, which calculates the stochastic shortest route for a given origin, destination and departure time. The starting point of this module is the algorithm of Dijkstra [2], i.e. the most common shortest path algorithm. This is a label setting algorithm, in which in each iteration the temporary node with the best label is made permanent and the labels of its neighbors are updated. In order to deal with stochastic travel times, two aspects of this algorithm need to be redesigned: calculating new labels and determining the best label out of a set.

Calculating a new label means combining a label (stochastic distribution) with a travel time distribution. We will distinguish two extreme cases: all links are correlated or all links are completely uncorrelated. Links are correlated if the traffic on one link has an influence on the traffic on the other link. This is for example the case on a highway where a traffic jam on one part (i.e. link) has an impact on the traffic of the neighboring parts. Nevertheless, in a realistic transportation network some of the links are correlated, while others are not, and the exact value lies somewhere between the two extremes presented here. Determining the exact correlations between all links would have a major impact on the calculation time. For correlated distributions the pointwise sum should be used, while for combining completely uncorrelated distributions, the convolution product can be applied. It should be noted that a convolution product consumes remarkably more calculation time than a pointwise sum. For a more detailed discussion of the convolution product and the pointwise sum, we would like to refer to the following section.

To determine the best distribution from a set, a comparison measure needs to be defined. For simplicity reasons, we opted to compare the 50% percentiles. In this way, the median values of the distributions are compared. If, at the other hand, the user wants routes with more certainty to arrive in time, the 90% percentiles can be used for comparison. This will be discussed in more detail in the following section.

## IV. DETAILED FRAMEWORK: OPTIMIZED DATA STRUCTURES AND ALGORITHMS

In the previous section a general overview of the system was given. In this section we will elaborate further on the concepts mentioned there, enabling a practical implementation of the system. Translating the theoretical system to a practical efficient implementation here means implementing discrete versions of the traditional techniques of calculus.

We make use of a specific data structure to model the stochastic (cumulative) distributions and then adapt a label setting algorithm in order to work with this stochastic information.

### A. Data Processing

The question that should be answered here is how to transform a large amount of data into a number of discrete distributions. First of all, a discrete distribution model needs to be defined, which is a tradeoff between the accuracy of the information and the storage space needed for it. Since we are working with actual measured data and we do not want to impose one standard stochastic distribution (defined by a number of parameters), we will define a distribution by a number of actual values. The more data points that are stored for a distribution, the more storage space is consumed, but limiting the storage size (in the extreme case to one single value) has a negative effect on the accuracy of the information. A distribution in our system is represented by  $n$  values, namely  $n$  predefined percentiles of the accumulated distribution. Experiments have shown that a distribution with 5 values is a good tradeoff between storage needs and accuracy, namely the 10%, the 30%, the 50%, the 70% and the 90% percentiles. The  $x\%$  percentile having a certain value means that there is  $x\%$  chance that traversing the link will take no longer than the given value. To determine these percentiles, all values need to be ordered, resulting in the  $x\%$  percentile on the  $\frac{n \cdot x}{100}$ -th place, with  $n$  the number of data points. If this number  $\frac{n \cdot x}{100}$  is not an integer, a weighted average of the neighboring values is used.

Instead of ordering the complete list of values and then determining the according percentiles, order statistics [11] can be used, which speed up calculations, as the data only needs to be ordered partially.

### B. Graph Modeling

As common practice, our graph model consists of a number of nodes together with a number of directed links, connecting these nodes. These links have cost objects assigned to them. An overview of the data structures is given in Figure 3.

