Reliable Dynamic In-vehicle Navigation

Ioannis Kaparias (MEng)

A thesis submitted for the degree of Doctor of Philosophy of the University of London

and Diploma of the Membership of Imperial College London

Centre for Transport Studies

Department of Civil and Environmental Engineering

Imperial College London, London, UK

August 2008

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Prof Michael G. H. Bell, for his essential and continuous guidance, support and encouragement throughout this PhD. Whilst giving me the freedom to develop my own ideas, he has always provided me with constructive feedback and critical reviews, aiming to improve my work as much as possible. Without his outstanding supervision this thesis would not have been possible at all.

Furthermore, I would like to thank BMW Group for sponsoring this work, and in particular the efforts and support of Dr Klaus Bogenberger and Dr Heidrun Belzner. Additionally, I would like to thank PTV AG and particularly Karsten McFarland for providing the test network used in the field experiments of this project, as well as ITIS Holdings Ltd and Ian Forshaw for supplying historical floating vehicle data for the field experiments.

Many thanks go to Prof Yanyan Chen from Beijing University of Technology, who particularly assisted me in the early stages of my PhD, and to Prof Hartmut Keller from TRANSVER Ltd and the other partners of the OFFENSIVE project for their valuable peer review throughout the duration of the project.

Further, I would like to thank my colleagues, office-mates and good friends Kyoung-A Park, Konstantinos Zavitsas and Panagiotis Angeloudis, the former two for helping me with the conduct of the field experiments, and the latter for always being around when I needed assistance with programming matters. Many additional thanks go to them, along with Susie, Ning, Jan-Dirk and Simon, for being such great office-mates and for always keeping a pleasant and happy atmosphere in room 205. A big thank you also goes to my uncle Kimon for taking the time to proof-read my thesis and to provide me with extremely useful comments.

Last but certainly not least, special thanks go to my parents and to all my friends around the world for always being there for me, for believing in me and for offering me their support and valuable advice throughout my study.

Declaration of contribution

At various stages during the PhD research reported here, collaboration has taken place with colleagues working on the subject. The output of this collaboration has been included in this thesis to better explain and support the research reported. In particular, the research carried out has built upon the work conducted for the OFFENSIVE project, sponsored by BMW Group, which started in January 2004 and ended in June 2007. I became a member of the project team in October 2004 and contributed to the development of the initial formulation of the new in-vehicle navigation algorithm, which is described in Chen et al (2005a) and Chen et al (2005b). This work was carried out in collaboration with Prof Yanyan Chen from Beijing University of Technology during her stay at Imperial College London in 2004, Prof Michael G. H. Bell and Dr Klaus Bogenberger from BMW Group, and is reported and appropriately referenced in Section 4.3.1.

In addition to that, collaboration with my supervisor, Prof Michael G. H. Bell, has taken place in the work described in Section 2.4.4, which involved the mathematical proof of the optimality of the reverse A* algorithm.

I hereby declare that apart from the collaborations referred to above the work described in this thesis has been carried out by myself.

Kaparias
………………………………………………………………
(Ioannis Kaparias)

Michael G. H. Bell
………………………………………………………………
(Prof Michael G. H. Bell)

Abstract

Having started off from luxury makes and models, in-vehicle navigation systems are now gradually spreading through the entire vehicle fleet, as drivers appreciate their usefulness. Increasingly sophisticated systems are being developed, having much more advanced functions than simple driving directions. This thesis presents a new approach for in-vehicle navigation, in which travel time reliability is incorporated in the route finding component of the navigation system. Based on historical traffic data and in the absence of current traffic information, positions in the road network at which it is likely to encounter delays, are predicted and avoided as much as possible by the route finding algorithm.

The thesis starts by reviewing shortest path algorithms and conjectures that the most appropriate algorithm to use is A*, which forms a vital part of the approach developed. Performing multiple runs of A* forwards and backwards on the road network, efficiency of the route finding procedure is achieved. The time-dependent version of the algorithm is also derived. Then, the thesis goes on to define reliability on a single link of the road network as the maximum delay that can be encountered with 90% confidence and extends this definition to derive the reliability of entire routes.

Having introduced the route finding procedure and the concept of reliability, the thesis presents the in-vehicle navigation approach, which involves computing a more reliable route from the driver's origin to his/her destination than the fastest, if this is unreliable. Additionally, the approach aims at computing multiple alternative partially disjoint but equivalently reliable routes to the driver, such that the congestion feedback effect can be avoided as much as possible, without the need of carrying out a dynamic traffic assignment, which would be impracticable in an in-vehicle system. A number of constraints are introduced so as to ensure that the resulting routes are acceptable to the driver (are not too long, etc). Hence, the main concept lies in initially computing the fastest time-dependent route, then applying penalties to the links characterised as unreliable (increasing the link weights in inverse proportion to their reliability) and re-running the route finding algorithm so as to find a more reliable route. After each run, the route obtained is checked against the constraints and if it does not satisfy them, it is discarded, the penalties are reduced and a new route is sought. In order to obtain alternative par-

tially disjoint routes, penalties are also applied to links that are already included in a previously computed and accepted route. The new algorithm, RDIN, is thus presented and mathematically formulated. An extension to RDIN for re-routing, RDIN-R, is also developed.

The software tool developed for the application of RDIN and RDIN-R, the Adaptive Reliable Imperial Advanced Navigation Engine (ARIAdNE) is described. A simulation example is given for demonstration and preliminary validation; then a number of field experiments are carried out in Central London to test the method in a real road network environment and to compare its performance with an existing conventional car navigation system. The results suggest that the method is workable and precise, while at the same time it is a promising step forward in the field of in-vehicle navigation.

Table of contents

List of figures

List of tables

CHAPTER 1

Introduction

1.1 Background

Intelligent Transport Systems (ITS) can be viewed as the integration and deployment of advanced technologies in areas such as information, communication and navigation in order to achieve a more efficient and safer transportation system (Huang et al, 1995). Especially in the last decade, the field of ITS has experienced a spectacular growth with emerging technologies being increasingly implemented in transport applications. Examples range from the development of new advanced information systems for both private and public transport, to the introduction of cutting-edge systems in road, rail and air transport aiming to reduce or even prevent accidents, and to the utilisation of new technologies for the protection of the environment.

As an integral part of ITS, in-vehicle navigation systems were initially treated as luxury goods and were only offered as premium accessories on high-priced cars. Nonetheless, this has changed and navigation systems have now become everyday consumer goods, as they are widely recognised, together with the traditional radio, as the most important source of information in the vehicle, and as their prices have dropped due to technological advances in mobile devices and applications. Sales reached 12 million units in 2006 in Europe and North America, and forecasts predict that they will reach 53 million units by 2012 (Berg Insight, 2007a), proving thus that in-vehicle navigation is a rapidly evolving field.

This study presents a novel in-vehicle navigation methodology with the prospect of being implemented in a new advanced navigation system, which will consider the reliability of travel time and will thus provide the user with accurate driving directions and travel time predictions. The aims and objectives of the study are set out in Section 1.3; the current section continues with the background of in-vehicle navigation, which includes a description of the benefits of such systems and a brief historical overview of the topic, while Section 1.2 presents the state-of-the-art of the field.

1.1.1 Benefits of in-vehicle navigation systems

In-vehicle navigation systems have benefits for both their users and the whole transport system. Starting from the users, who in this case are the drivers of the vehicles equipped with navigation devices, the primary purpose and benefit of in-vehicle navigation is assistance in unfamiliar areas. In areas of the road network, of which a driver has little or no knowledge, it is very likely that, in the absence of a navigation system, he/she will have difficulty in finding a route to his/her destination and will consequently have to stop and either look at a map or ask for directions. Apart from the loss of time that such a process requires, it also has safety implications, as it is possible that the driver will not find an appropriate place to stop and will end up looking at the map or even speak on his/her mobile phone whilst driving. Making use of an in-vehicle navigation system considerably eases the driver in this kind of situation.

A further benefit for the driver is assistance in adverse traffic conditions, i.e. in congested areas. A navigation system having access to appropriate traffic information (e.g. real-time congestion and accident reports) enables the driver to seek for alternative routes to his/her destination so as to avoid congested roads and hence benefit from important travel time savings. Additionally, by not driving on congested roads the driver also benefits from significant savings in terms of fuel consumption and vehicle wear and tear. The latter may not be crucial in the short-term; nonetheless it is an important cost to be considered in the long-term, as vehicles constantly subjected to adverse traffic conditions tend to deteriorate faster.

Considering the benefits of in-vehicle navigation for the transport system, the most important one is the fact that the use of such devices reduces the overall time lost in the network and eases congestion (Jeffery, 1981). Vehicles driving around, unsuccessfully looking for the right route constitute a significant proportion of the traffic flow of a network; the use of navigation systems results in trips being shorter, thus relieving the network from a considerable traffic load. Additionally, making trips shorter and decreasing the total driving time in the network results in less pollution, so navigation systems contribute to the protection of the environment. Also, by improving the safety of individual drivers, navigation systems also increase the overall safety in the network, as they result in less dangerous driving thus reducing the probability of occurrence of traffic accidents.

Naturally, navigation systems have also been criticised as having the opposite effects to their intended ones under certain conditions, particularly with respect to safety and congestion. Regarding safety, such systems are accused of in fact causing accidents instead of preventing them, as they often require input from the driver whilst en route and thus distract him/her from driving. With respect to congestion, on the other hand, and despite the fact that navigation systems generally reduce traffic volumes, they are blamed for guiding a large number of vehicles along specific routes, thus increasing traffic levels at some locations. Nevertheless, these are issues that are gradually being taken into account by manufacturers, such that their occurrence is eliminated in the development of newer systems.

1.1.2 Brief historical overview of in-vehicle navigation

Vehicle navigation has been in the spotlight of humanity for thousands of years, with the earliest vehicle navigation system dating as back as ancient China in 2600 B.C., when the so-called “south-pointing” carriage was invented. This was a two-wheeled cart on which a human figure was mounted, whose characteristic was that it was always pointing to the South, no matter which way the cart was moving. The south-pointing human figure was maintained in position through the action of a gear train (Zhao, 1997).

More sophisticated devices were developed later as a result of the advances in science, based on the invention of new navigation equipment, such as the magnetic compass and the odometer. However, it was not until enough technological advances in the field of wireless telecommunications were made that the first in-vehicle navigation system in its present form appeared.

The ERGS (Electronic Route Guidance System) was a system developed in the 1960s by the US Federal Highway Association, aiming to provide route guidance to vehicles (Rosen et al, 1970). Its components consisted of in-vehicle units, enabling the interaction of the system with the driver, and roadside proximity beacons with roadside controllers placed at intersections, used for two-way communication with the in-vehicle units. The concept of the system's operation was as follows: the driver entered the destination of his/her trip which was transmitted to the roadside beacon and controller each time the vehicle approached an intersection. The roadside controller on the other side computed the direction to the destination and transmitted, through the beacon, the requested driving manoeuvre to the in-vehicle unit. The controller was also connected to a centralised host for accessing traffic data.

In the 1970s a similar system was developed in Japan. The CACS (Comprehensive Automobile Traffic Control System) project, started in 1973 and completed in 1979, was presented as "a communication system that links vehicles in motion, roadside equipment and a central data processing core" (Totani, 1980). The system offered three functions, namely dynamic route guidance, emergency radio and driving information. Not only was this a significant improvement to ERGS in terms of the route guidance function (it was based on real-time data derived from measured travel times collected from vehicles equipped with the system), but also two entirely new functions were offered. The result of a field test that was carried out showed that guided vehicles benefited from significant travel time savings; nonetheless, a problem that was discovered was that when the vehicle's speed was higher than 100 km/h there was not enough time for data exchange between the roadside unit and the in-vehicle unit.

Two other systems using a similar concept to ERGS and CACS were developed in the 1980s in Europe, namely Autoguide (Catling and Belcher, 1989; Turner and Hoffman, 1990) in the UK and ALI-SCOUT (a.k.a. LISB or EURO-SCOUT) (Hoffmann and Janko, 1990; von Tomkewitsch, 1991) in Germany. The Autoguide system involved positioning through roadside beacons, enabling two-way communication with the in-vehicle unit. The route computation was based on real-time traffic data, gathered from other vehicles and managed by a control centre, and was updated each time the vehicle went past a beacon. While making use of similar components, the main feature of ALI-SCOUT was that it involved central route searching carried out for any given origin-destination pair at constant time intervals. Upon arrival at a junction with a roadside beacon, the vehicle transmitted real-time travel time information to the beacon and received back information about the quickest routes to all surrounding beacons. The travel time data measured and used in the calculation of the routes was also complemented by inductive loop data.

The introduction of satellite positioning in the 1990s gave a new dimension to in-vehicle navigation, as systems did not have to rely entirely on ground infrastructure anymore. Examples of experimental systems using the Global Positioning System (GPS) in order to identify their location include ADVANCE and TravTek in the US, and Travelguide in Canada. Starting from ADVANCE (Boyce et al, 1991; Boyce et al, 1993; Saricks et al, 1997), this was experimentally launched in 1991 by a consortium of public sector, academic and industrial partners in the area of Chicago, Illinois. The system consisted of in-vehicle units, which used GPS, dead reckoning (i.e. extrapolating the current position based upon a previously determined position, known speed, elapsed time and course) and map-matching (i.e. matching a set of geographical coordinates on a road map) techniques for establishing their locations, while map data was stored in an on-board CD-ROM. Ground infrastructure on the other hand was limited to the existence of the Traffic Information Centre (TIC), which was in continuous contact with the in-vehicle units through a two-way data exchange procedure. Vehicle units transmitted travel time measurements to the TIC through a dedicated radio frequency, and received traffic information from it, which they used in the route computations they offered to the drivers. While a large-scale field test with 5000 test vehicles was planned, only a small-scale deployment took place with 80 test vehicles.

Along the same concept and almost concurrently with ADVANCE, TravTek (Rillings and Lewis, 1991) was developed by a similar consortium of partners in the early 1990s in the city of Orlando, Florida. The functions offered by TravTek were navigation, real-time traffic information, route guidance (exact driving directions by voice) and information services. The aim of the TravTek test was to evaluate the user perception of in-vehicle navigation systems and its conduct involved equipping 100 rental vehicles with the system and surveying the drivers upon return of the vehicles. The result was that the drivers were overall positive towards this new system and made fewer driving errors compared to when using conventional maps (Ni and Deakin, 2002). In a similar concurrent project in Ontario, Canada, Travelguide (Heti, 1993) was developed. Its main difference was the fact that it was embedded in a portable device aimed at being used not only as an in-vehicle navigation system, but also as a general low-cost travel assistant. Hence, it also offered real-time information about public transport.

The advances in the satellite positioning field in the 1990s resulted in affordable GPS receivers becoming available, which gave particular boost to the development of so-called ‘autonomous’ in-vehicle navigation systems (as opposed to ‘centralised’ systems described above), i.e. systems not depending on the communication and data exchange with a traffic management/information centre and having all the information they require pre-loaded and stored on board the vehicle unit. Before satellite positioning became available, very few autonomous systems had been developed. Examples include ARCS (Automatic Route Control System) (French and Lang, 1973), which was the first basic autonomous navigation system developed and was using dead reckoning for positioning whilst driving on pre-determined routes, and CARIN (Car Information and Navigation) in the 1980s, which also used dead reckoning for positioning but was innovative because it was the first system to use an on-board CD-ROM for map storage (Zhao, 1997). Particularly in the last decade more autonomous navigation systems have appeared in the market; the development of new centralised systems on the other hand has decreased and the functions previously offered by such services (e.g. real-time traffic information) are now usually offered as extras on basic autonomous systems. The state-of-the-art of the in-vehicle navigation field is described in the next section.

1.2 Modern in-vehicle navigation systems

Starting from the forms in which modern navigation systems are available, this section goes on to describe the state-of-the-art of car navigation and the individual modules of which a modern navigation device consists. Then, following market research published in the literature, the evaluation of current systems by drivers is presented and their needs are identified.

1.2.1 State-of-the-art

In a report by Wood et al (2006), whose aim is to develop a guide for the evaluation of car navigation systems, the state-of-the-art of the field is presented. After identifying car navigation systems as a rapidly growing market with increasing demand from car users, it is pointed out that the market is divided, such that in-vehicle navigation is available to the user either from original equipment manufacturers (OEM) or from after-market suppliers. The former mainly include systems embedded in new cars by the manufacturers, usually as optional equipment, while the latter refer to after-market devices purchased and installed in the vehicle from independent suppliers.

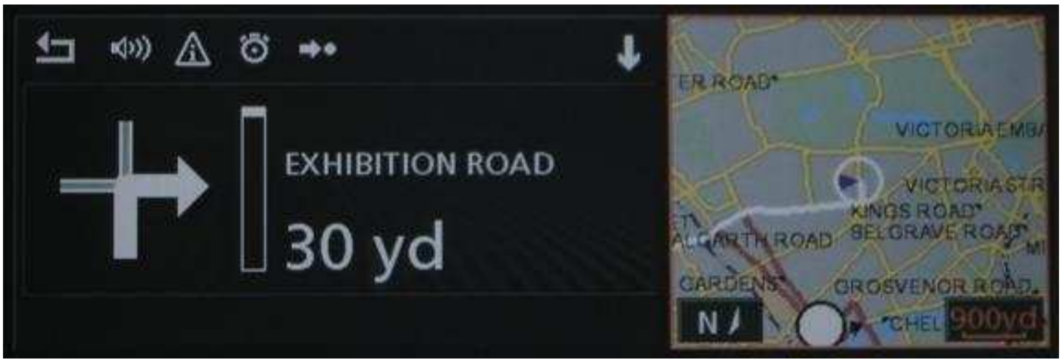


Figure 1.2-1: Embedded in-vehicle navigation unit (from BMW™)

Figure 1.2-2: Free-standing after-market in-vehicle navigation unit (from Garmin™)

There are seven different types of in-vehicle navigation devices, according to Wood et al (2006):

- Dedicated units integrated into the vehicle by the OEM (see Figure 1.2-1)
- In-vehicle entertainment devices with navigation installed as an accessory or replacement of a conventional function (e.g. radio, CD, tape)
- Dedicated after-market devices permanently installed in the vehicle
- Dedicated free standing after-market devices that can be temporarily installed in the vehicle (see Figure 1.2-2)

- Personal digital assistants (PDAs) with suitable software and connection to a GPS receiver
- Mobile phones
- Mobile data terminals

Also, systems are classified as either on-board or off-board (i.e. autonomous or centralised); in the former, the map data, routing software and GPS receiver are all in the vehicle, while in the latter, the vehicle is only equipped with a GPS receiver and the remaining entities are located on a central server. In a different classification by Berg Insight (2005), in-vehicle navigation consists of three segments: stand-alone navigation, PDA navigation and smartphone (i.e. advanced mobile phone) navigation. Especially the last two are increasingly gaining in popularity, as the current trend is towards the integration of navigation systems with other units (Wood et al, 2006). As an example, sales of navigation software for smartphones reached 0.6 million in Europe and North America in 2006, and it is estimated that by 2012 there will be 43 million mobile phone navigation users in those regions (Berg Insight, 2007b).

From the variety of available navigation systems and devices able to support personal navigation software, it can be seen that the in-vehicle navigation market is very competitive. In order to keep up with the growing competition, developers and suppliers of devices, software and services endeavour to improve their products as much as possible. Hence, systems containing increasingly more sophisticated functions than simple driving directions have been and are still being developed. These range from more advanced algorithms aiming to calculate better routes (i.e. more attractive to the driver, such as fastest route, shortest route, avoiding tolls, preferring motorways etc.) making use of higher quality data (e.g. real-time dynamic travel times rather than static data), to enhanced communication abilities involving real-time traffic information aiming to inform the driver on congested areas in the road network, and even to extra features not directly relevant to navigation, such as weather information, Bluetooth connectivity and media player. Features expected to be included in future navigation systems include automatic speech recognition, 3D map features, photo-realistic textures, multi-function capabilities and wireless internet connectivity (Berg Insight, 2007a).

Nevertheless, the basic structure of navigation systems is fairly similar among the available devices, regardless of the functions offered. The structure described by Zhao (1997) is given here, according to which in-vehicle navigation systems consist of individual modules, each intended to carry out a set of particular functions, as shown in Figure 1.2-3. These are: the digital map database module, the positioning module, the map-matching module, the route planning module, the route guidance module, the wireless communications module and the human-machine interface module.

The digital map database module holds digitised map information such as locations, road geometry, road classifications and traffic regulations with respect to the geographical co-ordinate system, while the positioning module fuses different sensor outputs and radio signals for the determination of the position of the vehicle. Sensors employed mainly include a GPS receiver, which establishes the geographical co-ordinates of the device, though in embedded systems this can also be complemented by a magnetic compass and an odometer for dead-reckoning. Blending the outputs of these two modules together, the map-matching module is responsible for associating the position measured by the positioning module with a location in the digital map database module, since errors in satellite positioning can be fairly significant (Ochieng et al, 2004; Quddus et al, 2007).

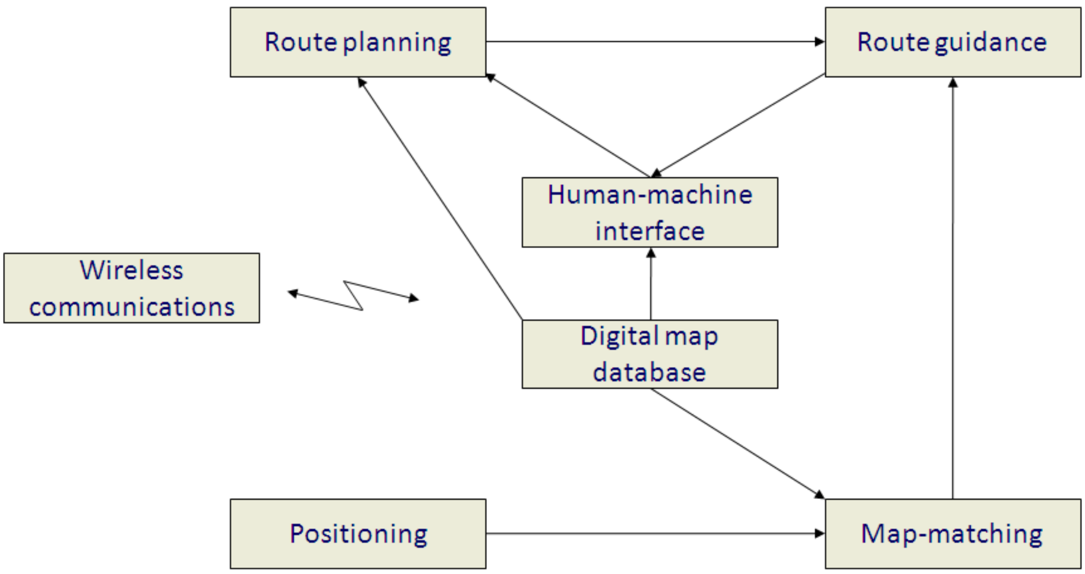


Figure 1.2-3: Modules of in-vehicle navigation systems (Zhao, 1997)

Considering the route planning module, its purpose is to calculate a route or a set of alternative routes for the driver to his/her destination prior to or during his/her journey. The data needed originates mainly from the map database module, but can also be complemented by the wireless communications module, whose aim is to transmit real-time traffic information to the system. The route guidance module on the other hand is responsible for guiding the driver to his/her destination according to the output of the route planning module and based on positioning by the map-matching and digital map database modules. Finally, the human-machine interface module enables the interaction between the driver and the system, transmitting the requests of the former to the latter and displaying its output to him/her.