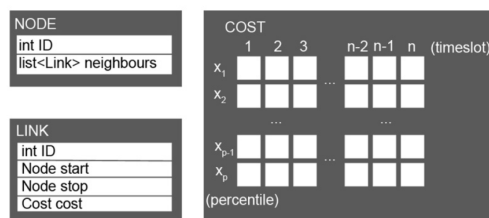


Figure 3 Data Structures of the Graph Model

A cost object contains a distribution for every time of the day. To translate the time to the discrete domain, we opted for dividing the day in predefined number of timeslots. According to the historical data at hand, timeslots of 15 minutes were used, leading to days of 96 timeslots. As mentioned before, distributions consist of 5 values which means that  $5 \cdot 96 = 480$  values need to be stored in a cost object of a single link. It should be noted that, concerning storage capabilities, most space is consumed by the cost objects, as can also be seen in Figure 3.

The network data at hand is data which is used in a number of commercial route planners. It contains a realistic

version of the Belgian road network. As driving time data can only be measured on the most important, i.e. high level, roads, the first mile data, i.e. the data of the small streets that lead to the houses in residential areas, was omitted. Adding this data to the system would have no real impact, except for small storage issues, as these links have single travel time values since no measurements and thus no distributions are available. The first mile route then can be calculated using the standard shortest path algorithm of Dijkstra.

### C. Stochastic Routing

As mentioned before, a time-dependent stochastic shortest path algorithm was developed, based on the algorithm of Dijkstra [2]. The main difference lies in the way the labels are calculated. The original algorithm of Dijkstra demands a graph with single cost values for every link. A tentative label of node  $b$  then is calculated as follows:

$$L(b) = L(a) + C((a, b))$$

where  $L(a)$  represents the label of the previous node  $a$  and  $C((a, b))$  is the cost of link  $(a, b)$ . When routing time-dependently, this cost is dependent upon the starting time  $t_s$  in the origin node, which leads to

$$L(b) = L(a) + C((a, b), (t_s + L(a)))$$

where  $C((a, b), T)$  represent the cost of link  $(a, b)$  at time  $T$ . It should be noted that we are working with travel times, which means that  $t_s + L(a)$  represents the time at which link  $(a, b)$  is traversed.

When working with stochastic distributions the sum operator cannot be applied. As stated before, two extreme cases are investigated: completely correlated links (pointwise sum) and completely uncorrelated links (convolution product). The pointwise sum can be defined as follows. Let us denote the stochastic distribution of link  $(a, b)$  at time  $T$  with  $P((a, b), T)$ , and the corresponding percentiles with  $P_{x\%}((a, b), T)$ . Furthermore, the  $x\%$  percentile of the distribution of label  $L(a)$  is represented by  $L_{x\%}(a)$ . The distribution of the tentative new label of node  $b$  then can be represented by the percentiles

$$L_{x\%}(b) = L_{x\%}(a) + P_{x\%}((a, b), (t_s + L_{x\%}(a)))$$

with all parameters as previously defined. It can be seen that calculating the pointwise sum of two distributions, as defined in this system, results in calculating no more than 5 sums, namely one for each percentile.

In contrast with the pointwise sum, the convolution product, which is used when all links are uncorrelated, requires all possible combinations to be taken into account. The convolution product of two distributions  $f$  and  $g$  is defined as

$$f * g = \int_{-\infty}^{\infty} f(x)g(t - x)dx$$

The associative characteristic of the convolution product enables it to be used in routing algorithms. We translated the original definition to a calculable version in the discrete space. In order to calculate the tentative new label of node  $b$ , the following sum needs to be calculated for all possible  $x$  and  $y$ :

$$S_{x,y}(b) = L_{x\%}(a) + P_{y\%}((a, b), t_s + L_{x\%}(a))$$

Subsequently, the set of all elements  $S_{x,y}(b)$  is ordered and the  $n$  percentiles of the distribution are collected. As shown in the data processing part of this section, these percentiles can be found faster by making use of order statistics. Nevertheless, in this case, a highly efficient mergesort algorithm can be applied to this set, as it can easily be divided into a number of ordered subsets. Calculating the sums  $S_{x,y}(b)$  for every value of  $x$  results in subsets which are ordered according to  $y$ . By merging different ordered sets recursively, a remarkable calculation time gain can be realized for ordering the complete set. For small numbers of percentiles using the Fast Fourier Transformation does not pay off.