The present study focuses on the route planning module and a further description of all other modules extends beyond its scope. The state-of-the-art in the route planning module, as reported in the study by Wood et al (2006), includes features such as multi-routing and link impedance. Multi-routing is applied whenever the market penetration level of a system is high, resulting in reduced travel time savings, due to the simultaneous use of the same routes by the vehicles; link impedance on the other hand is implemented whenever particular roads (or road classes) need to be avoided (e.g. to discourage systems from recommending shortcuts through residential areas). Both these techniques are employed later in the present study.

Before stating the aims and objectives of this work in the next section, the next sub-section reports on the status of navigation systems in terms of usage and user satisfaction, as this basically reflects the motivation behind it.

1.2.2 Usage of in-vehicle navigation systems

The fact that in-vehicle navigation systems are rapidly spreading through the vehicle fleet in Europe and North America was reported in the previous sub-section. Nonetheless, in order to identify the requirements for a new navigation system, one needs to look into the usage and evaluation of existing navigation systems from the viewpoint of the drivers. Several market research studies have been conducted and continue to be carried out, aiming to reflect the drivers' needs from a navigation device.

In an early study by Bonsall (1992), it was found that compliance with the route guidance offered by a navigation system decreases with increasing network familiarity, and is in fact reduced even further if the quality of the guidance given to the driver in previous occasions is poor (i.e. erroneous driving directions, inaccurate travel times etc.). The result is that drivers who are familiar with the network cease to trust the navigation system, as they start to believe that their own route-planning is better than the system's (Schofer et al, 1997). It is hence not surprising that, as shown by Svahn (2004), navigation systems are used mostly in environments which are unfamiliar to the driver, and in particular by drivers with a high annual mileage (e.g. taxi drivers, professional van drivers etc.).

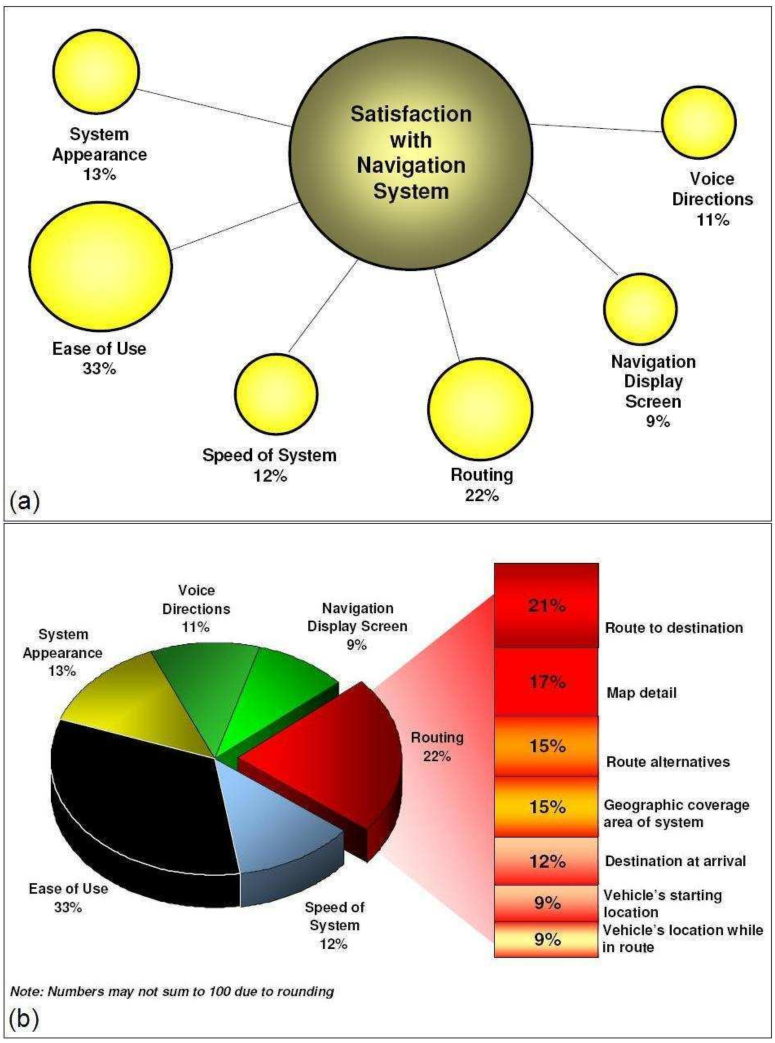


Figure 1.2-4: Factors affecting the satisfaction of users with navigation systems (J.D.Power and Associates, 2006)

The most notable study on navigation systems usage, however, is the annual "Navigation usage and satisfaction study", carried out in the US by J.D. Power and Associates. In the 2006 publication, which is the latest available at the time of the writing of the present study, 118 navigation systems were evaluated using a six-page questionnaire, which was distributed to the users by mail. With a target of 100 returns per model, 14103 usable completed questionnaires were received, which provided a sufficiently large sample (J.D.Power and Associates, 2006).

The study identified that routing is the second most important factor affecting user satisfaction with navigation systems, stated by 22% of the respondents, after ease of use, which was stated by 33% of the respondents (Figure 1.2-4a). Also, it was found that, among the respondents who consider routing as the most important aspect of their system, the majority (21%) value finding a correct route, with no erroneous driving directions, to their chosen destination most; nevertheless, computing route alternatives (15%) and correctly estimating the arrival time at the destination (12%) are also important features affecting the user satisfaction with navigation devices (Figure 1.2-4b).

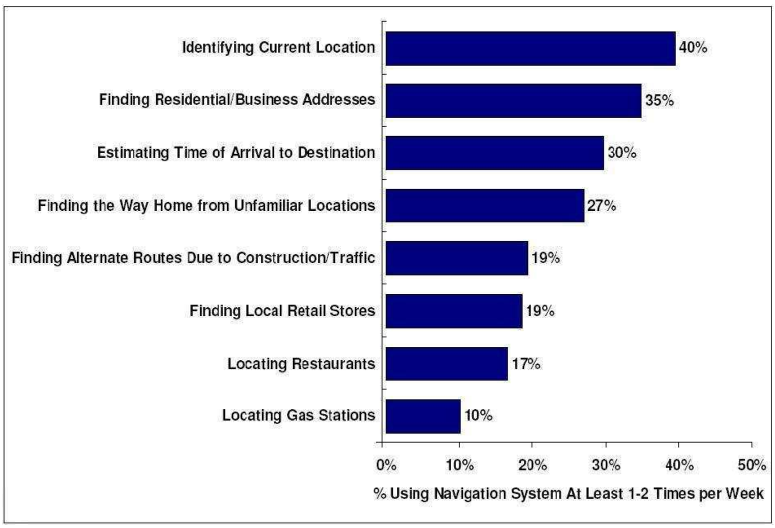


Figure 1.2-5: Purposes of using navigation systems

(J.D.Power and Associates, 2006)

A more important finding of the study was that 30% of the respondents using a navigation system at least once or twice a week use it to estimate the time of arrival at the destination, making this the third most important purpose of using such a system. Furthermore, 19% of the respondents use a navigation system for finding alternative routes to avoid road works or congestion, which in turn represents the fifth most important purpose of making use of such a device (Figure 1.2-5). The study also found that the ability to access subscription-free real-time traffic flow speed and real-time traffic information is at the top of the users' wish list for future features of navigation systems. This indicates that, while drivers are generally happy with their systems, they are not satisfied with their routing function and would like to see it being improved in the near future, particularly as regards the correctness of the routes computed and the accuracy of the estimated time of arrival. This is the motivation behind the research work reported here; its aims and objectives are described in the next section.

1.3 Objectives and outline

It was mentioned in the previous section that users would like to see more advanced features in the routing function of their navigation systems, so that they can obtain more accurate driving directions and travel time estimates, and also avoid congested areas. Incorporating real-time data as a standard feature (currently a premium service) in navigation systems would go some way to solve the problem; however, real-time data is collected, processed and distributed to vehicles by dedicated private companies, and including it as a standard feature in navigation systems would significantly increase their price, something drivers are rather reluctant to consider, according to J.D. Power and Associates (2006). Hence, a method to address the drivers' needs from a navigation system using historical traffic data and only minimal real-time information (incident and road works reports as broadcast on the radio) should be sought.

The aims and objectives of this study are presented first, followed by an outline of the individual chapters.

1.3.1 Aims and objectives

The aim of this work is to develop an advanced in-vehicle navigation routing strategy for an autonomous (on-board) system, taking travel time uncertainty into consideration and providing reliable route guidance (driving directions and estimated time of arrival) to the driver, under the condition of minimal traffic information availability. The new algorithm should be able to recognise, based on historical data, which roads and junctions in the road network are unreliable, reliability referring to the probability of not experiencing abnormal delays due to congestion. Then, it should calculate routes avoiding the unreliable roads and junctions as much as possible, to suggest to the drivers. The ultimate goal is to minimise the actual travel time, while at the same time maximising the reliability of the suggestion, such that the probability of the driver encountering delays is small throughout his/her entire journey.

Additionally, on-board devices have limited computing power (300-600 MHz) and need to support, apart from the route guidance function, a series of additional functions such as visual interface and positioning. Although algorithms are subsequently optimised as much as possible to minimise their computational requirements (e.g. using lower-level programming languages, such as assembly or binary code) for their implementation in on-board devices, computational efficiency (i.e. minimisation of the computation time) is an issue that needs to be considered early in the development of a new in-vehicle navigation algorithm. After all, prior to their implementation in a new navigation system, algorithms are thoroughly tested on conventional computers, and as using low-level programming is not an option due to coding complexity and intractability issues, computational efficiency can only be achieved through the choice and adaptation of the algorithms themselves (or through the use of supercomputers, though this is a highly costly solution). It is thus an additional aim of this work to ensure that the new in-vehicle navigation algorithm is as computationally efficient as possible.

The objectives of the study are:

- to identify a suitable route finding algorithm from a comprehensive literature review of the topic. As the route finding algorithm is a routine that needs to be called fairly frequently and hence be run several times is an on-board navigation device, it needs to be ensured that it is computationally efficient, while at the same being precise (outputting results, that are either optimal, or at least near-optimal). The fact that travel time in a road network is not constant but varies according to the time of the day and day of the week needs to be considered, so that the algorithm's accuracy is not compromised. Also, the resulting time-dependent (dynamic) algorithm should be modified accordingly, so as to be applicable on road networks containing features such as turn restrictions and dead-ends.

- to define a reliability measure to be used by the new routing algorithm. The measure needs to be understandable to the driver, and most importantly, convertible to a maximum delay value reflecting the amount of unpredictable delays that may be encountered along a road segment or entire route. This should be derived from travel time distributions obtained from historical measurements. A review of the topic of travel time uncertainty and reliability is to be carried out to identify a suitable measure; if no suitable measure is found, a new measure of reliability needs to be defined.

- to incorporate the previous two theoretical approaches into a single implementation framework, such that a new reliable dynamic in-vehicle navigation algorithm is formulated, which will use the route finding algorithm to compute route(s) for the user whilst at the same time avoiding potentially congested areas, as identified by the reliability measure. Also, a number of constraints are to be introduced to the optimisation problem that is formulated, so as to ensure acceptability of the route suggestions by the driver. The new algorithm is to be supported by a literature review of previously developed in-vehicle navigation strategies.

- to develop an algorithm for re-routing, based on the same concept, for the case where real-time traffic information on incidents and road works is available and the driver needs to alter his/her itinerary to avoid reported congestion, or for the case where the driver accidentally misses a driving direction.

- to implement the newly derived methodology so that it can be tested and evaluated. A software tool containing the algorithm and also providing a user interface similar to that of an actual navigation system needs to be developed.

- to validate the new approach through a set of specially designed experiments. Apart from an initial simulation experiment that needs to be carried out to test the new software tool and to draw some preliminary conclusions, the new algorithm is to be tested in the field, i.e. in a real-world situation using real data. The results from the new method need to be compared with the output from an existing navigation system in order to demonstrate its potential superiority.

1.3.2 Outline of the thesis

The present study consists of seven chapters, structured so as to reflect the research objectives listed in the previous sub-section. Each chapter consists of different sections and sub-sections, beginning with a short introduction and ending with some concluding remarks.

Introduction introduced the context of the research carried out, including the background and state-of-the art of in-vehicle navigation systems, and the aims and objectives of the study.

Chapter 2 deals with route finding in road networks and starts with a comprehensive literature review of the topics of static and time-dependent path finding. Then, a description of the adaptations made to path finding algorithms with respect to their application on road networks is given; this is followed by a formulation of the chosen route finding algorithm and the modifications made to it so as to maximise its efficiency, and by a presentation of the procedure developed to account for the time-dependent nature of travel time.

Chapter 3 is concerned with travel time uncertainty, variability and reliability, and begins with a review of these topics, including an account of existing measures of travel time reliability. A presentation of a new reliability measure, specifically developed for in-vehicle navigation, is then given, consisting of its basic definition for road segments (links), its extension for routes and its adaptation for the intended implementation.

Based on the definitions and measures presented in Chapters 2 and 3, Chapter 4 introduces the new complete reliable dynamic in-vehicle navigation methodology. Starting from a review of previous relevant research, two new algorithms are presented and mathematically formulated: the first algorithm is for pre-trip planning, where route guidance is sought prior to leaving the origin, whereas the second is intended for planning whilst en-route, in the case where re-routing needs to take place.

Chapter 5 presents the implementation of the algorithms described in Chapter 4 and introduces a new software tool developed for this purpose. The functions and user interface of the program are described. A simulation experiment carried out using the new software tool is presented next, including descriptions of the data simulation and experimental procedures, and an account of the results obtained is given. The simulation experiment constitutes preliminary testing of the new methodology.

Chapter 6 reports on the validation of the new approach through two field experiments carried out in Central London using floating vehicle data. The report includes descriptions of the acquisition and processing of the data required, and of the experimental procedure, as well as the presentation and discussion of the results obtained from both experiments.

The final chapter of the study, Chapter 7, gives an overview of the conclusions drawn from this work, identifies areas of potential further research and addresses some issues of practical implementation.

CHAPTER 2

Route finding in road networks

2.1 Introduction

Path finding (or route finding) is an essential part of an in-vehicle navigation system, as finding a route between two points in the road network is a function that may be called several times during the operation of the system. The range of routing requests in a navigation system is very broad; from simple requests for driving directions from an origin to a destination input by the driver, to advanced real-time routing according to multiple objectives and constraints, alternative routes and re-routing in more sophisticated systems.

This chapter deals with the development of the computational basis of the reliable dynamic in-vehicle navigation algorithm presented in this study, i.e. with the path finding algorithm employed. While the incorporation of advanced features into the routing algorithm, such as travel time uncertainty, alternative routes and constraints, is described in Chapter 4, the network search concept employed is presented here. Since the algorithm is later enriched with additional more advanced functions, it is important that its basic version is not only precise, but also tractable, i.e. easily manageable for large problems.

Though a fairly large number of methods for finding routes through a network have been developed, only few can be used in dynamic navigation, because there are operational requirements (see Section 1.3.1) arising from the fact that routes have to be computed in real-time and have to be supplied to the driver in as little time as possible. The suitability of a method depends both on the computational ability of the navigation device and the efficiency of the algorithm itself, which means that an efficient path finding algorithm is needed. Also, the incorporation of time-dependence, i.e. the consideration of the fact that travel time is not constant on all roads of the road network but varies with time, is a feature that needs to be considered.

The chapter is structured as follows; Section 2.2 gives a comprehensive literature review of route finding algorithms, covering the topics of network searching techniques, shortest path algorithms and time-dependent route finding algorithms. Following that, Section 2.3 describes some characteristics of real road networks and presents methods of dealing with these when applying route finding algorithms on them. Then, the A* shortest path algorithm, which is the one used in this study, is formulated in Section 2.4, followed by a description of a methodology aiming to incorporate time-dependence into it, in Section 2.5.

2.2 Background

This section reviews previous work on route finding. Initially a review of searching techniques is given, including the most widely-used shortest (and fastest) path algorithms, as well as their adaptations by different researchers. Then, the problem of path finding in networks, in which road travel times are not constant but vary over time is tackled and algorithms developed for its solution are presented.

2.2.1 Network searching techniques

Searching for a path in a network with respect to an objective (distance, travel time etc.) is one of the most frequently encountered problems, not only in transport engineering, but also in computer science and operations research (Ahuja et al, 1993). By representing a network as a directed graph consisting of nodes connected by links (corresponding to junctions and roads respectively in transport networks), it is possible to apply any route search method and to form a search tree, starting from a given origin and aiming at a destination. Many algorithms seeking solutions have been developed and can be characterised according to their results as optimal, if they are guaranteed to find the optimal path in the network with respect to the criterion to be optimised (see Section 2.2.2), or heuristic, if they are guaranteed to find any path provided one exists.

The most basic step of each search algorithm is node expansion, which indicates that each node selected for exploration has all of its successors generated before the next one is selected. Namely, as reported in many textbooks such as the ones by Pearl (1984), Ahuja and Magnanti (1993) and Russell and Norvig (2003), path finding algorithms can be grouped in three broad categories, according to the order in which nodes are expanded: depth-first search, breadth-first search and best-first search.

In depth-first search, priority is given to the nodes located at deeper levels of the search tree. Thus, after each expansion one of the newly generated children nodes is selected and this procedure is pursued until for some reason, progress is blocked. This could be either a so-called 'depth-bound', i.e. a stopping rule that, when triggered, returns the algorithm's attention to the deepest alternative not exceeding the bound, or a node recognised as a dead-end. In that case, the algorithm 'backtracks' to the most recent node it had not finished exploring. While this is a simple strategy, it is usually not preferred because it can be highly inefficient. As the method is not guaranteed to find the optimal path, it is characterised as heuristic.

In breadth-first search on the other hand, priority is assigned to nodes located at shallower levels of the search tree. Thus, breadth-first search algorithms proceed by exploring sections of the search tree in layers of equal depth until the destination is reached, and do not proceed to expand any child nodes before all parent nodes of the same layer have been expanded. Like depth-first search, breadth-first search algorithms are heuristic, as they are guaranteed to find a path if one exists which may not, however, be the optimal path. In the special case where such algorithms are applied on networks in which all link weights are equal (unweighted graphs), they are also optimal, as they always output the path with the fewest links. A more advanced version of breadth-first search is uniform-cost search (also called cheapest-first), which is optimal for weighted graphs and instead of progressing in layers of equal depth, it progresses in layers of equal cost, selecting the node with the lowest cost so far in each step (see Section 2.2.2).

Both depth-first search and breadth-first search are uninformed strategies, meaning that no additional information about states beyond that provided in the problem definition is available for them to use. Best-first search on the other hand is an informed strategy, which means that it knows whether expanding a particular node is “more promising” than another based on a so-called ‘evaluation function’, resulting from problem-specific knowledge beyond the definition of the problem itself. Best-first algorithms are not only optimal, but also more efficient (Russell and Norvig, 2003).

The next section reviews algorithms offering solutions to a special problem of route finding, which is the shortest (or fastest) path problem. Optimal algorithms, as well as heuristic adaptations performed by several researchers, are presented.

2.2.2 Shortest path algorithms

The problem of finding the shortest route in a network is of vital importance for a large number of applications (Ahuja et al, 1993). Initially the objective of new solution algorithms was to be able to compute the shortest route optimally; nonetheless it became clear that, since their computation time tends to significantly deteriorate with increasing network size, algorithms also need to be efficient, even if they are not optimal, and provide good heuristic solutions instead. In transport engineering, and more specifically in dynamic navigation, finding the shortest path is a subroutine that needs to be called very often. Due to the fact that the size of transport networks is usually large, it is of vital importance to have a shortest path algorithm, which is efficient enough to meet the real-time operational requirements. Therefore, a considerable amount of research has been devoted to incorporating various acceleration methods for standard shortest path algorithms.

It should be noted here, that the term shortest path does not necessarily refer to the actual distance, but to the criterion that is to be optimised. In the case of in-vehicle navigation, one is usually not interested in the least-distance route, but is more concerned with the least-travel-time (or least-cost or fastest) path. The link weights used for this computation are travel time values, rather than distance values. Therefore, in the problem formulation, the criterion to optimise (minimise) is the total travel time. The terms shortest path, fastest path and least-cost path are used interchangeably in the rest of the study.

The shortest path problem can be divided into four sub-problems. Namely, the so-called 'one-to-one' shortest path problem involves finding the shortest route between a single origin and a single destination in a network. In accordance with that, the so-called 'all-to-all' shortest path problem consists of finding all shortest routes between all origins and all destinations in a network. Equivalently defined, the so-called 'one-to-all' shortest path problem is about finding the shortest routes from one node to all other nodes in a network, while the so-called 'all-to-one' refers to finding the shortest routes from all nodes to a single destination.

Several comparative studies of shortest path algorithms have been carried out (Dreyfus, 1969; Kelton and Law, 1978; Boffey, 1984; Ahuja et al, 1993; Salim, 1997; Zhan and Noon, 1998; Banerjee and Sidhu, 2002; Fu et al, 2006). In particular according to the study by Fu et al (2006), most algorithms follow a standard recursive decision making procedure, consisting of an initialisation step, a node selection step, a node expansion step and a termination step, and only differ from each other in the node selection step, i.e. in the order in which they expand nodes until the destination is reached. In the remainder of this section the standard algorithms for the shortest path problem are presented.

Dijkstra's algorithm

Dijkstra's algorithm, concurrently developed by several researchers such as Dantzig (1960) but formulated and ultimately attributed to the Dutch mathematician E. W. Dijkstra (1959), is the most well-known shortest path algorithm. The procedure followed includes making locally optimal choices at each step (i.e. choosing the node with the lowest cost from the origin) to produce a globally optimal solution, finding the shortest route from the origin node to every other node in the network and thus solving the one-to-all shortest path problem.

The algorithm is optimal and can be categorised as uniform-cost search (see Section 2.2.1). Also it can be characterised as 'label-setting', meaning that for each node expanded it is guaranteed that the shortest path from the origin to that node has been found using only labelled nodes, and that the node will not be re-visited. Consequently, the search can be stopped at any time and still output the shortest paths from the origin to all expanded nodes up to termination, and if it is terminated as soon as the destination node has been expanded, it gives an optimal solution to the one-to-one shortest path problem.

The main advantages of Dijkstra's algorithm being precision and tractability, its main overhead is inefficiency, and more specifically the size of the search space it needs to explore for the one-to-one shortest path problem. Many unnecessary nodes need to be expanded, making the algorithm's running time extremely long and therefore, unsuitable for in-vehicle navigation applications. Bucket implementations of Dijkstra's algorithm, however, prove to be relatively faster (Cherkassky et al, 1994; Zhan and Noon, 1998). These include intermediate sorting of the nodes according to their label in order to determine which node to expand next. Bi-directional implementation is another way of acceleration of Dijkstra's algorithm (see corresponding sub-section).

Bellman-Ford algorithm

A predecessor of Dijkstra's algorithm and thus a more generalised version of it is the Bellman-Ford algorithm (Bellman, 1958; Ford and Fulkerson, 1962). The procedure followed is similar to Dijkstra's procedure, but instead of choosing the node with the lowest cost from the origin to expand next, the algorithm expands all children nodes generated at each step and terminates only when all nodes of the network have been expanded.

The Bellman-Ford algorithm is often characterised as a dynamic programming approach of breadth-first search and is a label-correcting algorithm, meaning that nodes can be re-visited at later stages of the search, and that the shortest path to each node can only be found after termination. As a consequence, it is suitable for the one-to-all shortest path problem but not for the one-to-one shortest path problem. Because it needs to expand a larger number of nodes, it has similar inefficiencies in applications as Dijkstra's algorithm and can therefore not be used in in-vehicle navigation.

Floyd's algorithm

A radically different approach to Dijkstra's and the Bellman-Ford algorithms is Floyd's algorithm (sometimes also referred to as Floyd-Warshall), which is aimed at solving the all-to-all shortest path problem (Floyd, 1962). The procedure behind it is based on the recursive concept that the shortest path between two nodes a and b is either a direct link between them, or a route through a third node c; if it is the latter, the shortest path from a to c is either a direct link or a route through another node d etc. The algorithm's output is two matrices, one of which contains the shortest path distances between all nodes of the network, while the other indicates which node lies on the path between two nodes, so that the path can be traced.

Though the algorithm performs quite well and has fairly low computation times considering the fact that it solves the all-to-all shortest path problem, it is inefficient when only a solution to the one-to-one problem is sought due to the fact that it solves a much larger problem than needed. Additionally, it has been proven that for sparse networks, i.e. networks whose number of actual links is significantly lower than the number of possible links based on the number of nodes, applying a repeated application of a one-to-all shortest path algorithm is more efficient than applying an all pairs shortest path algorithm (Kelton and Law, 1978). As transport networks are considered to be sparse and for in-vehicle navigation only a one-to-one shortest path is needed, Floyd's algorithm is not appropriate.

A algorithm*

Using heuristics is an effective way of reducing computation time and a common way of achieving this is by reducing the search area. The most popular heuristic search algorithm is the A* (pronounced A-star) algorithm (Hart et al, 1968) and the procedure it follows is very similar to Dijkstra's algorithm, the only difference being that the A* uses a heuristic function to estimate the distance from any point to the destination node of the network; this is usually the airline distance (Euclidian distance) to the destination (or, in the case of in-vehicle navigation, the travel time corresponding to that distance). The algorithm is optimal under the condition that the heuristic estimating function is admissible, i.e. does not overestimate the actual distance to the destination. A*'s ability to convert an uninformed search to an informed one results in less nodes being expanded and thus better efficiency.

The A* is classified as a best-first search algorithm and its concept is summarised as follows: the algorithm holds two lists, the open list and the closed list. The closed list contains all the nodes of the network that have been expanded, whereas the open list contains all the nodes that may be expanded at the next step. At each step, one node is expanded and moved from the open list to the closed list, while its successors are placed into the open list. For every node n, the evaluation function $f(n) = g(n) + h(n)$ is calculated, where $g(n)$ is the least distance so far from the origin to node n and $h(n)$ is a heuristic estimate of the distance from node n to the destination. The node to be expanded at each step is the node with the lowest $f(n)$ value, among the nodes in the open list (see Figure 2.2-1). As the A* algorithm is the one used in this study, a more detailed description is given in Section 2.4.

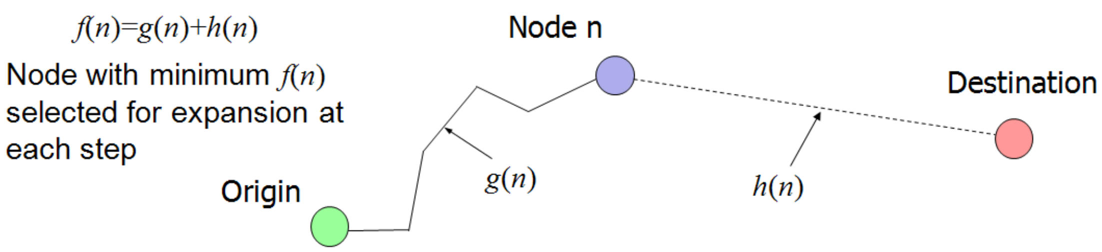


Figure 2.2-1: A* algorithm

Despite outperforming traditional shortest path algorithms while at the same time being optimal, the A* can still sometimes become very inefficient with increasing problem dimensions. The improvement of A*'s efficiency has been the objective of many research studies; some look into using more accurate heuristic estimates than the Euclidian distance (i.e. closer to the actual distance), while others attempt to modify the search procedure.

Jacob et al (1998) present a modified version of the A*, in which using a multiplicative bias factor, the heuristic estimate of the distance to the destination is given more weight than the actual distance from the origin in a node's label. The algorithm performs faster, however this occurs at the cost of precision. A different approach is reported by Bander and White (1991), whose so-called 'Interruptible A*' algorithm computes a sub-optimal path and is either stopped, or carries on to further iterations, so as to converge to the optimal path. In another study by the same researchers, the so-called 'Adaptive A*' is presented, in which the route search is guided by a better evaluation function, based on a collection of real-valued functions on the node set (equivalent to the standard A*'s heuristic function), a set of pre-determined optimal paths and a set of paths in the network that are considered desirable but are not necessarily optimal (Bander and White, 1998). Also, in a study by Lark et al (1995), an improved version of the A* is described, where knowledge acquired from human knowledge sources is used in the algorithm's evaluation function. In studies by Goldberg and Harrelson (2005) and by Lefebvre and Balmer (2007) the so-called 'Landmarks A*' is presented, which combines the standard A* search with a new lower-bounding technique based on landmarks and the triangle inequality, thus achieving better heuristic estimates and reducing the number of nodes expanded. The disadvantage of those methods however is the fact that a larger amount of data needs to be provided, which is often difficult to obtain.

Other research work on the A* algorithm is presented by Korf (1985), who presents the 'Iterative Deepening A*', following a depth-first search procedure but cutting-off the respective branch, when the total $g(n) + h(n)$ value exceeds a certain threshold, and claiming that not only it is optimal when $h(n)$ is admissible, but also that it is simpler to implement, as there is no need of open and closed lists; its drawback, nonetheless, is the need of re-visiting previously visited nodes. Pohl (1971) describes a bi-directional version of the A* (see next subsection), though not with very promising results. Finally, Wichmann (2004) uses the A* for route finding on digital terrains, applies penalties for high gradients and improves the performance of the algorithm by 'sub-sampling', i.e. by splitting the terrain into smaller pieces, however at the cost of precision. It can be thus concluded that for in-vehicle navigation applications, the traditional form of the A* algorithm is most suitable.

Bi-directional algorithms

Another method of heuristic search is to decompose the problem into smaller problems, and bi-directional search is the commonest way of achieving this. Namely, it is conjectured by many researchers that starting two search trees from the origin and the destination simultaneously should result in the two trees meeting halfway through the search space, thus reducing the total number of nodes expanded and achieving lower computation times.

The idea is first suggested by Nicholson (1966), who implements a bi-directional version of Dijkstra's algorithm, outputting some promising results, as the algorithm is both efficient and optimal. However, as is later identified by Dreyfus (1969), an important factor influencing the efficiency of a bi-directional algorithm is its stopping criterion, and the criterion used in Nicholson's algorithm makes it inferior to uni-directional algorithms. In a bi-directional implementation of the A* algorithm (Pohl, 1971), this finding is further enhanced, as it is shown that the two search trees seldom meet at the middle of the search space and grow on to become large uni-directional search trees. The study compares the two trees as two missiles passing each other. An improved version of Pohl's bi-directional A* is the so-called 'front-to-front' technique, where the objective of the two trees is to find each other, rather than to find the origin/destination. An implementation of this is reported in a study by de Champeaux and Sint (1977), as well as in a later study by de Champeaux (1983).

Despite the potential advantages that bi-directional algorithms may exhibit in terms of computation time, this type of search is not suitable for in-vehicle navigation applications, because it cannot be extended to solve the shortest path problem under time-dependence (see Section 2.2.3).

Hierarchical algorithms

A technique to accelerate shortest path algorithms is to reduce the number of searched links; this can be done by implementing a hierarchical model and dividing up the network into different sub-networks, and thus determine which links are more likely to be close to the shortest path and are therefore more 'worth' exploring. The various algorithms that are reported in different studies are concerned with reducing the computation time as much as possible. As identified by Fu et al (2006), the efficiency of hierarchical search algorithms depends on how a real road network is cast into a hierarchical framework (i.e. how many layers are used etc), and on how to control the search transition between layers (i.e. when to stay on a level and when to move to the next one).

In a study by Liu (1997), a hierarchical search algorithm is presented, where the network is split into two layers, one containing the major road network and the other consisting of smaller sub-networks of minor links. While the algorithm takes into account network features such as dead-ends, shortcuts and underpasses, and appends the appropriate links to the corresponding network level, it is based on the assumption that drivers prefer travelling on major roads, which is not necessarily true. The results obtained in terms of computation times are encouraging, though the solutions are not optimal any more.

In other studies, different hierarchical approaches are developed. Hock and Srikanthan (2001) adopt a similar hierarchical framework but identify that the origin and destination rarely lie on the higher level. Hence they describe a routing algorithm, which is based on promoting these to the higher level, resulting in an improvement in efficiency. Park et al (2001) introduce a methodology, which aims at limiting the transitions from one layer to the other, while Jagadeesh et al (2002) combine hierarchical search and network pruning, so as to further improve efficiency with what they call an “acceptable loss” of precision. Finally, Holzer (2003) makes use of hierarchical decomposition and pre-computed information to develop and analyse a “speed-up technique” for shortest path algorithms.

Though hierarchical search algorithms exhibit greater efficiency than others, their main drawback is the fact that the routes computed are almost guaranteed that they will not be optimal, as it may be possible that the fastest route will be completely missed by the algorithm, which will output a much longer route instead, simply preferring main roads. This may particularly be a problem in areas, where no main roads are present close to the origin and destination.

Other approaches and accelerating techniques

Another approach aiming to efficiently compute the shortest route is branch-pruning. In branch-pruning, an attempt to reduce the search area of an algorithm to a smaller window, in which only links more likely to be located on the shortest path due to their location and direction are contained, is made (Karimi, 1996); the result is that the heuristic algorithm used turns out to be fairly efficient, though not optimal. A similar technique is implemented by Quek and Srikanthan (2002), where offline pre-processing and pruning of the network is carried out.

Some research has attempted to improve the speed of the shortest path computation by applying accelerating measures to existing algorithms. An example of this is the work of Habbal et al (1994) and of Ziliaskopoulos et al (1997), who report on an implementation of shortest path algorithms on parallel computer architectures. These techniques cannot be used for in-vehicle navigation algorithms though, as parallel computer architectures cannot be practically realised in on-board units.

2.2.3 Time-dependent shortest path algorithms

When computing paths in a road network for in-vehicle navigation, traffic conditions are subject to continuous changes, such that the link travel times are not constant anymore, but time-dependent. Therefore, the task to be carried out is finding the shortest path in a dynamic, time-dependent network, where the travel times of links are not reflected by a single value but by a series of values, each corresponding to a different time interval of a day, and where different departure times result in different travel times experienced. The problem has been extensively studied in the past, not only in the context of transport engineering, but also in the context of communication systems.

Early work on the time-dependent shortest path problem is reported by Cooke and Halsey (1966), who present an adaptation of the Bellman-Ford algorithm, and by Dreyfus (1969), who extends that method to Dijkstra's algorithm. More recently, Orda and Rom (1990) present a theoretical study describing algorithms for solving the problem, while in a study by Ziliaskopoulos and Mahmassani (1993) the computation of shortest routes for all possible departure times whilst representing the label of a node by a vector instead of a single scalar, as is the case in static networks, is suggested. A different approach, adopted by Kim and Jung (2002), is to repeatedly apply a static shortest path algorithm over a certain number of time intervals, in order to account for changes in the link travel times; at the same time, it is attempted to reduce the computation time for subsequent searches by making use of the computed initial shortest path and only changing the elements of the path, whose weights have changed.

Nevertheless, an important finding is the identification of the fact that standard shortest path algorithms can be applied to time-dependent problems with worst-case computation times identical to those in static networks, provided the so-called consistency condition holds, according to which the travel time on a link should not decrease faster than actual time increases (Kaufmann and Smith, 1993). Although consistency bears some relation to the first-in-first-out (FIFO) principle of traffic flow, i.e. that vehicles will exit from a link in the same order they entered it, it is not assumed that vehicle interaction in the network is actually FIFO. In fact, it is very likely that FIFO is violated in reality, as vehicles may be travelling at different speeds on a link at a specific time resulting in faster vehicles overtaking slower ones. This, however, does not imply that consistency does not hold, since the lower travel times experienced by the faster vehicles are not a result of the variation of the traffic conditions (reflected in the variation of the space-mean speed) but only relate to the individual vehicles (microscopic level). Kaufmann and Smith thus note that consistency holds in road networks and should therefore be enforced in time-dependent travel time forecasting models. Based on that, Chabini and Lan (2002) present an adaptation of the A* algorithm for finding the shortest path in a time-dependent network, on which consistency holds.

In a study by Horn (2000) it is proven that consistency is implied if the following assumptions hold: 1) the network is invariant, 2) the lengths and travel speeds of the links are real values and greater than zero, 3) travelling occurs as fast as the links' characteristics (topology, speed limit) permit and 4) speed is the same for all parts of a link. Assuming a so-called 'constant acceleration regime', i.e. that in a specific time interval link speed increases/decreases linearly, the study presents a methodology of incorporating time-dependence into various one-to-one and one-to-all shortest path algorithms, including Dijkstra's and A*, and reports on the results obtained from their comparison. The comparison is performed over a period of 12 hours (12am to 12pm) and four time intervals are defined, reflecting peak and off-peak times.

Following a similar concept, the most important contribution to the field of time-dependent shortest path algorithms is a study by Sung et al (2000), whose so-called 'flow speed model' (FSM) is an efficient method of ensuring consistency in a time-dependent forecasting model and finding the shortest route in any time-dependent network. The main idea behind Sung's FSM is that, while travel time intervals may result in discontinuities in the link travel time distribution, causing consistency to be violated in the model, speed intervals do not affect the corresponding link speed distribution. Basically, an abrupt increase in link speed results in all vehicles on the link accelerating. A change in the speed of a link results in different vehicles experiencing different travel times on the link, according to their location at the time of the change; this is not taken into consideration in any of the previous approaches and is the main advantage of the FSM compared to them. Sung's FSM is the method adopted to reflect time-dependence in this study, and is described in more detail in Section 2.5.

Other work on the time-dependent shortest path problem includes research on the so-called 'stochastic time-dependent least-time path problem'; however, as this also covers the topic of path finding under uncertainty, it is reviewed in Chapter 4.

2.2.4 Summary

Having reviewed the topics of static and time-dependent path finding, the following remarks can be made:

- While depth-first procedures are not optimal, breadth-first search procedures can be optimal under certain conditions. Best-first algorithms, on the other hand, are not only optimal, but are also more efficient.

- Though many algorithms (and modifications thereof) exist for finding the shortest path in a network and many techniques for improving their efficiency have been developed, the most suitable algorithm for in-vehicle navigation applications is the standard A* algorithm, using the Euclidian distance as the most admissible heuristic, as it is both precise and efficient for processors of navigation devices (see Section 1.3.1).

- When searching for the shortest route in a time-dependent network, standard static shortest path algorithms can be used and guaranteed to be optimal, provided that consistency holds. From a large number of methods existing in the literature, the most suitable to be used in in-vehicle navigation applications is Sung's flow speed model.

2.3 Modelling road network features

The traditional algorithmic structure used by label-setting or label-correcting path finding algorithms (such as the A* algorithm) on artificial grid networks is the one of a directed or undirected graph. The network consists of nodes and links, an origin and a destination node are specified and the implemented algorithm proceeds by setting or altering labels, placed on each node. Nevertheless, when attempting to implement such algorithms on real road networks for the purpose of in-vehicle navigation, this structure creates problems, mainly with respect to three issues: representation of network features, representation of turn restrictions, and positioning of the vehicle in the network. Therefore, modifications to this structure are made in this study in order to convert it to a form which reflects these features. The approaches developed in order to account for the above issues are described in the next sections.

2.3.1 Representation of road network features

The representation of the features of road networks needing to be modelled so that route finding algorithms can be applied on them is reported here. The features under consideration include one-way streets and dead-ends. Regarding the former, networks are often represented by undirected graphs; nonetheless, in the case for real road networks, due to the presence of one-way streets and two-way roads with different characteristics in each direction, it has to be ensured that all links are directed.

A practice that is frequently encountered is the representation of roads as two-way links by default, and the closure of one direction in the case of one-way streets by either setting its traffic flow capacity to zero or by setting its speed limit to zero; such is the representation of the network in the PTV VISUM™ traffic analysis tool. However, this has disadvantages in cases where in a two-way street the two directions have different geometries; it that case, two different two-way links have to be created, each having one direction closed to all traffic, which is very inefficient as many more elements than needed have to be present. The method that is chosen in this study is therefore one, where each link element is one-way by default and possesses a set of characteristics; in case a two-way road is needed, this is represented by two separate opposing single-direction links.

As concerns dead-end roads, these also need to be represented by two opposing single-direction links, since a vehicle that enters a dead-end has to be able to drive out of it. Also, a permitted U-turn from the inbound to the outbound link should be present so as to ensure that driving in and out of the dead-end is possible.

2.3.2 Turn restrictions

The main feature, which differentiates real road networks from artificial networks, is the existence of turn restrictions. Right turns in the UK (left turns in Continental Europe and in the US) are very often banned, due to the fact that there is not adequate space to accommodate the formation of the resulting queue of turning vehicles. This can also be the case for left turns in the UK, when the turning radius is not long enough to enable turning of larger vehicles, or even straight-on movements, when they cannot be accommodated in the traffic signal cycle of the corresponding junction.

Several attempts to model turn prohibitions have been made in the past. The first method is proposed by Wattleworth and Shuldiner (1963). The approach adopted is to substitute every junction by a smaller sub-network of dummy nodes and links. Due to its simplicity, this approach has been widely used in transport applications. However, it has two major drawbacks: it significantly alters the network structure and it requires a great deal of computing power to accommodate and process all the additional elements that are created. As an indication, a node representing a four-arm junction corresponds to four dummy nodes and twelve dummy links, meaning that for each node of the network, sixteen new elements have to be created.

To overcome the computational inefficiency caused by the creation of dummy elements, various different techniques have been introduced. An important contribution to the field is the study by Kirby and Potts (1969), where a new method is proposed, according to which penalties are applied to turning movements. For prohibited turns, the corresponding penalty value is set to infinity. A similar approach is also adopted in a study by Easa (1985). Another important contribution is the 'extended forward star structure' (EFSS), introduced by Ziliaskopoulos and Mahmassani (1996). According to this, there is a list with all allowed movements in the network and every node holds as many labels as links emanating from it. Shortest path algorithms can then be executed on this modified network structure, while all labels are updated gradually.

Although both these methods eliminate the need for dummy elements, reducing thus the required amount of computing power, their drawback lies in the fact that each node has to hold multiple labels, hindering thus the application of label-setting and label-correcting algorithms (such as A*). More specifically in the example of the A*, if a node is reached from one direction, some of its successor nodes will not be added to the open list because the respective turns will be banned. Only the node's labels corresponding to allowed turns will be updated and the node will be moved to the closed list, despite the fact that there will be successor nodes which can be reached from a different direction but which will not be in the open list and will therefore never be explored. Therefore, a node cannot be considered to be expanded and placed in A*'s closed list unless all of its labels have been updated, i.e. it has been visited from all possible directions. This results in a very large number of nodes in the open list and very few nodes in the closed list.

The approach adopted here is a modified version of the EFSS. Namely, similarly to the EFSS, a list of all allowed movements is held, where each movement is represented by three nodes (start node - middle node – end node) or by two links (start link – end link). Instead of assigning each node multiple labels, according to the number of links emanating from it, the labels are placed on the links, as shown in Figure 2.3-1. Each link is thus treated as two parts (start and end), each part holding its own label.