## V. RESULTS

We deployed the stochastic and time-dependent routing system, according to the specifications stated in the previous section, as a web service of which a stripped down version is publicly available at [14]. This system uses the measured historical data to construct a number of distributions for each link. Subsequently, a network is built with time-dependent and stochastic link costs. The 'best' route then can be calculated on demand for every origin, destination and departure time. In this section, the operation and characteristics of this system will be clarified, making use of a number of results.

The system routes time-dependently, which means that the result is dependent upon the time of the day. We calculated a route between Ghent and Liège at different departure times, namely at 6:00, 7:00, 8:00, 9:00 and 10:00. Figure 4 shows a part of these routes, namely the part around Brussels with a high chance of traffic jams at rush hour. We supposed the links to be completely uncorrelated and made use of the convolution product to combine distributions. It can be seen that before (6:00) and after (10:00) the rush hour the standard route is taken, namely the highway from Ghent over Brussels to Liège. In the rush hour the highway links with the traffic jams will be avoided by taking secondary roads, in order to arrive earlier at the destination. Leaving at 7:00, the highway before Brussels is jammed when arriving there, which means that taking secondary roads from Aalst to Brussels is a better option. Nevertheless, once arrived in Brussels, the highway around Brussels can be taken to continue the trip. This is not possible when leaving at 8:00. When leaving at 9:00 the highway before Brussels is no longer congested, while the highway around Brussels still is. This results in the route as can be seen in the figure.

Moreover, route calculations on 1000 randomly chosen origin-destination pairs with a random departure time have shown that by routing time-dependently an average travel time gain of around 15 % can be realized.

As shown in the previous section distributions can be combined in two different ways, namely by making use of the convolution product or by applying the pointwise sum. To calculate a path between an origin and a destination at a certain starting hour, it takes approximately 30 milliseconds (ms) when the convolution product is used, and 20 ms with

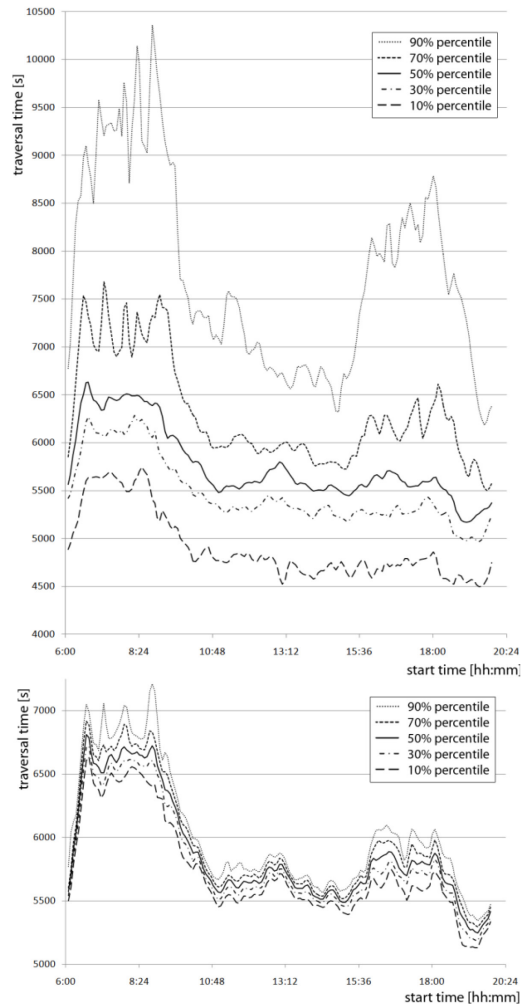


the pointwise sum, as expected since the pointwise sum needs fewer calculations to come to a solution. These results are the average calculation times of 1000 paths between a random origin and a random destination at a random starting hour. Paths in the network at hand averagely contain 100 hops.