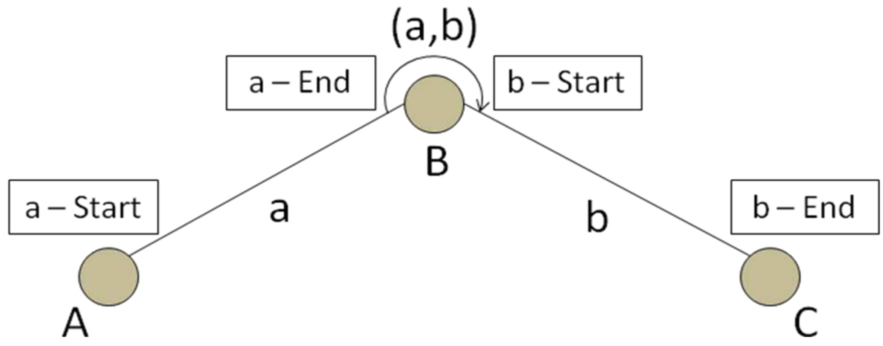


Figure 2.3-1: Representation of junction movements placing labels (framed) on the links

Label-setting path finding algorithms are modified accordingly. Two open lists (open start and open end list) and two closed lists (closed start and closed end) are kept, containing links instead of nodes. Each link part (start or end) is treated as a node in the conventional version of the algorithm as regards the manipulation of the labels. Thus, if the start part of a link has its label updated, it is placed in the open start list and if its end part has its label updated, the link is placed in the open end list. Similarly, if the start or the end part of the link is expanded, the link is removed from the open start or open end list and is placed in the closed start or closed end list respectively.

This is a very efficient method of representing intersection movements, as it enables the application of path finding algorithms without introducing any extra elements to the network. A new adaptation of the A* algorithm to this technique is made in this study, however this is described in more detail in Section 2.4, where a mathematical formulation is also presented.

2.3.3 Positioning of the vehicle

In artificial networks the origin and destination of a route, as well as the position of the vehicle, are identified as nodes. To be exact, any journey always starts from an origin node and ends at a destination node. However, in real road networks, where the nodes correspond to junctions, this approach has a serious drawback, and that is the fact that not only the position of the vehicle is required, but also its direction. It is possible, that a vehicle is located on a road between two junctions, facing towards one of them and not being able to make a U-turn; if the position of the vehicle is expressed as the nearest node, it is very likely that a recommended route requires the vehicle to make illegal or impossible movements.

Figure 2.3-2 clarifies the practical implementation issue described in the previous paragraph. As can be seen, starting from the origin node, two routes are computed to the destination (dotted line and continuous line). However, the two routes depart in opposing directions from the origin node, assuming that the vehicle is located exactly on that junction and that the driver is able to drive in any direction. This is very unlikely to occur in real life.

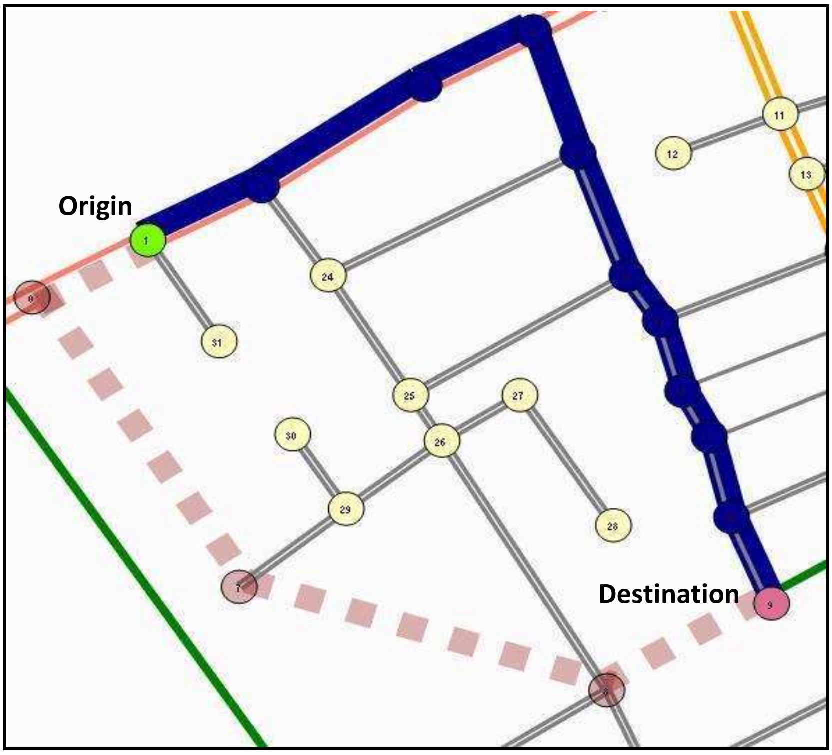


Figure 2.3-2: Problem of using nodes in path finding in real road networks

A solution to the issue presented here is to slightly modify the network structure and make it link-based rather than node-based. Namely, when specifying the origin and destination, an origin-link and a destination-link are specified, considering that the origin or the destination respectively are located somewhere on this link. This can be inaccurate for longer links such as rural stretches of interurban motorways, where the distance between two nodes (corresponding to exits) can be several kilometres. In urban environments however, links are much shorter and are rarely longer than 200 metres long, which considerably limits the negative effects of such an approximation. As the present study focuses on urban areas, this level of accuracy is considered to be acceptable.

Using this technique, the setting of the origin link or the current position link immediately limits the next allowed movements. Similarly, since links are single-directional, the setting of the destination link results in the direction of approach of the destination being fixed, which is very important in some cases. In any case, when the user is prompted to enter the destination of his/her trip, it is most likely that this will be a street name, therefore only referring to a link.

2.4 The standard A* algorithm

As was derived from the literature review of Section 2.2, the most suitable route finding algorithm for an in-vehicle navigation system is the A*, as introduced by Hart et al (1968). In an in-vehicle navigation algorithm, nevertheless, it is possible that multiple subsequent A* runs are required, in order to either compute multiple routes, or to re-compute a single route so as to meet a number of constraints. For that case, Chen et al (2006) suggest using information obtained from previous runs in subsequent ones; this can be achieved by performing the runs in opposite directions, such that an initial A* search is run backwards, i.e. from the destination to the origin, while the subsequent searches are run forwards as normal, i.e. from the origin to the destination. By using the labels from the reverse run as heuristic estimates in the forward runs, the efficiency of the forward runs can be significantly improved, since the estimates will be accurate for the nodes visited by the reverse run and fewer new nodes will have to be expanded.

A modified version of Chen's suggestion is developed in this study and is described in detail in Chapter 4. This section focuses on the A* algorithm, both forward and reverse, and presents mathematical formulations. Also, it is demonstrated that running the A* in either direction is equally optimal.

The algorithm formulations are presented taking into account the ways of modelling real road network features so that path finding algorithms can be applied in in-vehicle navigation. A description of the forward A* (FA*) algorithm is given first, followed by a proof of its optimality; then a formulation of the reverse A* (RA*) algorithm is given, followed by the corresponding proof of its optimality.

2.4.1 The FA* algorithm

As reported in Section 2.2.2, the A* is a best-first search algorithm, holding an open and a closed list containing nodes. Implementing the turn restrictions representation method introduced in Section 2.3.2, according to which two labels are placed on every link at its start and end points, the algorithm keeps four lists holding links: open-start, open-end, closed-start and closed-end. A link can be concurrently contained in both open-start and open-end or closed-start and closed-end, but not in two lists of the same type (i.e. open-start and closed-start).

The algorithm proceeds by examining all links in open-start and open-end and choosing the link l and part x ($x=s$ for start or $x=e$ for end) with the lowest evaluation function value $f(l_x)$. The evaluation function is defined as:

$$f(l_x) = g(l_x) + h(l_x) \tag{2.4-1}$$

where $g(l_x)$ is the travel time from the end part of the origin link o_e to l_x and $h(l_x)$ is the heuristic estimate of the travel time from l_x to the start part of the destination link d_s. For the chosen link part the alternative next positions (i.e. link parts) are deduced; if $x=s$, then the only available next position is l_e and hence l is added to the closed-start and open-end lists, while it is removed from the open-start list; the label $f(l_e)$ is worked out using equation 2.4-1. If on the other hand $x=e$, the set of successor links $\Gamma(l)$ of l is generated according to the existing allowed movements, l is removed from the open-end list and added to the closed-end list, while all the links of $\Gamma(l)$ are added to the open-start list and their f labels are worked out using equation 2.4-1. The algorithm stops when the destination link d has been added to the closed-start list.

For each link part l_x the heuristic travel time estimate $h(l_x)$ to the destination is calculated by considering the start node of link l if $x=s$ and the end node of link l if $x=e$. Assuming that it is possible to travel at the speed limit on a motorway (highest available road category, also see Chapters 5 and 6) following a straight line between the corresponding node and the start node of the destination link d, an estimate of the travel time to the destination is obtained; as this is always going to be a lower bound to the actual travel time value, this can be used as the heuristic estimate in the FA* algorithm.

The mathematical formulation of the FA* algorithm in road networks follows. Some definitions and notation are given first, followed by the formulation of the procedure.

Definitions and notation

Consider a network G, consisting of a set of nodes N, a set of directed links V and a set of movements M, specifying the allowed turning movements between links. Movements are represented by their start and end links, such that movement (a,b) connects links a and b.

o:	The origin link
d:	The destination link
l_x:	Part of link l, s for start, e for end
$t(l)$:	The travel time of link l
$\lambda(l)$:	The length of link l
$\delta(a,b)$:	The delay of movement (a,b)
$g(l_x)$:	The actual travel time from o_e to l_x
$h(l_x)$:	The estimated travel time from l_x to d_s
$f(l_x)$:	Evaluation function of the travel time from o_e to d_s through l_x
$OP_s \subset V$:	Open-start list
$OP_e \subset V$:	Open-end list
$CL_s \subset V$:	Closed-start list
$CL_e \subset V$:	Closed-end list
$\Gamma(l)$:	The set of successor links of link l
$s^{-1}(l)$:	The predecessor link of l on the optimal path
L:	The list of links in the optimal path
T:	The travel time of the optimal path
Λ:	The length of the optimal path

Procedure

<u>Algorithm FA*</u>

Step 0 (Initialisation): Set $OP_e = \{o_e\}$, $OP_s = CL_s = CL_e = L = \varnothing$, $g(o_e) =$

	0, $g(o_s) = \infty$, $h(d_s) = 0$, $T = 0$, $\Lambda = 0$.
	$\forall\ l \in V \backslash \{o\}$ set $g(l_s) = g(l_e) = \infty$.
	$\forall\ l \in V \backslash \{d\}$ input $h(l_s)$ and $h(l_e)$. Input $h(d_e)$.
Step 1 (Select link part for expansion):	$l_x^{exp} = \mathrm{Argmin}_{l_x \in (OP_s \cup OP_e)} [f(l_x) = g(l_x) + h(l_x)]$.
	$CL_x = CL_x \cup \{l_x^{exp}\}$, $OP_x = OP_x \setminus \{l_x^{exp}\}$.
Step 2 (Branch and update):	If $x = s$
	If $l_s^{exp} = d_s$ then go to Step 3.
	$OP_e = OP_e \cup \{l_e^{exp}\}$, $g(l_e^{exp}) = g(l_e^{exp}) + t(l^{exp})$,
	and, if $l_e^{exp} \in CL_e$, $CL_e = CL_e \setminus \{l_e^{exp}\}$.
	Else if $x = e$
	$\forall\ u \in \Gamma(l^{exp})$:
	$OP_s = OP_s \cup \{u_s\}$.
	If $g(u_s) \geq g(l_e^{exp}) + \delta(l^{exp},u)$,
	then $g(u_s) = g(l_e^{exp}) + \delta(l^{exp},u)$, $s^{-1}(u) = l^{exp}$
	and if $u_s \in CL_s$, $CL_s = CL_s \setminus \{u_s\}$.
	Go to Step 1.
Step 3 (Trace path and terminate):	Set $l' = d$.
	Repeat until $l'' = o$:
	$l'' = s^{-1}(l')$, $T = T + \delta(l'',l') + t(l'')$, $\Lambda = \Lambda + \lambda(l'')$,
	$L = L \cup \{l''\}$ and set $l' = l''$.
	Output fastest path, containing the links in L with travel time T and length Λ. Terminate.

It should be noted here that if for each link part l_x the heuristic estimate to the destination is set to $h(l_x) = 0$, the FA* algorithm reduces to Dijkstra's algorithm, as it follows the same procedure, the only difference being the fact that no information about the network is used, making the search uninformed.

2.4.2 Optimality of the FA* algorithm

It is proven by Hart et al (1968) that the standard A* algorithm (i.e. FA*) is optimal, i.e. that the label $g(n)$ of a node n is equal to the least cost from the origin to n. This sub-section presents a modified version of the optimality proof, purposely referring to in-vehicle navigation applications in real road networks with turn restrictions. The proof follows the series of propositions reported in the study of Hart et al, nevertheless using the network structure introduced in Section 2.3 in the same way that the formulation of the algorithm is given.

The notation introduced in the previous sub-section also applies here, with some additional notation and definitions being introduced. Firstly, it should be noted that a part of a link contained in the open-start or open-end lists is designated as 'open', while it is referred to as 'closed' if the link is contained in the closed-start or closed-end lists. Also, $t_{min}(a_x,b_y)$ denotes the least travel time between two link parts a_x and b_y. If $x=s$, then the travel time $t(a)$ of link a is included in $t_{min}(a_x,b_y)$, while if $x=e$, $t(a)$ is not included. Similarly, $y=s$ implies that $t(b)$ is not included in $t_{min}(a_x,b_y)$, while $y=e$ means that it is included. Finally, $t(m_x,n_y)$ denotes the travel time between two connected link parts, either the start and end parts of a single link, or the end and start of two adjoining links connected by a movement. In the former case, $m=n=c$ while $x=s$ and $y=e$; thus $t(m_x,n_y) = t(c_s,c_e) = t(c)$. In the latter case, $m \neq n$ while $x=e$ and $y=s$; thus $t(m_x,n_y) = t(m_e,n_s) = \delta(m,n)$, where $\delta(m,n)$ is the delay of the movement (m,n) connecting links m and n.

The following propositions pre-suppose that between any pair of links there is a path with minimum travel time, designated by P_{min}. Proposition 2.4-1 is the key to the others.

Proposition 2.4-1: For any non-closed link part l_x and fastest path P_{min} from o_e to l_x, there is an open link part $i_y \in P_{min}$ with $g(i_y) = t_{min}(o_e,i_y)$.

Proof: If o_e is open (first iteration of the algorithm is not yet complete) then $i_y = o_e$ and $g(o_e) = t_{min}(o_e,o_e) = 0$ and Proposition 2.4-1 is trivially true. Suppose o_e is closed. Let *CL** be the set of closed link parts on P_{min} for which $g(i_y) = t_{min}(o_e,i_y)$ if $i_y \in CL^*$. CL^* is not empty, as $o_e \in CL^*$. Choose $l_z \in CL^*$ with the largest $t_{min}(o_e,l_z)$ and call this l_z^*. Note that $l_z^* \neq l_x$, as l_x is non-closed. Let $i_y \in P_{min}$ be the successor of l_z^* (possibly $i_y = l_z$). By Step 2 of the algorithm it is known that $g(i_y) \leq t_{min}(o_e,l_z^*) + t(l_z^*,i_y)$. But $t_{min}(o_e,i_y) = t_{min}(o_e,l_z^*) + t(l_z^*,i_y)$ as $i_y \in P_{min}$. Hence $g(i_y) \leq t_{min}(o_e,i_y)$. Since by definition $g(i_y)$ is the lowest travel time encountered so far, $g(i_y) \geq t_{min}(o_e,i_y)$. Thus $g(i_y) = t_{min}(o_e,i_y)$. Moreover, $i_y \notin CL^*$ as $t_{min}(o_e,i_y) > t_{min}(o_e,l_z^*)$

since $t(l_z^*,i_y) > 0$. So i_y must be open, otherwise $i_y \in CL^*$. QED

Choose a heuristic $h(l_x) \geq 0$ such that $h(l_x) \leq t_{min}(l_x,d_s)$ for all $l \in V$. This implies that $h(d_s) = 0$. The heuristics thereby represent lower bounds for the cost from l_x to d_s.

<u>Proposition 2.4-2</u>: Suppose that FA* has not terminated and that P_{min} is the least travel time path from o_e to d_s. There is an open link part $i_x \in P_{min}$ such that $f(i_x) \leq t_{min}(o_e,d_s)$.

<u>Proof</u>: By Proposition 2.4-1 there is an open node $i_x \in P_{min}$ with $g(i_x) = t_{min}(o_e,i_x)$. As the algorithm has not terminated this could be d_s. So $f(i_x) = g(i_x) + h(i_x) = t_{min}(o_e,i_x) + h(i_x) \leq t_{min}(o_e,i_x) + t_{min}(i_x,d_s)$. As $i_x \in P_{min}$, $t_{min}(o_e,i_x) + t_{min}(i_x,d_s) = t_{min}(o_e,d_s)$. QED

<u>Proposition 2.4-3</u>: The least travel time path is found by the FA* algorithm in real road networks.

<u>Proof</u>: Assume otherwise. There are two possibilities, the algorithm fails to terminate (Case 1) or it terminates without finding the least cost path (Case 2).

Case 1: Without the re-visiting of closed link parts, the algorithm terminates after at most $|2V|$ iterations, where $|V|$ is the number of links, as one open link part is closed at each iteration (Step 1 of the algorithm). No link part further in cost terms than $t_{min}(o_e,d_s)$ from d_s will be expanded, as by Proposition 2.4-2 there exists an open link part i_x within a travel time of $t_{min}(o_e,d_s)$ from d_s directly before termination and this will be expanded instead. A closed link part i_x is only reopened when $g(i_x)$ is reduced and adding a loop to the original path will not do this. No optimal path will contain loops. There are a finite number of non-looping paths between any pair of links. Hence each part can only be closed (and therefore opened) a finite number of times, so the algorithm must terminate.

Case 2: Suppose the algorithm terminates with $f(d_s) = g(d_s) > t_{min}(o_e,d_s)$. By Proposition 2.4-2, there exists just before termination at d_s an open link part i_x on the least travel time path such that $f(i_x) \leq t_{min}(o_e,d_s) < f(d_s)$. Thus i_x would have been selected for expansion instead of d_s, contradicting the assumption that the algorithm terminated. QED

It can be therefore concluded that the FA* algorithm is optimal in real road networks, as it finds the fastest path from the origin to the destination. The following two sub-sections present the RA* algorithm and its optimality proof.

2.4.3 The RA* algorithm

In the same way as the FA* algorithm, the objective of the RA* algorithm is to find the fastest path between the origin o and the destination d, however starting the search from the start part of the destination link d_s and aiming for the end part of the origin o_e and using all links with their directions reversed. The algorithm proceeds by examining all links in the open-start and open-end lists, choosing the link l and part x ($x=s$ for start or $x=e$ for end) with the lowest evaluation function value $f(l_x)$ defined according to Equation 2.4-1; $g(l_x)$ and $h(l_x)$ are defined differently though, since $g(l_x)$ represents the travel time as computed by the algorithm at that stage from l_x to d_s, while $h(l_x)$ is a heuristic estimate of the travel time from o_e to l_x. The latter can be calculated either based on the Euclidian distance, if the RA* is the first path finding algorithm run of an in-vehicle navigation strategy, or using the labels set by a previous run of FA*.

For the chosen link part l_x the alternative previous positions (i.e. link parts) are deduced; if $x=e$, then the only available previous position is l_s and hence l is added to the closed-end and open-start lists, while it is removed from the open-end list; the label $f(l_s)$ is also worked out using equation 2.4-1. If on the other hand $x=s$, the set of predecessor links $\Gamma^{-1}(l)$ of l is generated according to the existing allowed movements in M, l is removed from the open-start list and added to the closed-start list, while all the links of $\Gamma^{-1}(l)$ are added to the open-end list and their f labels are worked out from equation 2.4-1. The algorithm stops when the origin link *o* has been added to the closed-end list.

The mathematical formulation of the RA* algorithm is given next.

Definitions and notation

The notation used in the formulation of FA* is used here; some additional and differing notation introduced here is the following:

$\Gamma^{-1}(l)$:	The set of predecessor links of link l
$s(l)$:	The successor link of l on the optimal path
$g(l_x)$:	The actual travel time from l_x to d_s
$h(l_x)$:	The estimated travel time from o_e to l_x
$f(l_x)$:	Evaluation function of the travel time from o_e to d_s through l_x

Procedure

Algorithm RA*

Step 0 (Initialisation): Set $OP_s = \{d_s\}$, $OP_e = CL_e = CL_s = L = \varnothing$, $g(d_s) = 0$, $g(d_e) = \infty$, $h(o_e) = 0$, $T = 0$, $\Lambda = 0$.

$\forall\ l \in V\backslash\{d\}$ set $g(l_s) = g(l_e) = \infty$.

$\forall\ l \in V\backslash\{o\}$ input $h(l_s)$ and $h(l_e)$. Input $h(o_s)$.

Step 1 (Select link part for expansion): $l_x^{exp} = \text{Argmin}_{l_x \in (OP_s \cup OP_e)} [f(l_x) = g(l_x) + h(l_x)]$.

$CL_x = CL_x \cup \{l_x^{exp}\}$, $OP_x = OP_x \setminus \{l_x^{exp}\}$.

Step 2 (Branch and update): If $x = e$

If $l_e^{exp} = o_e$ then go to Step 3.

$OP_s = OP_s \cup \{l_s^{exp}\}$, $g(l_s^{exp}) = g(l_s^{exp}) + t(l^{exp})$,

and, if $l_s^{exp} \in CL_s$, $CL_s = CL_s \setminus \{l_s^{exp}\}$.

Else if $x = s$

$\forall\ u \in \Gamma^{-1}(l^{exp})$:

$OP_e = OP_e \cup \{u_e\}$.

If $g(u_e) \geq g(l_s^{exp}) + \delta(u,l^{exp})$,

then $g(u_e) = g(l_s^{exp}) + \delta(u,l^{exp})$, $s(u) = l^{exp}$

and if $u_e \in CL_e$, $CL_e = CL_e \setminus \{u_e\}$.

Go to Step 1.

Step 3 (Trace path and terminate): Set $l' = o$.

Repeat until $l'' = d$:

$l'' = s(l')$, $T = T + \delta(l',l'') + t(l'')$, $\Lambda = \Lambda + \lambda(l'')$,

$L = L \cup \{l''\}$ and set $l' = l''$.

Output fastest path, containing the links in L with travel time T and length Λ. Terminate.

2.4.4 Optimality of the RA* algorithm

It is shown in Section 2.4.2 that the FA* algorithm is optimal in real road networks with turn restrictions, i.e. that the label $g(l_x)$ of each link part l_x is equal to the least travel time from the origin to l_x. While Chen et al (2006) conjecture that the RA* algorithm is also optimal this is not verified. This sub-section sets out to prove that the RA* algorithm is also optimal in real road networks, i.e. that $g(l_x)$ is equal to the least travel time from l_x to the destination in the unreversed network. The proof follows the propositions given in Section 2.4.2, however, this time with the direction of all links reversed and starting from the destination. The notation and definitions used so far also apply here.

Proposition 2.4-4: For any non-closed link part l_x and fastest path P_{min} from l_x to d_s, there is an open link part $i_y \in P_{min}$ with $g(i_y) = t_{min}(i_y,d_s)$.

Proof: If d_s is open (first iteration of the algorithm is not yet complete) then $i_y = d_s$ and $g(d_s) = t_{min}(d_s,d_s) = 0$ and Proposition 2.4-4 is trivially true. Suppose d_s is closed. Let CL^* be the set of closed link parts on P_{min} for which $g(i_y) = t_{min}(i_y,d_s)$ if $i_y \in CL^*$. CL^* is not empty, as $d_s \in CL^*$. Choose $l_z \in CL^*$ with the largest $t_{min}(l_z,d_s)$ and call this l_z^*. Note that $l_z^* \neq l_x$, as l_x is non-closed. Let $i_y \in P_{min}$ be the predecessor to l_z^* (possibly $i_y = l_z$). By Step 2 of the algorithm it is known that $g(i_y) \leq t_{min}(l_z^*,d_s) + t(i_y,l_z^*)$. But $t_{min}(i_y,d_s) = t_{min}(l_z^*,d_s) + t(i_y,l_z^*)$ as $i_y \in P_{min}$. Hence $g(i_y) \leq t_{min}(i_y,d_s)$. Since by definition $g(i_y)$ is the lowest cost encountered so far, $g(i_y) \geq t_{min}(i_y,d_s)$. Thus $g(i_y) = t_{min}(i_y,d_s)$. Moreover, $i_y \notin CL^*$ as $t_{min}(i_y,d_s) > t_{min}(l_z^*,d_s)$ since $t(i_y,l_z^*) > 0$. So i_y must be open, otherwise $i_y \in CL^*$. QED

Choose a heuristic $h(l_x) \geq 0$ such that $h(l_x) \leq t_{min}(o_e,l_x)$ for all $l \in V$. This implies that $h(o_e) = 0$. The heuristics thereby represent lower bounds for the cost from o_e to l_x.

Proposition 2.4-5: Suppose that RA* has not terminated and that P_{min} is the least travel time path from o_e to d_s. There is an open link part $i_x \in P_{min}$ such that $f(i_x) \leq t_{min}(o_e,d_s)$.

Proof: By Proposition 2.4-4 there is an open node $i_x \in P_{min}$ with $g(i_x) = t_{min}(i_x,d_s)$. As the algorithm has not terminated this could be o_e. So $f(i_x) = g(i_x) + h(i_x) = t_{min}(i_x,d_s) + h(i_x) \leq t_{min}(i_x,d_s) + t_{min}(o_e,i_x)$. As $i_x \in P_{min}$, $t_{min}(i_x,d_s) + t_{min}(o_e,i_x) = t_{min}(o_e,d_s)$. QED

Proposition 2.4-6 : The least travel time path is found by RA*.

Proof: Assume otherwise. There are two possibilities, the algorithm fails to terminate (Case 1) or it terminates without finding the least cost path (Case 2).

Case 1: Without the re-visiting of closed link parts, the algorithm terminates after at most $|2V|$ iterations, where $|V|$ is the number of links, as one open link part is closed at each iteration (Step 1). No link part further in cost terms than $t_{min}(o_e,d_s)$ from o_e will be expanded, as by Proposition 2.4-5 there exists an open link part i_x within a travel time of $t_{min}(o_e,d_s)$ from o_e directly before termination and this will be expanded instead. A closed link part i_x is only re-opened when $g(i_x)$ is reduced and adding a loop to the original path will not do this. No optimal path will contain loops. There are a finite number of non-looping paths between any pair of links. Hence each part can only be closed (and therefore opened) a finite number of times, so the algorithm must terminate.

Case 2: Suppose the algorithm terminates with $f(o_e) = g(o_e) > t_{min}(o_e,d_s)$. By Proposition 2.4-5, there exists just before termination at o_e an open link part i_x on the least travel time path such that $f(i_x) \leq t_{min}(o_e,d_s) < f(o_e)$. Thus i_x would have been selected for expansion instead of o_e, contradicting the assumption that the algorithm terminated. QED

It is therefore concluded that, similarly to FA*, the RA* is optimal.

2.5 Calculation of time-dependent travel times

The previous section presented the FA* and RA* algorithms, specifically adapted for real road networks, and proved that whether forward or reverse, the A* algorithm is optimal. This section deals with time-dependence and how to incorporate it in A*; this is achieved using the method of Sung et al (2000), called ‘flow speed model’, introduced briefly in Section 2.2.3. Its main advantage is the fact that it can retain the precision of label-setting and label-correcting path finding algorithms in time-dependent networks, while at the same time ensuring that consistency holds. This is attained by using link speed profiles rather than travel time profiles.

In the next two sub-sections the forward and reverse versions of Sung's flow speed model, intended for the forward and reverse A* algorithms of an in-vehicle navigation strategy, are presented. It should be noted that the procedures described aim at computing the travel time $t(l)$, experienced by a vehicle entering link l at a particular time of the day, and the delay $\delta(a,b)$, experienced by a vehicle following junction movement (a,b), which are then input into the algorithms of Section 2.4.

The notation used in the previous section also applies here, such that the same network G as in Section 2.4 is considered here. If Z is a set of time intervals $[\tau_k, \tau_{k+1})$, $k = 0, 1, \ldots, m-1$ separated by the time points $\tau_0 < \tau_1 < \tau_2 < \ldots < \tau_{m-1} < \tau_m$ and covering one day, then for every time interval $\Delta\tau_k : [\tau_k, \tau_{k+1})$, every link l has a speed value $\upsilon_k(l)$ and every movement (a,b) has a delay value of $\delta_k(a,b)$.

2.5.1 Forward time-dependent travel time calculation

Using Sung's flow speed model, time-dependent link travel time and movement delay values can be calculated, taking into account the departure time from each point of the FA* search. Basically, a vehicle entering link l, of length $\lambda(l)$ at time $\tau_{ent}(l)$, where $\tau_{ent}(l) \in Z$ and $\tau_0 \leq \tau_{ent}(l) < \tau_1$, will exit link l at time $\tau_{ex}(l) \in Z$, such that $\tau_{ex}(l) - \tau_{ent}(l) = t(l)$, which is the travel time of link l. The time of exit of the vehicle from link l, $\tau_{ex}(l)$, is calculated as follows:

	$\tau_{ex}(l) = \tau_{ent}(l) + \lambda(l) / \upsilon_0(l)$	if	$\lambda(l) / \upsilon_0(l) < \tau_1 - \tau_{ent}(l)$
else	$\tau_{ex}(l) = \tau_1 + (\lambda(l) - \lambda_0) / \upsilon_1(l)$	if	$(\lambda(l) - \lambda_0) / \upsilon_1(l) < \tau_2 - \tau_1$
else	$\tau_{ex}(l) = \tau_2 + (\lambda(l) - \lambda_1) / \upsilon_2(l)$	if	$(\lambda(l) - \lambda_1) / \upsilon_2(l) < \tau_3 - \tau_2$
:			
:			
else	$\tau_{ex}(l) = \tau_k + (\lambda(l) - \lambda_{k-1}) / \upsilon_k(l)$	if	$(\lambda(l) - \lambda_{k-1}) / \upsilon_k(l) < \tau_{k+1} - \tau_k$
:			
:			

else $\tau_{ex}(l) = \tau_{m-1} + (\lambda(l) - \lambda_{m-2}) / \upsilon_{m-1}(l)$ if $(\lambda(l) - \lambda_{m-2}) / \upsilon_{m-1}(l) < \tau_m - \tau_{m-1}$

where:

$\lambda_0 = \upsilon_0(l)\,(\tau_1 - \tau_{ent}(l))$

$\lambda_1 = \lambda_0 + \upsilon_1(l)\,(\tau_2 - \tau_1)$

:

:

$\lambda_{k-1} = \lambda_{k-2} + \upsilon_{k-1}(l)\,(\tau_k - \tau_{k-1})$

:

:

$\lambda_{m-2} = \lambda_{m-3} + \upsilon_{m-2}(l)\,(\tau_{m-1} - \tau_{m-2})$

The above procedure is intended to be run whenever a link's travel time is required in the algorithm, since computing values in advance is meaningless, as these will go out of date very soon. At each step of FA*, the entry time to the link is known, as it can be simply obtained from adding the $g(l_s)$ label of link l to the departure time from the origin. A visualisation of the method is shown in Figure 2.5-1.

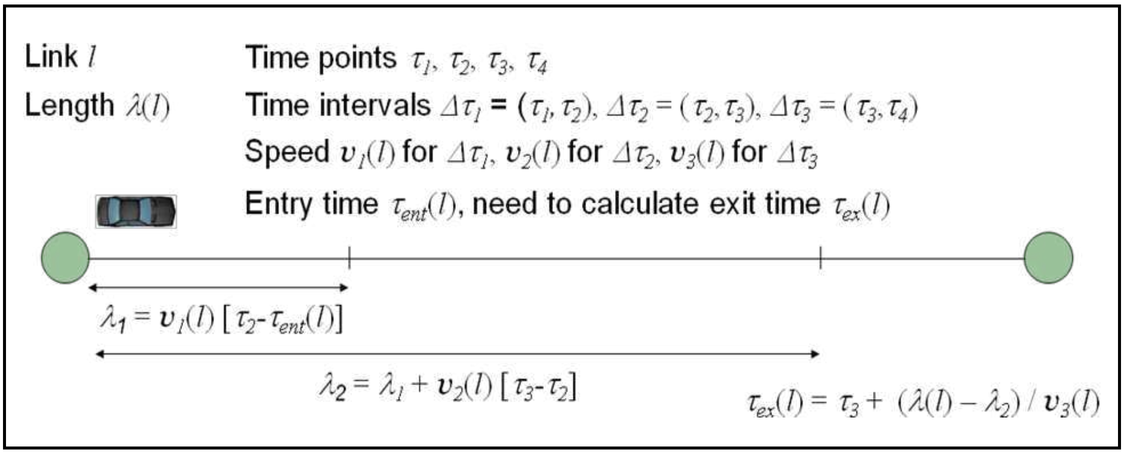


Figure 2.5-1: Forward time-dependent travel time calculation

When it comes to calculating the time-dependent delay $\delta(a,b)$ of a movement (a,b), the above technique cannot explicitly be applied, because movements do not have length and speed values; instead, a delay value is given for each time interval. Thus, the delay value $\delta_k(a,b)$ for each time interval $\Delta\tau_k : [\tau_{k-1}, \tau_k)$ is converted to a fictional speed value $\upsilon_k(a,b)$, such that $\upsilon_k(a,b) = \lambda(a,b) / \delta_k(a,b)$, where $\lambda(a,b)$ is a fictional movement length value. Hence, a vehicle exiting link a (entering movement (a,b)) at time $\tau_{ex}(a) \in Z$ will enter link b (exit movement (a,b)) at time $\tau_{ent}(b) \in Z$, such that $\tau_{ent}(b) - \tau_{ex}(a) = \delta(a,b)$, which is the delay of movement (a,b). As $\tau_{ex}(a)$ is known from the $g(a_e)$ label in the FA* algorithm, the procedure aims at calculating $\tau_{ent}(b)$.

2.5.2 Reverse time-dependent travel time calculation

For the RA* algorithm, the time-dependent travel time calculation is carried out simply by inverting the above procedure and attempting to derive the entry time $\tau_{ent}(l)$ into link l from a known exit time $\tau_{ex}(l)$ from it, where $\tau_{ex}(l) \in Z$ and $\tau_{m-1} \leq \tau_{ex}(l) < \tau_m$. This is done as follows:

	$\tau_{ent}(l) = \tau_{ex}(l) - \lambda(l) / \upsilon_{m-1}(l)$	if	$\lambda(l) / \upsilon_{m-1}(l) < \tau_{ex}(l) - \tau_{m-1}$
else	$\tau_{ent}(l) = \tau_{m-1} - (\lambda(l) - \lambda_{m-1}) / \upsilon_{m-2}(l)$	if	$(\lambda(l) - \lambda_{m-1}) / \upsilon_{m-2}(l) < \tau_{m-1} - \tau_{m-2}$
else	$\tau_{ent}(l) = \tau_{m-2} - (\lambda(l) - \lambda_{m-2}) / \upsilon_{m-3}(l)$	if	$(\lambda(l) - \lambda_{m-2}) / \upsilon_{m-3}(l) < \tau_{m-2} - \tau_{m-3}$
:			
:			
else	$\tau_{ent}(l) = \tau_{k+1} - (\lambda(l) - \lambda_{k+1}) / \upsilon_k(l)$	if	$(\lambda(l) - \lambda_{k+1}) / \upsilon_k(l) < \tau_{k+1} - \tau_k$
:			
:			
else	$\tau_{ent}(l) = \tau_2 - (\lambda(l) - \lambda_2) / \upsilon_1(l)$	if	$(\lambda(l) - \lambda_2) / \upsilon_1(l) < \tau_2 - \tau_1$
else	$\tau_{ent}(l) = \tau_1 - (\lambda(l) - \lambda_1) / \upsilon_0(l)$	if	$(\lambda(l) - \lambda_1) / \upsilon_0(l) < \tau_1 - \tau_0$

where:

$\lambda_{m-1} = \upsilon_{m-1}(l)\,(\tau_{ex}(l) - \tau_{m-1})$

$\lambda_{m-2} = \lambda_{m-1} + \upsilon_{m-2}(l)\,(\tau_{m-1} - \tau_{m-2})$

:

:

$\lambda_{k+1} = \lambda_{k+2} + \upsilon_{k+1}(l)\,(\tau_{k+2} - \tau_{k+1})$

:

:

$\lambda_2 = \lambda_3 + \upsilon_2(l)\,(\tau_3 - \tau_2)$

$\lambda_1 = \lambda_2 + \upsilon_1(l)\,(\tau_2 - \tau_1)$

At each step of RA*, the exit time from a link is known, as it can be simply obtained from subtracting the $g(l_e)$ label of link l from a given (or desired) arrival time at the destination; how the latter is obtained is described in more detail in Chapter 4. A visualisation of the method is shown in Figure 2.5-2.

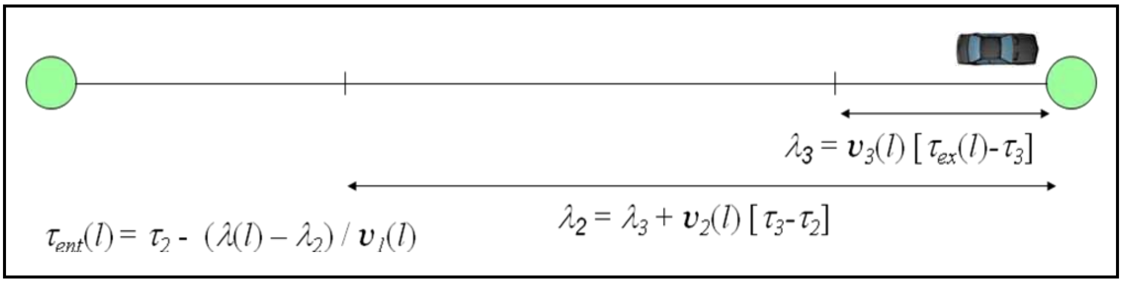


Figure 2.5-2: Reverse time-dependent travel time calculation

In the same way as in the forward time-dependent travel time calculation, the delay value $\delta_k(a,b)$ for each time interval $\Delta\tau_k : [\,\tau_{k-1}, \tau_k)$ is converted to a fictional speed value $\upsilon_k(a,b)$, such that $\upsilon_k(a,b) = \lambda(a,b) / \delta_k(a,b)$, where $\lambda(a,b)$ is a fictional movement length value. Hence, a vehicle entering link b (exiting movement (a,b)) at time $\tau_{ent}(b) \in Z$ will have exited link a (entered movement (a,b)) at time $\tau_{ex}(a) \in Z$, such that $\tau_{ent}(b) - \tau_{ex}(a) = \delta(a,b)$, which is the delay of movement (a,b). As $\tau_{ent}(b)$ is known from the $g(b_s)$ label in the RA* algorithm, the procedure aims at calculating $\tau_{ex}(a)$.

It should be noted here, that RA* using the above procedure as a solution to find the time-dependent shortest path, is optimal, since, as derived by Daganzo (2002), any algorithm that solves the forward time-dependent shortest path problem is shown also to solve backward problems of a conjugate type, i.e. with the link directions reversed.

2.5.3 Numerical examples

In order to demonstrate the methods of forward and reverse time-dependent travel time calculations, two numerical examples are given here.

Example 1: Forward time-dependent travel time calculation

Consider link l of length $\lambda(\mathrm{l})$ = 2500 m. Also, consider the time points τ_0 = 0 s, τ_1 = 300 s, τ_2 = 600 s and τ_3 = 900 s, specifying three 5-minute time intervals $\Delta\tau_0 = [\tau_0, \tau_1)$, $\Delta\tau_1 = [\tau_1, \tau_2)$ and $\Delta\tau_2 = [\tau_2, \tau_3)$. The speed on the link for each time interval is $v_0(\mathrm{l})$ = 55 km/h (15.27 m/s), $v_1(\mathrm{l})$ = 10 km/h (2.78 m/s) and $v_2(\mathrm{l})$ = 45 km/h (12.5 m/s). Setting an entry time of a vehicle into the link at $\tau_{\mathrm{ent}}(\mathrm{l})$ = 211 s, the travel time t(l) on the link is to be calculated.

First, the values of $\lambda(\mathrm{l})$ / $v_0(\mathrm{l})$ and $\tau_1 - \tau_{\mathrm{ent}}(\mathrm{l})$ are calculated:

$$\lambda(\mathrm{l}) / v_0(\mathrm{l}) = 2500 / 15.27 = 163.72 \text{ s}$$
$$\tau_1 - \tau_{\mathrm{ent}}(\mathrm{l}) = 300 - 211 = 89 \text{ s}$$

Since $\lambda(\mathrm{l}) / v_0(\mathrm{l}) > \tau_1 - \tau_{\mathrm{ent}}(\mathrm{l})$, the procedure goes into the next iteration and the value of λ_0 needs to be computed:

$$\lambda_0 = v_0(\mathrm{l})\,(\tau_1 - \tau_{\mathrm{ent}}(\mathrm{l})) = 15.27 \cdot 89 = 1359.03 \text{ m}$$

The values of $(\lambda(\mathrm{l}) - \lambda_0) / v_1(\mathrm{l})$ and $\tau_2 - \tau_1$ are calculated next:

$$(\lambda(\mathrm{l}) - \lambda_0) / v_1(\mathrm{l}) = (2500 - 1359.03) / 2.78 = 410.42 \text{ s}$$
$$\tau_2 - \tau_1 = 600 - 300 = 300 \text{ s}$$

Since $(\lambda(\mathrm{l}) - \lambda_0) / v_1(\mathrm{l}) > \tau_2 - \tau_1$, the procedure goes into the next iteration and the value of λ_1 needs to be computed:

$$\lambda_1 = \lambda_0 + v_0(\mathrm{l})\,(\tau_2 - \tau_1) = 1359.03 + 2.78 \cdot 300 = 2193.03 \text{ m}$$

The values of $(\lambda(\mathrm{l}) - \lambda_1) / v_2(\mathrm{l})$ and $\tau_3 - \tau_2$ are calculated next:

$$(\lambda(\mathrm{l}) - \lambda_1) / v_2(\mathrm{l}) = (2500 - 2193.03) / 12.5 = 24.56 \text{ s}$$
$$\tau_3 - \tau_2 = 900 - 600 = 300 \text{ s}$$

Since $(\lambda(l) - \lambda_1) / \upsilon_2(l) < \tau_3 - \tau_2$, the procedure stops and the exit time from the link is computed, using the appropriate formula:

$$\tau_{ex}(l) = \tau_2 + (\lambda(l) - \lambda_1) / \upsilon_2(l) = 600 + 24.56 = 624.56 \text{ s}$$

Hence, the travel time experienced by the vehicle on the link is:

$$t(l) = \tau_{ex}(l) - \tau_{ent}(l) = 624.56 - 211 = 413.56 \text{ s} = 6.89 \text{ min}$$

Example 2: Reverse time-dependent travel time calculation

In the same link and for the same time period as in Example 1, the travel time $t(l)$ of the link is to be calculated, this time though using the reverse procedure, starting from the exit time $\tau_{ex}(l)$ = 624.56 s and working out the entry time $\tau_{ent}(l)$.

First, the values of $\lambda(l) / \upsilon_2(l)$ and $\tau_{ex}(l) - \tau_2$ are calculated:

$$\lambda(l) / \upsilon_2(l) = 2500 / 12.5 = 200 \text{ s}$$
$$\tau_{ex}(l) - \tau_2 = 624.56 - 600 = 24.56 \text{ s}$$

Since $\lambda(l) / \upsilon_2(l) > \tau_{ex}(l) - \tau_2$, the procedure goes into the next iteration and the value of λ_2 needs to be computed:

$$\lambda_2 = \upsilon_2(l) \, (\tau_{ex}(l) - \tau_2) = 12.5 \cdot 24.56 = 307 \text{ m}$$

The values of $(\lambda(l) - \lambda_2) / \upsilon_1(l)$ and $\tau_2 - \tau_1$ are calculated next:

$$(\lambda(l) - \lambda_2) / \upsilon_1(l) = (2500 - 307) / 2.78 = 788.85 \text{ s}$$
$$\tau_2 - \tau_1 = 600 - 300 = 300 \text{ s}$$

Since $(\lambda(l) - \lambda_2) / \upsilon_1(l) > \tau_2 - \tau_1$, the procedure goes into the next iteration and the value of

λ_1 needs to be computed:

$$\lambda_1 = \lambda_2 + \upsilon_1(l)\ (\tau_2 - \tau_1) = 307 + 2.78 \cdot 300 = 1141 \text{ m}$$

The values of $(\lambda(l) - \lambda_1) / \upsilon_0(l)$ and $\tau_1 - \tau_0$ are calculated next:

$$(\lambda(l) - \lambda_1) / \upsilon_0(l) = (2500 - 1141) / 12.5 = 89 \text{ s}$$
$$\tau_1 - \tau_0 = 300 - 0 = 300 \text{ s}$$

Since $(\lambda(l) - \lambda_1) / \upsilon_0(l) < \tau_2 - \tau_1$, the procedure stops and the entry time into the link is computed, using the appropriate formula:

$$\tau_{ent}(l) = \tau_1 - (\lambda(l) - \lambda_1) / \upsilon_0(l) = 300 - 89 = 211 \text{ s}$$

The result matches the entry time in Example 1, which shows that the reverse procedure is correct.