**Figure 4 Time-dependent Route Calculation**

As can be seen in Figure 1, routes are returned together with a stochastic distribution of their travel times. In order to get an overview of the resulting distributions during a whole day, the shortest route was calculated between a fixed origin and destination, namely Ghent and Liège, at different departure times. The result is shown in Figure 5, with above the pointwise sum (correlated links) and below the convolution product (uncorrelated links). The most important observation from this figure is the fact that the resulting pointwise sum distributions are more diverse than the convolution product distributions, i.e. the difference between the 90% percentile and the 10% percentile values is much larger. This can be explained intuitively as the convolution product takes all values of the distribution into account, resulting in extreme values being weakened by the others. When applying the pointwise sum, the extreme values remain and are even amplified since extreme percentiles usually have extreme values and vice versa. Moreover, in the case that the links are correlated, a delay on one link can have a large impact on the travel times of the succeeding links, while this is not the case when the links are uncorrelated. Nevertheless, the 50% percentiles show similar values for both the correlated and the uncorrelated links. This is obvious as in both cases these are combinations of the average values. Furthermore, the rush hours can be observed in this figure. Traversal times in the rush hour are much higher, which indicates that there is a major rush hour in the morning around 8am and a smaller one in the evening around 5pm.



**Figure 5 Day Overview**  
(top: pointwise sum - bottom: convolution product)

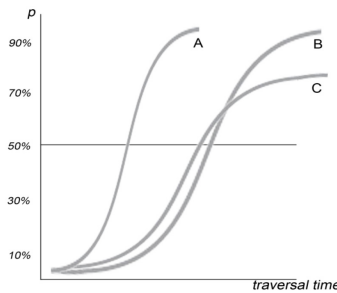
## VI. THE FUTURE SYSTEM

In this article, a basic routing system was presented, which will be further extended in the future. First, we intend to make the system multimodal, i.e. making use of two or more modes of transportation. Subsequently, two improvements to the operation of the temporary set of the algorithm will be presented. Lastly, we will go into the comparison of the stochastic distributions.

For one thing, a larger (European) network will be used and real-time updates of the travel time data will be provided. Moreover, other modes of transportation will be added, making the network multimodal. As these modes are mostly public transportation modes bound to timetables, the travel time data is time-dependent but not stochastic. In order to make routing possible in the multimodal network, we assume the distribution of these travel times to be constant. These extensions are not yet implemented, as we currently are gathering the appropriate data. In this section a number of optimizations are presented, which can speed up the calculations of this future routing system. We will focus on three aspects of the system: the data processing step, the convolution product of two distributions and the comparison between the different distributions.

Secondly, the temporary set (now implemented as a priority queue) used in the shortest path algorithm can be optimized. Since a lookup operation in a priority queue is quite expensive, by not removing duplicate nodes when updating the temporary set, a significant speedup can be realized. Moreover, the temporary set can be implemented as a Fibonacci heap ([11] and [12]), which enables the temporary set updating operation to run in  $O(1)$  amortized time.

In our system, to compare two distributions with each other, the 50% percentile is compared, in order to avoid the extensive calculations of determining the surface between the distribution's curves. This is a good measure in most cases, but is not always accurate. This is illustrated in Figure 6, in which 3 cumulated travel time distributions are depicted.



**Figure 6 Comparison between (accumulated) distributions**

Comparing distribution A with distribution B and A with C results in distribution A to be the best, i.e. the distribution with the lowest traversal times. This can be verified both intuitively and according to the definition. Comparing distributions B and C with each other is much harder. Intuitively we would indicate distribution B as the best, as the traversal times are almost equal to those of C for the lower percentiles and, for the higher percentiles, it has much smaller traversal times than distribution C. Nevertheless, according to the definition, distribution C is perceived as the best, i.e. it has the lowest value for the 50% percentile. To avoid these situations, another comparison measure is proposed, which leads to a partial ordering relationship. Instead of only comparing a single value of the distributions, all values can be compared. This results in a Pareto optimal set in which no distribution dominates any other. A distribution dominates another one if and only if all of its values are better than the corresponding values of the other distribution. Instead of the algorithm of Dijkstra, now a multiple objective shortest path algorithm [13] should be applied.

## VII. CONCLUSIONS

In this article, a case study of a novel time-dependent and stochastic routing system was presented. Making use of historical data, which is extracted from cell phone signals on the road network, probability distributions are calculated for every link at certain times of the day. Subsequently, a time-dependent and stochastic graph is built out of realistic network data and these distributions. To the best of our knowledge, the combination of these data structures and an

especially adapted algorithm has never been used before. In this graph, routes can be calculated time-dependently and stochastically for each origin, destination and departure time. Two combining options are provided for the distributions, namely one in which is supposed that all links are uncorrelated (convolution product) and one in which is supposed that they are completely correlated (pointwise sum). A stripped down version of this routing system can be tested out at [14].

It is shown experimentally that this system indeed calculates the routes dynamically and avoids the predicted traffic jams. Moreover, these routes can be calculated in approximately 30 ms and 20 ms for the convolution product and the pointwise sum case respectively, for paths of averagely 100 hops. Furthermore, the pointwise sum causes the resulting distribution to be more diverse, i.e. to have a larger difference between the 90% and the 10% percentile, in comparison with the convolution product.

The routing system described in this paper is an industrial-strength commercial system, which will be further improved in the future. In the previous section a number of optimizations were presented, which may speed up the calculations. Furthermore, the system will be extended to cope with more data sets (day-of-week) and multiple transportation modes. At this moment, only road traffic is taken into account. We are currently adding railroad traffic and other public transportation means. The system, as presented in this paper, is implemented [14] and used for commercial purposes.

## REFERENCES

- [1] IBBT (2009-2011). MobiRoute. <http://www.ibbt.be/en/projects/overview-projects/p/detail/mobiroute-2>, accessed March 2012
- [2] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, 1, pp. 269-271, 1959.
- [3] D. Dellinger and D. Wagner, "Time-Dependent Route Planning," *Robust and Online Large-Scale Optimization*, LNCS 5868, pp. 207-230, 2009.
- [4] K. Cooke and E. Halsey, "The Shortest Route through a Network with Time-Dependent Intermodal Transit Times," *Journal of Mathematical Analysis and Applications*, 14, pp. 493-498, 1966.
- [5] B.C. Dean, "Continuous-Time Dynamic Shortest Path Algorithms," Master's thesis, Massachusetts Institute of Technology, 1999.
- [6] H. Frank, "Shortest Paths in Probabilistic Graphs," *Operations Research*, 17, pp. 583-599, 1969.
- [7] E.H. Sigal, A.A.B. Pritsker and J.J. Solberg, "The Stochastic Shortest Route Problem," *Operations Research*, 28, pp. 1122-1129, 1980.
- [8] R.W. Hall, "The Fastest Path through a Network with Random Time-Dependent Travel Times," *Transportation Science*, 20, pp. 182-188, 1986.
- [9] L. Fu and L.R. Rilett, "Expected Shortest Path in Dynamic and Stochastic Traffic Networks," *Transportation Research Part B*, 32, pp. 499-516, 1998.
- [10] A. Azaron and F. Kianfar, "Dynamic shortest path in stochastic dynamic networks: Ship routing problem," *European Journal of Operational Research*, 144, pp. 138-156, 2003.
- [11] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. *Introduction to Algorithms* - 2<sup>nd</sup> ed. MIT Press, Cambridge, 2001.
- [12] M.L. Fredman and R.E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM*, 34, pp. 569-615, 1987.
- [13] E.Q.V. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, 16, pp. 236-245, 1984.
- [14] S. Demeyer, "Stochastic Routing." <http://dna.intec.ugent.be/StochasticRouting>, accessed July 2012