2.6 Concluding remarks

In this chapter the topic of route finding in road networks was examined. Initially, a comprehensive review of path finding algorithms, including shortest path algorithms and time-dependent shortest paths, was given. Then, the features of real road networks and how these are modelled so as to be considered in path finding algorithms were described.

It was concluded from the literature review, that the most suitable algorithm for the path finding routine in dynamic navigation is A*, as it has many advantages compared to other algorithms in terms of both efficiency and precision. A section reporting on the main characteristics of the A* algorithm and presenting a mathematical formulation of its forward version (FA*) specifically for road networks with features such as turn restrictions was given. A formulation of the RA* algorithm, that is one that is searching from the destination to the origin was also developed, intended to be executed in conjunction with the forward version so as to take advantage of pre-computed information when multiple runs of the A* are required. Optimality proofs for both FA* and RA* were derived.

Finally, the formulation of a method for incorporating time-dependence in both FA* and RA* was presented, whose main characteristic was that time-dependent travel times were computed without violating consistency. Numerical examples demonstrating the applicability of the approach were also given.

In spite of the advantages that the FA* and RA* algorithms have, there are also some weaknesses that need to be mentioned here. A weakness relating to the original A* algorithm, is the heuristic used when estimating the travel time from any point of the network to the destination, in the case of FA*, or the origin, in the case of RA*. An estimate based on the Euclidian distance is guaranteed to underestimate the actual travel time ensuring optimality of A*, however it may be greatly underestimating it, resulting thus in longer running times. There may be other heuristics, which may still guarantee optimality but also provide closer estimates, such that the running time can be reduced even further, or even heuristics that slightly overestimate the actual distance thus yielding routes that remain precise without being optimal any more, but which provide great savings in terms of computation time. This is a field that deserves further attention; nonetheless, this is beyond the scope of this study.

Another weakness can be identified in the method developed for modelling real road networks, and more specifically in the positioning, where it is assumed that the exact position of the vehicle can be extrapolated to a position located anywhere on the link. While exact positioning is required when it comes to implementing the FA* and RA* algorithms in an on-board navigation system, it requires a fairly significant amount of work, including processes such as map matching, which is not required at this stage and is therefore beyond the scope of this study, which concentrates on the actual in-vehicle navigation methodology. Nonetheless, it is a very important task that needs to be undertaken when these algorithms are to be run in a real navigation system.

Having described the path finding algorithms used in the in-vehicle navigation strategy presented in this study, the next chapter goes on to define reliability, based on the distribution of travel times on links and routes.

CHAPTER 3

Travel time uncertainty, variability and reliability

3.1 Introduction

The fact that travel time is not constant, but entails an element of variability, resulting in uncertainty when attempting to predict it, has been recognised in the literature for a long time. Already in a very early study by Wardrop (1952) it is noted that journey travel times follow a skewed distribution with a long 'tail' representing the few very slow vehicles, such that it is very likely for the mean travel time to be exceeded. In a later study by Thomson (1968), travel time variability is identified as an important characteristic of a road network and it is pointed out that the unpredictability of travel time is one of the most important sources of time losses.

More recently however, the term reliability is adopted in transport; originating from computer science (IEEE, 1990), it is introduced so as to express the quality of operation of a transport mode of any kind (road, rail etc). The terms variability and reliability are very closely related and have thus been the objective of many research studies.

In a study by Lomax et al (2003), definitions for both reliability and variability are provided.

According to this study, reliability is defined as being “commonly used in reference to the level of consistency in transport service for a mode, trip, route or corridor for a time period”. It is also noted, that reliability is viewed by travellers in relation to their experience. Variability on the other hand “might be thought of as the amount of inconsistency in operating conditions”. In contrast to reliability, variability is viewed rather from a “facility perspective, and, therefore, relates more to the concerns of transportation agencies”.

A rough differentiation that can thus be performed on the two terms could be that, while the term variability describes the phenomenon of inconsistency in travel time, focusing on stochastic processes and distributions, the term reliability provides a measure of this phenomenon, concentrating on indices. In the case of in-vehicle navigation, which is the focus of this study, one is interested in considering the variability of travel time, which causes uncertainty in the travel time that is to be expected, expressed as the reliability of various network elements, such as roads, or even of the entire road network.

The next section gives a literature review of the topic of travel time variability. Variability is first identified and analysed, then ways of expressing it as measures of reliability are described. This builds the foundation of a new reliability measure, developed and introduced in the following section, which is intended to be used in in-vehicle navigation in the following chapters.

3.2 Literature review

A literature review on travel time variability, uncertainty and reliability is presented in this section, so as to provide the background to the proposed reliability measure, described in the next section. At first, the importance of travel time uncertainty and reliability to travellers is identified, followed by a description of the causes and components of variability. Studies investigating the distribution of travel times and studies attempting to model travel time variability and reliability are reviewed next. These are followed by a discussion on measures of travel time reliability adopted by several studies in the past. Finally, an appraisal of methods developed in the literature, aiming to compute the reliability of a path from the reliability of individual links is given, along with a summary of the review conducted here.

3.2.1 The importance of travel time uncertainty and reliability

The importance of travel time uncertainty has been the objective of many research studies in the past and has therefore been extensively analysed from the traveller's perspective. Many studies have concluded that although travel time is an important factor affecting the traveller's route choice behaviour, travel time variability can be even more important. Travellers are interested in how long it will take them to reach their destination, but are even more concerned with the reliability of their prediction of total travel time. A wrong travel time prediction results in either an early arrival at the destination or in a delay. None of these situations is appreciated by the traveller, with delays usually having more severe consequences for him/her (e.g. late arrival at the workplace) and therefore not being tolerated. Hence, much research has focused on developing methods for modelling travel time reliability.

In an empirical study, Jackson and Jucker (1981) prove that from a travel time distribution, both the mean and the variance are important when a traveller chooses a route (or even transport mode) to work. They then develop a mean-variance model to reflect this and they define a so-called "risk-aversion" parameter, which quantifies an individual traveller's trade-off between time and variability.

In a further empirical study, Abdel-Aty et al (1995) conduct a computer-aided telephone survey, where respondents give their choices from a series of hypothetical binary choice sets regarding their travel. The results show that, not only the reliability of travel time is important to a traveller's route choice, but also that this can be influenced by traffic information, since this is a way to reduce travel time uncertainty.

Noland and Small (1995) identify that many travellers adopt a safety margin, termed "head start", during their morning commute, so as to take travel time uncertainty into account as much as possible and to avoid late arrival at the workplace. The study concludes that uncertainty is more "expensive" for travellers than actual travel time and that it would thus be more profitable to introduce measures aiming to reduce uncertainty rather than travel time.

An empirical study by Lam and Small (2001) investigates the value of travel time and reliability, where reliability is defined as the difference between the 90^{th} percentile travel time value and the median of the travel time distribution (see also Section 3.2.4). The actual behaviour of commuters faced with a choice between a free and a tolled road is monitored and route choice models are developed, combined with other choices such as time of day and car occupancy. The results show that reliability is an important factor affecting route choice; however, women tend to be more risk-averse and prefer thus a reliable alternative to a fast one.

Bates et al (2002) formulate the perception of travel time uncertainty by travellers as an additional "schedule disutility" function in their journey time valuation function and estimate it to be approximately equal to a scalar multiplier of the standard deviation of the travel time distribution. Liu et al (2004) formulate travellers' route choice as a mixed logit model, containing coefficients representing individual travellers' preferences towards travel time, reliability and cost. As opposed to previous empirical studies, real-time data from loop detectors is used in this study and a genetic algorithm is subsequently implemented to identify the parameters that result in the best match between the route choice model and the observed traffic volumes.

More studies, aiming to model travel time reliability exist in the literature; comprehensive reviews of this topic have been carried out by Bates et al (2001) and Noland and Polak (2002). Having thus demonstrated the importance of travel time uncertainty and reliability, the next section describes their causes and components.

3.2.2 The causes and components of travel time variability

This section firstly identifies why travel time varies; then, travel time variability is disaggregated and the components forming it are described.

The causes of travel time variability

Robinson (2005) identifies that the factors affecting travel time, and hence travel time variability, can be classified into the following four key generic drivers:

- Demand
- Capacity
- Vehicle performance and

- Control.

The first driver is related proportionally to travel time, such that an increase in demand causes the travel time to also increase. Examples of this include commuting (increased demand at certain times of the day), a sports event (increased demand in the proximity of a stadium at a certain time and day), availability of vehicle storage facilities, such as parking spaces (fewer parking spaces at a location increase demand, as drivers are forced to drive around for some time seeking a space), as well as more long-term effects, such as long-term trends of traffic growth.

Conversely, the second driver is inversely proportional to travel time, such that a decrease in capacity causes it to increase. Examples of that include incidents (accidents or road works may partially or completely block transport links, thus reducing their capacity), adverse weather (bad weather conditions may reduce visibility and safety on the road, thus forcing the vehicles to drive slowly), parked vehicles (parked vehicles may partially block the road or reduce visibility, thus forcing drivers to drive slowly or even halt at certain locations) etc.

The third driver (the performance of an individual vehicle) may affect travel time both positively or negatively. Examples include vehicle speed and driving speed (a vehicle has maximum speed at which it can travel, but it is up to the driver to choose the speed at he/she wishes to drive), risk acceptance (some travellers tend to drive riskier than others, e.g. they accept smaller gaps in the opposing traffic flow while executing a right turn), human factors (e.g. the mood of a driver affecting his/her driving behaviour) etc.

Finally, the fourth driver (traffic control) may also affect travel time both positively and negatively. Examples include speed limits (the travel time on a long link depends on the imposed speed limit, as the majority of the drivers will use it to determine their driving speed), on-line and off-line traffic signals (traffic signals control the number of vehicles travelling on a link, thus affecting capacity), tolls (setting a toll for a specific road reduces the demand) etc.

The components of travel time variability

Several studies have attempted to disaggregate travel time variability into components. For example, in a study by Bates et al (2002), travel time variability is defined as an unpredictable variation in travel times, consisting of two components: day-to-day variability and incident-related variability. In another study by Wong and Sussman (1973), three components are identified, depending on how predictable the source of variability is: regular condition-dependent variations (predictable travel time variations, resulting from the difference in rush hour and slack hour traffic, summer and winter conditions, day of the week etc), irregular condition-dependent variations (unpredictable abnormal network conditions, such as bad weather, accidents etc) and random variations (short term unpredictable variations, not affecting all vehicles on the same route, and being the reason why two vehicles departing at the same time from a given origin to a destination on the same route will most likely experience different travel times).

However, the most commonly used way of disaggregating travel time variability is by temporal scale, such that each individual component refers to the variability due to the differences between time periods (Montgomery and May, 1987; Park et al, 1999; Robinson, 2005; Robinson and Polak, 2007). Namely, one can distinguish between the following components of variability:

- Vehicle-to-vehicle
- Period-to-period
- Day-to-day
- Season-to-season

Vehicle-to-vehicle travel time variability is the short-term unpredictable travel time variation experienced by different vehicles on the same link at the same time and corresponds to the random variation defined by Wong and Sussman (1973) in the work mentioned earlier in this section. Period-to-period variability on the other hand is the variation in travel time that occurs over the course of a single day. Day-to-day variability represents the difference in travel time over the same time period but between different days, while season-to-season variability can be observed in the long run, reflecting the difference in travel time on a road during different times of the year, e.g. summer and winter.

In in-vehicle navigation applications, all four components of variability are important and should be taken into consideration. The next section reviews previous research work on the determination of the probability distribution, which best reflects the distribution of travel times.

3.2.3 The distribution of travel times

The distribution of travel times has been extensively investigated in the last decades, as it is a very important topic when it comes to modelling travel time variability and reliability. A large number of research studies are available in the literature, attempting to fit the distribution of travel times to one of the continuous probability distributions.

Wardrop (1952) first identifies that travel times follow a skewed distribution. This is subsequently verified by Herman and Lam (1974), through an empirical study on travel time variability. The study concludes that travel times indeed follow a skewed distribution and that only the 60% lower values fit a normal distribution well, contradicting thus the assumption made at the time, that travel times are normally distributed. In order to fit the actual distribution of travel times, the log-normal or gamma distribution is suggested.

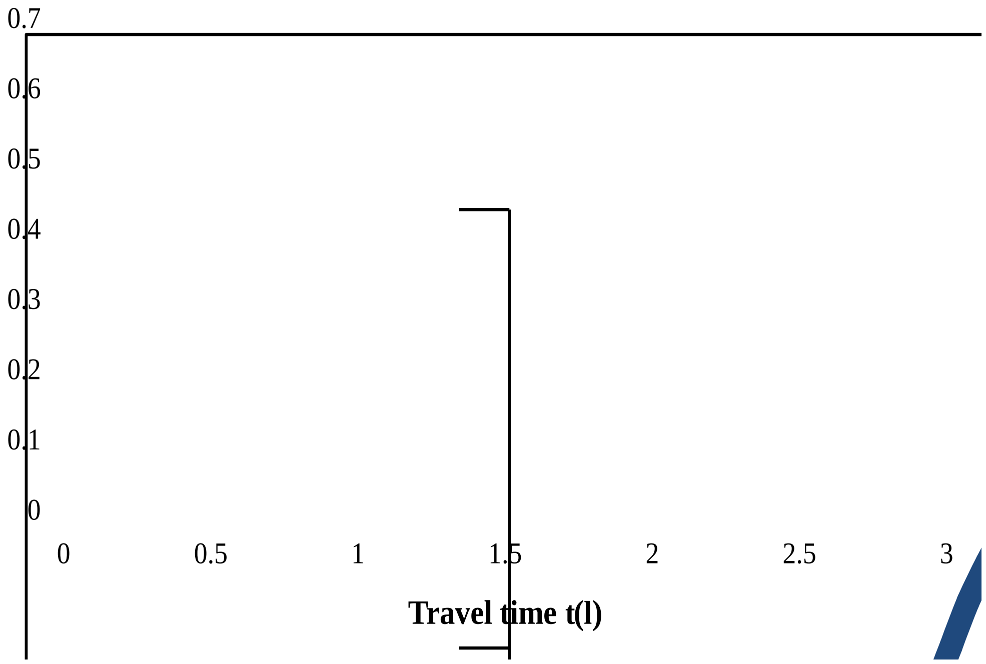


Figure 3.2-1: Log-normal probability density function

In later studies, the above finding is further confirmed. Namely, in a small empirical study by Polus (1979), it is concluded that travel times best fit a gamma distribution. A log-normal or gamma distribution is suggested by Dandy and McBean (1984), while studies by Mogridge and

Fry (1984), Montgomery and May (1987), Rakha et al (2006) and Chen et al (2007) also derive a log-normal fit.

Despite more recent studies assuming a normal distribution for travel times for reasons of simplicity (Lomax et al, 2003), the skewness of the distribution remains an important characteristic that needs to be considered when analysing travel time variability and reliability, and hence a gamma or log-normal distribution would be more appropriate. Hence, it is assumed in this study that travel times are log-normally distributed; this is based on the widely accepted assumption that speeds are normally distributed, which is also empirically confirmed in Appendix A. The assumption is used in Section 3.3, where a new measure of reliability is defined. An example of a log-normal probability density function plot is shown in Figure 3.2-1.

3.2.4 Modelling travel time reliability

Several studies, whose objective it is to model travel time variability, and hence reliability, have been carried out in the past. Most of them are empirical and are presented in the next paragraphs.

One of the first empirical studies by Smeed and Jeffcoate (1971) attempts to model day-to-day travel time variability, by recording the travel time of a particular commuter route every day for a period of time, for various departure times. Due to the fact that only limited data is used in the study, it is not possible to measure period-to-period or vehicle-to-vehicle variability; however, an interesting finding that is presented is that the travel time is higher on Monday than it is on other days, and that there is more variability in the morning peak than in the evening peak. This is subsequently confirmed by Mogridge and Fry (1984), who measure travel time variability over a route from South London to Central London.

Herman and Lam (1974), in the same empirical study in which they investigate the distribution of travel times (Section 3.2.3) by studying the travel time on a number of different commuter routes into a common workplace in Detroit, find that the variability is greater in the evening peak than in the morning peak. This result contradicts the result by Smeed and Jeffcoate (1971); although the reason for this may lie in the fact that not enough data is available, it still suggests that the characteristics of travel time variability differ from link to link and between different environments.

The study by Polus (1979), mentioned in the previous section, also makes use of data from arterial routes in Chicago to develop both a statistical model and a regression model to predict reliability. Reliability is defined as the inverse of the standard deviation of the travel time distribution. In a further study by Hendrickson and Planck (1984), the travel time variability, expressed as the flexibility in the traveller's departure time, of various transport modes in Pittsburgh is analysed and a logit model of simultaneous transport mode and departure time choice is estimated. It is found that the variability of commuting travel time is generally low in the study area; nonetheless, the reason for this may be again, the lack of sufficient data and the different features of travel time variability in different environments.

More recently, attempts to relate travel time variability with congestion have been made. Bates et al (2002) identify that variability is the result of variations in demand; namely, while demand is in general below capacity, there may be time periods where it is in excess of it, which is the prime explanation of the formation of queues. The study looks into the travel time distribution and identifies that the standard deviation increases in line with the mean, but when the mean reaches a turning point and starts decreasing, it continues to rise for some time before it starts dropping. This relationship between variability and congestion is empirically investigated in a study by Black et al (2004), where the travel times of journeys in Leeds and London are measured and compared. The result is the derivation of a relationship between the coefficient of variation, defined as the ratio of the standard deviation over the mean of the travel time distribution, and the so-called "congestion index", defined as the ratio of actual travel time over the free flow travel time.

Most of the empirical research on modelling reliability so far has been limited by lack of sufficient data. Nevertheless, advances in technologies for data collection, such as Automatic Number Plate Recognition (ANPR) or detectors (ILD – Inductive Loop Detectors, RTMS – Remote Traffic Microwave Sensors etc), have improved the accuracy of the data collected, such that more recent studies have been able to draw more accurate conclusions. Using larger datasets does not only provide better estimates when it comes to measuring the variability of travel times, but also enables the observation of the individual temporal components of travel time variability, i.e. vehicle-to-vehicle, day-to-day, period-to-period and season-to-season.

In a study by Sutch (2001), travel time data is collected from the M6 motorway in the UK using

ANPR cameras. It is thus concluded, that weekdays can be split in three classes in terms of travel time variability: Monday, Tuesday-Thursday and Friday. It is also discovered that more day-to-day variability is to be expected in the periods after the period with the highest mean travel time than in those before, i.e. that there is more variability in the recovery from the peak than in the build-up to the peak. In a study by Frith (2000), a similar categorisation of days is derived. Using six days of data from ANPR cameras on the A12 road in the UK, it is discovered that roads with a morning peak carry more traffic on Mondays, while roads with an evening peak carry more traffic on Fridays, and that day-to-day variability increases with increasing mean travel time. This result is supported by a further study by Porter (2001), who also makes use of ANPR data.

A number of studies have used data from detectors to model travel time reliability. Using data from ILD in Orange County in California, Oh and Chung (2006) calculate the day-to-day, period-to-period and vehicle-to-vehicle travel time variability for links and routes, and use it as a measure of travel reliability. Robinson (2005) and Robinson and Polak (2007) utilise ILD data from Central London and model day-to-day, period-to-period and vehicle-to-vehicle variability using travel time estimates from the so-called k-Nearest Neighbour method. Eleftheriadou and Cui (2007) estimate reliability, based on a travel time estimation model considering four factors (congestion, weather, work zones and incidents) and generating scenarios and the probability of occurrence of each scenario using RTMS data from Philadelphia.

An important source of data for the estimation and modelling of travel time reliability is floating car data. For example, such is the work of Chen et al (2007), who use the data logs from taxis so as to analyse the temporal distribution characteristics of the travel time reliability on the road network of Beijing, i.e. to measure day-to-day and period-to-period variability. Their main finding is, as in many other studies, that reliability is low during peak hours and high during off-peak hours.

Other methods for modelling travel time reliability have also been used in past research. For example, Noland et al (1998) use simulation to generate link travel time distributions for a network, for which they estimate travel time reliability. Li et al (2007) present a method for estimating travel time variability from speed measurements instead of travel time measurements. Pattanamekar et al (2003) and Clark and Watling (2005) develop analytical models for estimating the probability distribution of travel time in the light of normal day to day variations in the travel demand matrix of a road network. Finally, studies by Asakura et al (2001), Bell et al (1999) and Du and Nicholson (1997) determine reliability by performing sensitivity analysis.

Having reported on previous work on modelling travel time reliability, the next section presents the various measures of reliability that have been introduced by several researchers.

3.2.5 Existing measures of reliability

Besides modelling the reliability of travel times, a considerable amount of research has focused on defining adequate measures for quantifying this reliability. Most of them use the various characteristics of the travel time distribution, such that two types of measures can be identified: measures indicating the probability that a certain link is unusable and measures attempting to quantify the amount of congestion that may be encountered on a link.

The most widely used reliability measure so far is the one defined by Bell and Iida (1997), which is expressed as the probability of a link to be uncongested. Bell and Iida assume that the condition of traffic flow on a link is binary, such that the link's state of operation is either 'normal' or 'abnormal', and that the link is either congested or uncongested. They then consider that the reliability in a transport network has two aspects, connectivity reliability and travel time reliability, where the former indicates the probability that traffic can reach a given destination at all and the latter depicts the probability that traffic can reach a given destination within a given time.

While this measure is suitable for quantifying the reliability of a network, it has the disadvantage that it does not give any indication on the amount of congestion that may be encountered. Other measures developed along the same concept are similarly limited, such as the probability indicator measure developed by Chen et al (2007), defined as "the probability of travel rates less than a threshold value" (where the travel rate is defined as travel time per unit length), and the reliability measure adopted by Eleftheriadou and Cui (2007), defined as "the percent of trips that reach a destination over a designated facility within a given travel time (or equivalently, at a given travel speed or higher)".

Thus, a number of studies attempt to quantify the reliability of a link by using the travel time distribution directly. The first measure is adopted in the study by Polus (1979), where reliability is defined as the inverse of the standard deviation of the link's travel time distribution:

$$R = \frac{1}{\sigma} \tag{3.2-1}$$

The main disadvantage of this measure, however, is the fact that it is not dimensionless. This is also the case of measures developed in further studies, such as Dandy and McBean (1984), who use the 95th percentile travel time, and Lam and Small (2001), who quantify reliability as the difference between the 90th percentile and the median of the travel time distribution.

Some studies develop reliability measures, not only considering the width of the travel time distribution, but also its skewness. For instance, van Lint and van Zuylen (2005) propose two reliability metrics, based on the 10th, the 50th and the 90th percentiles of the travel time distribution. The metrics proposed are:

$$\lambda^{\text{skew}} = \frac{\text{T90} - \text{T50}}{\text{T50} - \text{T10}} \tag{3.2-2}$$

and

$$\lambda^{\text{var}} = \frac{\text{T90} - \text{T10}}{\text{T50}} \tag{3.2-3}$$

However, the most important contribution to defining reliability measures has been made by Lomax et al (2001; 2003), who present a series of measures of reliability and categorised them in three groups, as statistical range measures, buffer time measures and tardy trip indicators. Statistical range measures are defined as presenting information in a relatively "unprocessed format", meaning that they are mainly based on concepts only familiar to statisticians and generally not understandable to ordinary travellers. Examples of statistical range measures include:

- the travel time window, which is the interval defined by the mean travel time and a "plus or minus" relationship with the standard deviation
- the percent variation (or coefficient of variation), which is the ratio of the standard de-

viation over the mean travel time

- the variability index, which is the ratio of the peak to off-peak travel time, which in turn is defined as the difference in the upper and lower 95th percentiles

Buffer time measures, on the other hand, are defined as being intended to relate well to the way travellers make decisions, and are therefore understandable by them, in a way that they indicate how much extra time should a traveller allow for his/her journey, to account for uncertainty in the travel conditions. Examples of buffer time measures include:

- the buffer time, defined as the difference between the 95th percentile travel time and the mean travel time
- the buffer time index, which introduces the concept of travel rate (time per distance unit), defined as the ratio of the difference of the 95th percentile travel rate and the mean travel rate, over the mean travel rate
- the planning time index, defined as the upper end of the buffer time index (i.e. the 95th percentile travel time over the average travel time)

Ultimately, tardy trip indicators are defined as representing the unreliability impacts using the amount of late trips. Examples of tardy trip indicators include:

- the Florida reliability statistic, which indicates the proportion of unreliable trips, i.e. the percentage of trips with travel times greater than expected
- the on-time arrival, which indicates the proportion of trips with travel times less than 110% of the mean travel time
- the misery index, which expresses the average travel time for the 20% longest trips, such that a measure of “how bad are the worst trips?” can be obtained

Following the categorisation of the various reliability measures, Lomax et al (2001; 2003) recommend three measures as the most appropriate to use, one from each category. These are, the percent variation, the buffer time index and the misery index:

$$\text{Percent Variation} = \frac{\text{Standard Deviation}}{\text{Average Travel Time}} \times 100 \tag{3.2-4}$$

$$\text{Buffer Time Index} = \frac{95^{\%}\text{ Travel Rate} - \text{Average Travel Rate}}{\text{Average Travel Rate}} \times 100 \qquad (3.2\text{-}5)$$

$$\text{Misery Index} = \frac{\text{Av. Travel Rate for Worst 20\% Trips} - \text{Av. Travel Rate}}{\text{Av. Travel Rate}} \qquad (3.2\text{-}6)$$

3.2.6 Calculating path travel time reliability

Regardless of which reliability measure is used, a problem that arises is the calculation of the reliability of a path, consisting of a series of links. When it comes to probability measures, the reliability of a path is the probability that congestion will not be encountered on any of the links forming the path. Thus, when the reliability of a link is the probability of not encountering congestion on the link, the reliability of the path will be simply the product of the reliabilities of all the links forming the path, assuming statistical independence between them (though this is generally not the case in reality). Extending this approach, the reliability of a path set, representing the probability of encountering congestion on all paths forming a path set, can be calculated using the technique described by Ahmad (1982).

Nonetheless, when it comes to the other type of reliability measures, i.e. the ones attempting to quantify the amount of congestion that may be encountered and its possible consequences, it is more sensible to calculate the variance of the path travel time from the variances of the travel times of the individual links. The state-of-practice in the calculation of the variance of a path is to compute the sum of the variances of the individual links, such that, for the travel time t_p of path p, and for each of the m links forming it, $i \in L$ with travel time t_i:

$$\text{var}(t_p) = \sum_{i \in L} \text{var}(t_i) = \frac{\sum_{i \in L} \overline{t_i}^2}{m} - \overline{t_p}^2 = \frac{\sum_{i \in L} \overline{t_i}^2}{m} - \left(\sum_{i \in L} \overline{t_i}\right)^2 \qquad (3.2\text{-}7)$$

where $\overline{t_i}$ and $\overline{t_p}$ denote the mean travel times of link i and path p.

The weakness of this method is the fact that it assumes statistical independence between the travel times of the different links forming the path. As has been identified by several studies, such as Bates et al (2002), there is correlation between successive links on a path, particularly in urban areas, which results from phenomena such as ‘blocking back’. Consequently, as shown by Rakha et al (2006), equation 3.2-7 tends to significantly underestimate the path variance.

Another method for calculating the path variance is developed by Sherali et al (2006), who use the maximum and minimum travel time coefficients of variation (standard deviation over mean) (CV) of each link on the path to construct bounds on the path’s travel time CV. The study derives that:

$$CV_p^{max} \geq \left(\frac{\sum_{i \in L} var(t_i)}{\sum_{i \in L} \bar{t}_i^2} \right)^{1/2} \geq CV_p^{min}$$

and therefore,

$$var(t_p) = \frac{\bar{t}_p^2}{\sum_{i \in L} \bar{t}_i^2} \left(\frac{\sum_{i \in L} \bar{t}_i^2}{m} - \left(\sum_{i \in L} \bar{t}_i \right)^2 \right) \qquad (3.2\text{-}8)$$

In the study by Rakha et al (2006), three more methods of estimation of a path’s CV are proposed, namely, the mean of the CVs of the individual links (equation 3.2-9), the median CV of the individual links (equation 3.2-10) and the mid-point between the maximum and minimum CVs among all links (equation 3.2-11) are presented:

$$var(t_p) = \frac{\bar{t}_p^2}{m^2} \left(\sum_{i \in L} \frac{\sigma_i}{\bar{t}_i} \right)^2 = \frac{\bar{t}_p^2}{m^2} \left(\sum_{i \in L} \frac{var(t_i)}{\bar{t}_i} \right)^2 \qquad (3.2\text{-}9)$$

$$var(t_p) = \left(\bar{t}_p \cdot Med_i(CV_i) \right)^2 \qquad (3.2\text{-}10)$$

$$var(t_p) = \frac{\bar{t}_p^2}{4} \left(max_{i \in L}(CV_i) - min_{i \in L}(CV_i) \right) \qquad (3.2\text{-}11)$$

Rakha et al (2006) also empirically compare the methods expressed by equations 3.2-7 to 3.2-11 and find that, while equation 3.2-7 underestimates the actual path travel time variance, the other equations overestimate it with errors of different scales, according to the time of the day in question. The most accurate and at the same time consistent estimates are given by equation 3.2-9, which is the one ultimately suggested by the study, as this seems to empirically compensate for the losses caused by the existence of correlation between successive links.

3.2.7 Summary

From reviewing the topic of travel time uncertainty, variability and reliability, the following remarks can be expressed:

- Travel time variability is very important to travellers; many travellers are risk-averse and are therefore prepared to choose a longer route if it is more reliable, i.e. if they can be certain that they will arrive on time at their destination. Thus, incorporating this in in-vehicle navigation will definitely be a very useful feature that will advance the current status of the car navigation systems technology.

- Travel time variability can be disaggregated into four components, each one of which is responsible for the differences in the travel time observed between different time periods. All four components should be considered in an in-vehicle navigation strategy.

- Travel times follow a right-skewed distribution, which is very close to the log-normal or the gamma distribution. It can be therefore safely assumed, that travel times are log-normally distributed.

- There is a considerable amount of research devoted to modelling travel time variability, which ranges from small empirical studies with limited data due to lack of advanced technologies in data collection methods, to analytical models.

- A number of reliability measures have been adopted by several studies; however, most of them assume that travel times are normally distributed, and the ones that do not do so are rather difficult to use in in-vehicle navigation, as they cannot be easily converted to a measure that is understandable by the travellers. A new measure therefore needs to be developed.

- When calculating the reliability of a path given the reliabilities of the individual links forming the path, state-of-practice methods tend to underestimate the actual value due to the existence of correlation between the link travel times – they are not statistically independent. In order to overcome this, the best experimentally tested method is the calculation of the coefficient of variation of the path as the average of the coefficients of variation of the individual links.

3.3 Quantifying reliability in in-vehicle navigation

Having reviewed previous work on measures of travel time variability and reliability, it has become apparent that a new way of quantifying the reliability of travel time for in-vehicle navigation applications is required. Namely, a measure enabling the characterisation of links and junction movements (as separate entities), and routes as "reliable" or "unreliable", such that it is possible to act accordingly in an in-vehicle navigation strategy, is needed. The new measure should, however, meet a number of requirements, ensuring its accuracy and simplicity.

The requirements of the new measure are presented at first; then, the new measure, consisting of two reliability indices, is defined based on the assumption that travel times are log-normally distributed, for both links and junction movements. After a series of mathematical formulations expressions for the derivation of the reliability indices from a distribution of speeds are given. The reliability indices of a path consisting of a series of links and junction movements are then derived. The inverse procedure, i.e. the calculation of statistical values from reliability index values is also developed. Finally, a procedure developed for calculating the reliability indices under time-dependence in analogy to the methods derived in Section 2.5 is described. Numerical examples demonstrating the application of the theoretical concepts developed are subsequently shown.

3.3.1 Requirements of a new reliability measure

The first requirement that the new reliability measure needs to meet is the comprehensibility by the driver. The measures that have been used in the literature so far have had the major drawback, that it was not possible to communicate to the driver the reliability of a route, because it was in a non-understandable form. Taking into account the fact that the driver is interested in how late or how early he/she will arrive at the destination, i.e. the amount of congestion that will be encountered rather than the probability of encountering congestion, the new measure should be designed to be easily convertible to a travel time value.

Furthermore, in order to ensure that the new reliability measure will be independent of the units used, a dimensionless quantity should be chosen. Thus, it will make no difference to the calculation of reliability whether travel time is measured in seconds or minutes, whether distance is measured in kilometres or miles and whether speed is given in metres per second, miles per hour or kilometres per hour. This will enable the comparison between links, junction movements and routes, whose characteristics are given in different units.

In addition, the new reliability measure should be independent of the length of the link, so that the division of a link into two without any physical changes leaves the reliability measure unaffected. When it comes to calculating the reliability of a path, consisting of a series of links and junction movements, it should be ensured that the measure used takes into account the statistical dependence between them, as it is very likely that congestion on one link will result in more links becoming congested too.

A new method for quantifying reliability is defined and analysed in the next sections. The method starts from analysing the variables used for links, then the concept is extended to junction movements and paths. A procedure for calculating reliability under time-dependence is also developed, and numerical examples are given for demonstration purposes.

3.3.2 Definition of the reliability indices

Two indices, characterising the reliability of link travel times and junction movement delays are presented next. The link reliability indices are defined first; then, they are extended to apply to junction movement delays.

Link travel time reliability

As travel times on a link have been assumed to be log-normally distributed, the distribution of the travel time $t(l)$ of a link l is $t(l) \sim \text{Log-N}(\mu(l), [\sigma(l)]^2)$, where $\mu(l)$ and $\sigma(l)$ are the mean and standard deviation of the natural logarithm of travel time respectively. They are connected to the mean $\bar{t}(l)$ and variance $\text{var}[t(l)]$ of the travel time distribution by the following expressions:

$$\mu(l) = \ln\left(\bar{t}(l)\right) - \frac{1}{2}\ln\left(1 + \frac{\text{var}[t(l)]}{[\bar{t}(l)]^2}\right)$$

$$\sigma(l) = \sqrt{\ln\left(1 + \frac{\text{var}[t(l)]}{[\bar{t}(l)]^2}\right)}$$

Defining the dimensionless travel time variation logarithm as

$$T_{\log}(l) = \ln\left(1 + \frac{\text{var}[t(l)]}{[\bar{t}(l)]^2}\right) \tag{3.3-1}$$

the expressions become:

$$\mu(l) = \ln\left(\bar{t}(l)\right) - \frac{1}{2}T_{\log}(l) \tag{3.3-2}$$

$$\sigma(l) = \sqrt{T_{\log}(l)} \tag{3.3-3}$$

The standard deviation of the distribution indicates the spread of the travel time values around the mean. Hence, large $\sigma(l)$ values mean that the spread of the logarithms of the travel times around the mean of the logarithms of the travel times $\mu(l)$ is great, and so is the spread of the actual travel times around the mean travel time $\bar{t}(l)$, resulting in a greater uncertainty regarding the travel time that is to be experienced on link l. Because travel times are log-normally distributed, a confidence interval of the travel time to be experienced on the link is given by $\{\mu_{geo}(l) / [\sigma_{geo}(l)]^{z_{\alpha/2}}, \mu_{geo}(l) \cdot [\sigma_{geo}(l)]^{z_{\alpha/2}}\}$, where $\mu_{geo}(l) = e^{\mu(l)}$ and $\sigma_{geo}(l) = e^{\sigma(l)}$ are the geometric mean and standard deviation of the travel time distribution respectively, and $z_{\alpha/2}$ is the standard normal distribution tail probability for a confidence coefficient α, corresponding to a confidence level of $1-\alpha$. More specifically, for confidence coefficients $\alpha = 0.1$,

$\alpha = 0.05$ and $\alpha = 0.001$, corresponding to confidence levels of 90%, 95% and 99% respectively, the tail probabilities are $z_{0.05} = 1.65$, $z_{0.025} = 1.96$ and $z_{0.0005} = 2.58$.

Thus, based on these definitions, the confidence interval of the travel time experienced on link l can be expressed as $\{e^{\mu(l) - z_{\alpha/2}\sigma(l)}, e^{\mu(l) + z_{\alpha/2}\sigma(l)}\} = \{t(l)_{\alpha/2}, t(l)_{1-\alpha/2}\}$. This is an asymmetrical interval around $\bar{t}(l)$ giving an indication of what the maximum travel time (upped bound) and the minimum travel time (lower bound) on the link can be. However, for a link to be characterised as 'reliable' or 'unreliable', the scale of the maximum and minimum travel times with respect to the mean travel time needs to be known, as the same maximum and minimum travel time values can have different effects on different mean values. For example, an additional maximum travel time value of 5 minutes is very large when the link has a mean travel time of also 5 minutes, whereas it is fairly small if the link has a mean travel time of 30 minutes. Additionally, a dimensionless length-neutral measure of the reliability of the links is required, so as to be able to compare links with each other.

Thus, two reliability indices, namely the lateness index and the earliness index, are defined. The lateness reliability index, defined as

$$r_L(l) = \frac{\bar{t}(l)}{t(l)_{1-\alpha/2}} = \frac{\bar{t}(l)}{\exp\left[\mu(l) + z_{\alpha/2}\,\sigma(l)\right]} = \exp\left[\ln\left(\bar{t}(l)\right) - \mu(l) - z_{\alpha/2}\cdot\sigma(l)\right] \qquad (3.3\text{-}4)$$

is proposed as a determining measure for the characterisation of links as 'reliable' or 'unreliable' regarding their lateness. On the other hand, the earliness reliability index, defined as

$$r_E(l) = \frac{t(l)_{\alpha/2}}{\bar{t}(l)} = \frac{\exp\left[\mu(l) - z_{\alpha/2}\,\sigma(l)\right]}{\bar{t}(l)} = \exp\left[-\ln\left(\bar{t}(l)\right) + \mu(l) - z_{\alpha/2}\cdot\sigma(l)\right] \qquad (3.3\text{-}5)$$

is proposed as a determining measure for the characterisation of links regarding their earliness.

Using equations 3.3-2 and 3.3-3, both reliability indices can be expressed in terms of the mean and variance of the travel time distributions. Thus

$$r_L(l) = \exp\left[\ln\left(\bar{t}(l)\right) - \mu(l) - z_{\alpha/2} \cdot \sigma(l)\right] \Leftrightarrow$$

$$r_L(l) = \exp\left[\ln\left(\bar{t}(l)\right) - \left(\ln\left(\bar{t}(l)\right) - ½ \cdot T_{log}(l)\right) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)}\right] \Leftrightarrow$$

$$r_L(l) = \exp\left[½ \cdot T_{log}(l) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)}\right] \quad (3.3\text{-}6)$$

and

$$r_E(l) = \exp\left[-\ln(\bar{t}(l)) + \mu(l) - z_{\alpha/2} \cdot \sigma(l)\right] \Leftrightarrow$$

$$r_E(l) = \exp\left[-\ln\left(\bar{t}(l)\right) + \left(\ln\left(\bar{t}(l)\right) - ½ \cdot T_{log}(l)\right) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)}\right] \Leftrightarrow$$

$$r_E(l) = \exp\left[-½ \cdot T_{log}(l) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)}\right] \quad (3.3\text{-}7)$$

$r_L(l)$ takes values ranging from 0 to 1, as the minimum value that the denominator in equation 3.3-4 may take is $\bar{t}(l)$. Lower values indicate low lateness reliability and high values show that the link is reliable. In the extreme cases, when $r_L(l)$ is close to 0, the maximum travel time is much longer than the mean travel time and hence the link is extremely unreliable in terms of lateness; on the other hand, when $r_L(l)$ is close to 1, the maximum travel time is approximately equal to the mean travel time, which means that almost no deviation from the mean travel time is to be expected, making the link very reliable in terms of lateness.

Similarly, the value of $r_E(l)$ also ranges between 0 and 1, as $\bar{t}(l)$ is the maximum value the nominator in equation 3.3-5 can take. Small values indicate that the minimum travel time is much smaller than the mean, implying hence a low reliability in terms of earliness, as a much earlier arrival than the one predicted is possible. Conversely, values close to 1 indicate that there is small variation of travel times and hence the link is reliable in terms of earliness.

Of course, the accuracy and reliability of both reliability indices strongly depends on the confidence level chosen. If a very high confidence level is chosen, the estimates are very reliable and one can be certain that the travel time will lie between the calculated maximum and minimum travel times; however, the confidence interval will be large and, in order to account for even the extreme values, the indices will not be representative of the non-extreme travel time values. On the other hand, if a lower confidence level is chosen, the indices will reflect the non-extreme travel times very well; however, most of the abnormally occurring long and short travel times will not be considered.

Since neither early nor late arrival at the destination are tolerated, both indices have to be taken into account when providing route guidance; however, due to the asymmetry of the interval, one of the two travel time bounds, maximum and minimum, is more critical, because it deviates more from the mean and implies therefore a greater level of uncertainty and hence, a lower reliability index. As the distribution in question is log-normal, and hence right-skewed, the maximum possible deviation from the mean is the upper bound of the interval, i.e. $t(l)_{1-\alpha/2}$, and the critical index is thus $r_L(l)$.

Junction movement delay reliability

In analogy to the travel time of links, the delay of junction movements can also be assumed to be log-normally distributed, such that the distribution of the delay $\delta(a,b)$ of a movement (a,b) is $\delta(a,b) \sim \text{Log-N}(\mu(a,b), [\sigma(a,b)]^2)$. Thus, the delay variation logarithm of movement (a,b) can be defined as

$$T_{log}(a,b) = \ln\left(1 + \frac{\text{var}[\delta(a,b)]}{[\bar{\delta}(a,b)]^2}\right) \tag{3.3-8}$$

and the lateness and earliness indices as

$$r_L(a,b) = \exp\left[½ \cdot T_{log}(a,b) - z_{\alpha/2} \cdot \sqrt{T_{log}(a,b)}\right] \tag{3.3-9}$$

and

$$r_E(a,b) = \exp\left[-½ \cdot T_{log}(a,b) - z_{\alpha/2} \cdot \sqrt{T_{log}(a,b)}\right] \tag{3.3-10}$$

respectively.

Whilst the travel times of links cannot have a value of zero, this is not the case of the junction movement delays, and therefore, an analysis of the effect of the extreme values of the mean and variance of the delays on the reliability indices is required. Namely, when the variance of the junction delay is zero, for non-zero mean delay, the delay variation logarithm is also zero and hence both reliability indices are equal to 1; such a result is expected, since the non-variability of the movement delay means that the movement is very reliable. On the other hand, as delay is by definition non-negative, it is not possible to have zero mean delay with non-zero variance, and hence, the reliability of movements with a mean delay of zero can be considered to be 1, as no variability is present.

3.3.3 Calculation of reliability from a distribution of speeds

Although in a theoretical framework one usually works with travel time distributions as in Section 3.3.2, this data is rarely available in this form and is usually expressed as speed measurements. Although the conversion of a speed measurement to a travel time measurement is simple, given that the link's length is constant, carrying out this procedure can be very inefficient, as a large number of measurements will need to be converted. Besides that, it is possible that the actual speed measurements will not be given, and only descriptive speed statistics will be provided. Thus, working with the speed distribution and attempting to relate the reliability of speed with the reliability of travel time, the relationship of the travel time and speed variances needs to be investigated.

Namely, for link l of length $\lambda(\mathrm{l})$, normally distributed speed $\upsilon(\mathrm{l})$ and log-normally distributed travel time t(l), it is $\mathrm{t(l)} = \lambda(\mathrm{l}) / \upsilon(\mathrm{l})$; when it comes to the mean travel time, $\bar{\mathrm{t}}(\mathrm{l}) = \lambda(\mathrm{l}) / \bar{\upsilon}_s(\mathrm{l})$, where $\bar{\upsilon}_s(\mathrm{l})$ represents space-mean speed, which is defined as the harmonic mean of the speed distribution. The travel time variation logarithm, defined in equation 3.3-1, is thus affected, such that

$$\mathrm{T_{log}(l)} = \ln\left(1 + \frac{\mathrm{var}\left(\frac{\lambda(\mathrm{l})}{\upsilon(\mathrm{l})}\right)}{\left(\frac{\lambda(\mathrm{l})}{\bar{\upsilon}_s(\mathrm{l})}\right)^2}\right) = \ln\left(1 + \frac{[\lambda(\mathrm{l})]^2 \cdot \mathrm{var}\left(\frac{1}{\upsilon(\mathrm{l})}\right)}{[\lambda(\mathrm{l})]^2 \cdot \left(\frac{1}{\bar{\upsilon}_s(\mathrm{l})}\right)^2}\right) \Leftrightarrow$$

$$\mathrm{T_{log}(l)} = \ln\left(1 + [\bar{\upsilon}_s(\mathrm{l})]^2 \cdot \mathrm{var}\left(\frac{1}{\upsilon(\mathrm{l})}\right)\right) \qquad (3.3\text{-}11)$$

In order to evaluate $T_{log}(l)$, the variance of the distribution of the inverse of speed needs to be known. Performing a first degree Taylor series expansion of the $1/\upsilon(l)$ term around $E[\upsilon(l)]$ (the expected value of the speed distribution, which is the time-mean speed $\bar{\upsilon}(l)$), the following expression is obtained:

$$\frac{1}{\upsilon(l)} \cong \frac{1}{E[\upsilon(l)]} - (\upsilon(l) - E[\upsilon(l)]) \cdot \frac{1}{(E[\upsilon(l)])^2} \tag{3.3-12}$$

Taking the expected values of both sides of the expression:

$$E\left(\frac{1}{\upsilon(l)}\right) \cong \frac{1}{E[\upsilon(l)]} - E[(\upsilon(l) - E[\upsilon(l)])] \cdot \frac{1}{(E[\upsilon(l)])^2} = \frac{1}{E[\upsilon(l)]} \tag{3.3-13}$$

Subtracting equation 3.3-13 from equation 3.3-12:

$$\frac{1}{\upsilon(l)} - \frac{1}{E[\upsilon(l)]} \cong -(\upsilon(l) - E[\upsilon(l)]) \cdot \frac{1}{(E[\upsilon(l)])^2} \Leftrightarrow$$

$$\left(\frac{1}{\upsilon(l)} - \frac{1}{E[\upsilon(l)]}\right)^2 \cong (\upsilon(l) - E[\upsilon(l)])^2 \cdot \frac{1}{(E[\upsilon(l)])^4} \Leftrightarrow$$

$$E\left(\left(\frac{1}{\upsilon(l)} - \frac{1}{E[\upsilon(l)]}\right)^2\right) \cong E((\upsilon(l) - E[\upsilon(l)])^2) \cdot \frac{1}{(E[\upsilon(l)])^4} \Leftrightarrow$$

$$\mathrm{var}\left(\frac{1}{\upsilon(l)}\right) \cong \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^4} \tag{3.3-14}$$

Hence, substituting the variance term in equation 3.3-11 with the result derived in equation 3.3-14:

$$T_{log}(l) \cong \ln\left(1 + [\bar{\upsilon}_s(l)]^2 \cdot \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^4}\right) = \ln\left(1 + \frac{[\omega(l)]^2 \cdot \mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^2}\right) \tag{3.3-15}$$

where $\omega(l) = \bar{\upsilon}_s(l)/\bar{\upsilon}(l)$ is the ratio of the space mean speed over the time mean speed. Since the time mean speed is always greater than or equal to the space mean speed, $\omega(l)$ only takes values in the range 0 to 1. Using the finding of Rakha and Wang (2005) relating time-mean speed and space-mean speed, i.e.

$$\bar{\upsilon}_s(l) \cong \bar{\upsilon}(l) - \frac{\mathrm{var}[\upsilon(l)]}{\bar{\upsilon}(l)}$$

equation 3.3-15 can be reformulated as

$$T_{\log}(l) \cong \ln\left(1 + \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^2} \cdot \left(1 - \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^2}\right)^2\right) \tag{3.3-16}$$

which can be useful when the actual speed measurements are not given and only the time-mean speed and variance of the distribution are provided.

Expression 3.3-15 or 3.3-16 can then be substituted into equations 3.3-6 and 3.3-7, so as to calculate the earliness and lateness indices in terms of the speed distribution. Using equation 3.3-14, it can also be found that:

$$\mathrm{var}[t(l)] = \mathrm{var}\left(\frac{\lambda(l)}{\upsilon(l)}\right) = [\lambda(l)]^2 \cdot \mathrm{var}\left(\frac{1}{\upsilon(l)}\right) \cong [\lambda(l)]^2 \cdot \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^4} \tag{3.3-17}$$

3.3.4 Calculation of the path reliability indices

So far, the definition of the reliability indices of individual links is fairly simple; however, the problem becomes slightly more complicated when it comes to computing the reliability indices of a path, consisting of a series of links and junction movements. Namely, the lateness and earliness indices $R_L(p)$ and $R_E(p)$ of path p, consisting of n elements (links and junction movements) and having a log-normally distributed travel time $T(p)$, are:

$$R_L(p) = \frac{\bar{T}(p)}{T(p)_{1-\alpha/2}} = \exp\left[\tfrac{1}{2} \cdot T_{\log}(p) - z_{\alpha/2} \cdot \sqrt{T_{\log}(p)}\right] \tag{3.3-18}$$

and

$$R_E(p) = \frac{T(p)_{\alpha/2}}{\overline{T}(p)} = \exp\left[-\tfrac{1}{2}\cdot T_{log}(p) - z_{\alpha/2}\cdot\sqrt{T_{log}(p)}\right] \quad (3.3\text{-}19)$$

respectively, where the path travel time variation logarithm is

$$T_{log}(p) = \ln\left(1 + \frac{var[T(p)]}{[\overline{T}(p)]^2}\right).$$

While the mean travel time of the path can be easily calculated, simply by adding up the mean travel times of the elements forming the path, such that

$$\overline{T}(p) = \sum_{i=1}^{n} \overline{t}(l_i) \quad (3.3\text{-}20)$$

the issue that arises is the calculation of the total travel time variance. As described in section 3.2.6, Rakha et al (2006) proved that the most accurate method for computing the travel time variance of a path, is to compute the expected coefficient of variation as the conditional expectation over all realisations of the various elements that make up the path and thus assume that the path's coefficient of variation is the mean coefficient of variation over all elements. Adjusting equation 3.2-9,

$$var[T(p)] = \frac{[\overline{T}(p)]^2}{n^2}\cdot\left(\sum_{i=1}^{n}\frac{\sqrt{var[t(l_i)]}}{\overline{t}(l_i)}\right)^2 \quad (3.3\text{-}21)$$

It should be noted here, that for reasons of simplicity, junction movements can be assumed to be equivalent to links in the path variance calculation, and thus the notation $t(l_i)$ can be considered to also apply to junction movements in the following expressions.

Using equation 3.3-21, the path travel time variation logarithm becomes

$$T_{log}(p) = \ln\left(1 + \frac{\frac{[\overline{T}(p)]^2}{n^2}\cdot\left(\sum_{i=1}^{n}\frac{\sqrt{var[t(l_i)]}}{\overline{t}(l_i)}\right)^2}{[\overline{T}(p)]^2}\right) \Leftrightarrow$$

$$T_{log}(p) = \ln\left(1 + \left(\frac{1}{n} \cdot \sum_{i=1}^{n} \frac{\sqrt{var[t(l_i)]}}{\bar{t}(l_i)}\right)^2\right) \Leftrightarrow$$

$$T_{log}(p) = \ln\left(1 + \left(\frac{1}{n} \cdot \sum_{i=1}^{n} \sqrt{\frac{var[t(l_i)]}{[\bar{t}(l_i)]^2}}\right)^2\right) \tag{3.3-22}$$

Reformulating equation 3.3-1 for the link travel time variation logarithm:

$$T_{log}(l) = \ln\left(1 + \frac{var[t(l)]}{[\bar{t}(l)]^2}\right) \Leftrightarrow$$

$$\exp\left[T_{log}(l)\right] = 1 + \frac{var[t(l)]}{[\bar{t}(l)]^2} \Leftrightarrow$$

$$\exp\left[T_{log}(l)\right] - 1 = \frac{var[t(l)]}{[\bar{t}(l)]^2}.$$

Hence, equation 3.3-22 becomes:

$$T_{log}(p) = \ln\left(1 + \left(\frac{1}{n} \cdot \sum_{i=1}^{n} \sqrt{\exp\left[T_{log}(l_i)\right] - 1}\right)^2\right) \tag{3.3-23}$$

Substituting expression 3.3-23 into equations 3.3-18 and 3.3-19 enables the calculation of the earliness and lateness indices for path p.

3.3.5 Calculation of the travel time variation logarithm from reliability

So far, the reliability of links, junction movements and paths has been calculated, assuming that the travel time variation logarithm is either available, or can be obtained from the travel time or speed distribution. In this section, the inverse procedure is developed, which assumes that the reliability index values are available, but not the link distributions; a formula for the variation logarithm in terms of reliability is derived, which may be useful for computing the reliability indices of a path when only the reliability indices of the links and junction movements forming the path are available.

Namely, reformulating equations 3.3-6 and 3.3-7, expressions for the travel time variation logarithm in terms of the reliability indices are obtained. From equation 3.3-6:

$$r_L(l) = \exp\left[½ \cdot T_{log}(l) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)}\right] \Leftrightarrow$$

$$\ln\left(r_L(l)\right) = ½ \cdot T_{log}(l) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)} \Leftrightarrow$$

$$½ \cdot T_{log}(l) - z_{\alpha/2} \cdot \sqrt{T_{log}(l)} - \ln\left(r_L(l)\right) = 0$$

and solving the quadratic equation in terms of the $\sqrt{T_{log}(l)}$:

$$\sqrt{T_{log}(l)} = z_{\alpha/2} \pm \sqrt{\left(z_{\alpha/2}\right)^2 + 2\ln\left(r_L(l)\right)} \Leftrightarrow$$

$$T_{log}(l) = \left(z_{\alpha/2} \pm \sqrt{\left(z_{\alpha/2}\right)^2 + 2\ln\left(r_L(l)\right)}\right)^2$$

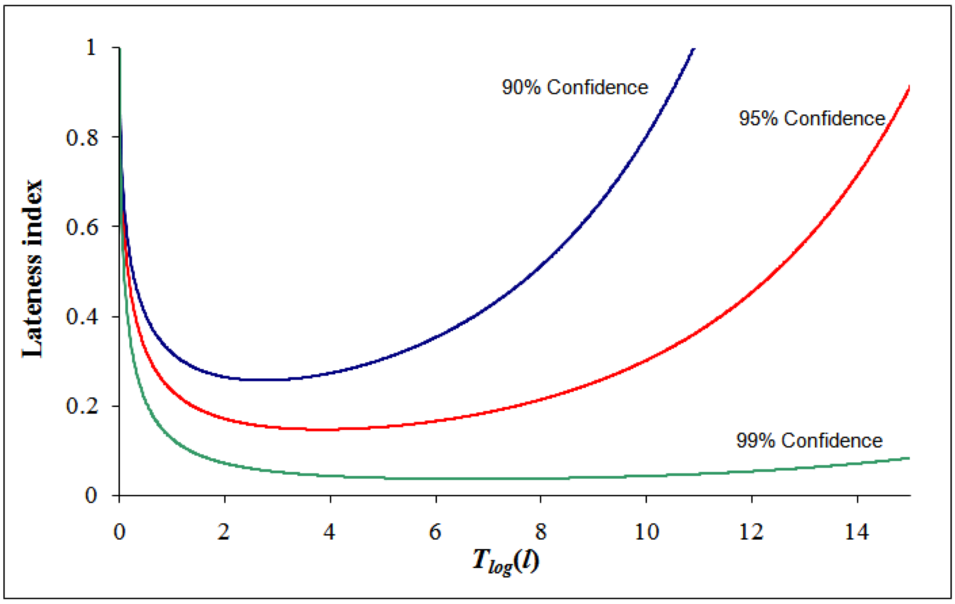


Figure 3.3-1: The relationship between $T_{log}(l)$ and $r_L(l)$

Thus, there are two solutions for $T_{log}(l)$, one of which needs to be chosen. Plotting $r_L(l)$ against $T_{log}(l)$ for different confidence levels (Figure 3.3-1), it can be seen that $r_L(l)$ decreases with increasing $T_{log}(l)$ values, until it reaches a minimum, from where it starts increasing again. The reason for this is the fact that the log-normal travel time distribution is right-skewed, the skew being determined by the standard deviation of the logarithm of travel time, $\sigma(l)$, which is equal to the root of the travel time variation logarithm (equation 3.3-3).

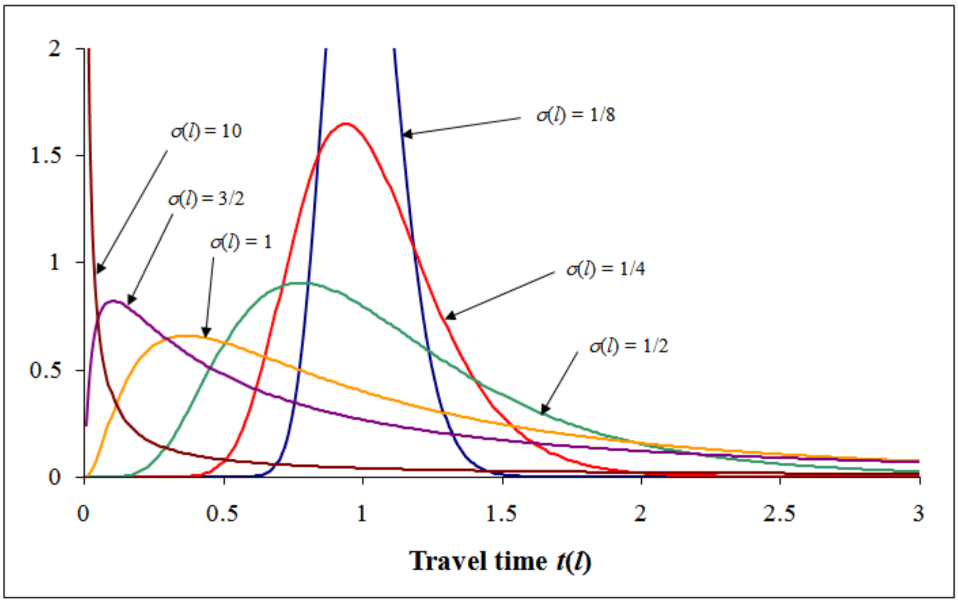


Figure 3.3-2: The influence of $\sigma(\mathrm{l})$ on the skew of the log-normal distribution

As can be seen on Figure 3.3-2, for large $\sigma(\mathrm{l})$ values and for the same $\mu(\mathrm{l})$ value, the distribution becomes so strongly skewed that the $1\text{-}\alpha/2$ percentile travel time starts shifting again towards the mean, thus increasing $\mathrm{r_L(l)}$. In theory, sufficiently high $\mathrm{T_{log}(l)}$ values would result in $\mathrm{r_L(l)}$ values larger than 1; however, this would only occur if the $1\text{-}\alpha/2$ percentile travel time becomes smaller than the mean travel time, which is not likely in reality.

It will now be checked, whether the largest of the two solutions for $\mathrm{T_{log}(l)}$ is infeasible as being outside of the realistic range of the travel time distribution, i.e. the standard deviation of the travel time is much larger than the mean travel time, and if, as such, it can be discarded. Calculating $\mathrm{T_{log}{*}(l)}$, which is the $\mathrm{T_{log}(l)}$ value for which $\mathrm{r_L(l)}$ becomes minimum:

$$\frac{\mathrm{dr_L(l)}}{\mathrm{dT_{log}(l)}} = \frac{\mathrm{d}}{\mathrm{dT_{log}(l)}}\left(\exp\left[½\cdot \mathrm{T_{log}(l)} - \mathrm{z}_{\alpha/2}\cdot\sqrt{\mathrm{T_{log}(l)}}\right]\right) =$$

$$= \exp\left[½\cdot \mathrm{T_{log}(l)} - \mathrm{z}_{\alpha/2}\cdot\sqrt{\mathrm{T_{log}(l)}}\right] \cdot \frac{\mathrm{d}}{\mathrm{dT_{log}(l)}}\left(½\cdot \mathrm{T_{log}(l)} - \mathrm{z}_{\alpha/2}\cdot\sqrt{\mathrm{T_{log}(l)}}\right).$$

and

$$\frac{dr_L(l)}{dT_{log}(l)} = 0 \Leftrightarrow$$

$$\exp\left[½ \cdot T_{log}*(l) - z_{\alpha/2} \cdot \sqrt{T_{log}*(l)}\right] \cdot ½ \cdot \left(1 - z_{\alpha/2} / \sqrt{T_{log}*(l)}\right) = 0 \Leftrightarrow$$

$$1 - z_{\alpha/2} / \sqrt{T_{log}*(l)} = 0 \Leftrightarrow$$

$$\sqrt{T_{log}*(l)} = z_{\alpha/2} \Leftrightarrow$$

$$T_{log}*(l) = \left(z_{\alpha/2}\right)^2$$

This corresponds to a $T_{log}*(l)$ value of 2.72, 3.84 and 6.66 for 90%, 95% and 99% confidence respectively. The corresponding minimum lateness index value, $r_L*(l)$ is:

$$r_L*(l) = \exp\left[½ \cdot T_{log}*(l) - z_{\alpha/2} \cdot \sqrt{T_{log}*(l)}\right] \Leftrightarrow$$

$$r_L*(l) = \exp\left[- ½ \cdot \left(z_{\alpha/2}\right)^2\right]$$

which translates to 0.256, 0.146 and 0.035 for 90%, 95% and 99% confidence respectively.

From the $T_{log}*(l)$ value obtained, it can be found that

$$\ln\left(1 + \left(\frac{var[t(l)]}{[\bar{t}(l)]^2}\right)^*\right) = \left(z_{\alpha/2}\right)^2 \Leftrightarrow$$

$$\left(\left(\frac{\sqrt{var[t(l)]}}{[\bar{t}(l)]}\right)^*\right)^2 = \exp\left[\left(z_{\alpha/2}\right)^2\right] - 1 \Leftrightarrow$$

$$\left(\frac{\sqrt{var[t(l)]}}{[\bar{t}(l)]}\right)^* = \sqrt{\exp\left[\left(z_{\alpha/2}\right)^2\right] - 1}$$

which means that $r_L(l)$ values begin to increase again when the ratio of the standard deviation over the mean travel time becomes larger than $\sqrt{\exp[(z_{\alpha/2})^2] - 1}$. This corresponds to 3.77, 6.75 and 27.87 for 90%, 95% and 99% confidence respectively. While the ratio for the 99% confidence level is unrealistically large, this is not the case for the 90% and 95% levels, as cases where the standard deviation of travel time is 4 or 7 times larger than the mean travel time may be rare, but are still possible. Hence, the largest of the two solutions for $T_{log}(l)$ cannot be discarded, as there is a possibility that it is in the range of the travel time distribution.

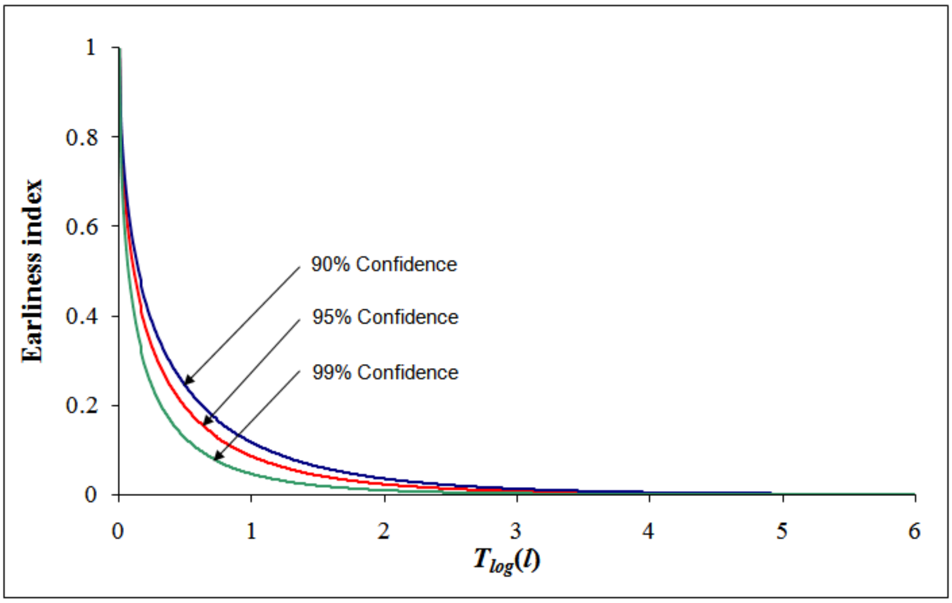


Figure 3.3-3: The relationship between $T_{log}(l)$ and $r_E(l)$

A solution based on the earliness reliability index will therefore be sought, in order to express $T_{log}(l)$. Plotting $r_E(l)$ against $T_{log}(l)$ (Figure 3.3-3), it can be seen that only one solution in terms of $T_{log}(l)$ exists. From equation 7:

$$r_E(l) = \exp\left[-\tfrac{1}{2}\cdot T_{log}(l) - z_{\alpha/2}\cdot\sqrt{T_{log}(l)}\right] \Leftrightarrow$$

$$\ln\left(r_E(l)\right) = -\tfrac{1}{2}\cdot T_{log}(l) - z_{\alpha/2}\cdot\sqrt{T_{log}(l)} \Leftrightarrow$$

$$\tfrac{1}{2}\cdot T_{log}(l) + z_{\alpha/2}\cdot\sqrt{T_{log}(l)} + \ln\left(r_E(l)\right) = 0$$

and solving the quadratic equation in terms of the $\sqrt{T_{log}(l)}$:

$$\sqrt{T_{log}(l)} = -z_{\alpha/2} \pm \sqrt{\left(z_{\alpha/2}\right)^2 - 2\ln\left(r_E(l)\right)}$$

One of the two solutions is always negative and is therefore infeasible. The expression for $T_{log}(l)$ is thus:

$$\sqrt{T_{log}(l)} = -z_{\alpha/2} \pm \sqrt{(z_{\alpha/2})^2 - 2\ln(r_E(l))} \Leftrightarrow$$

$$T_{log}(l) = \left(-z_{\alpha/2} + \sqrt{(z_{\alpha/2})^2 - 2\ln(r_E(l))}\right)^2 \qquad (3.3\text{-}24)$$

3.3.6 Time-dependent link reliability calculation

Having derived a formula for calculating the travel time variation logarithm from reliability index values, this sub-section presents the procedure for calculating time-dependent reliability index values, when these are not constant on a link but vary with time. The methodology employed can be seen as an addition to the procedure described in Section 2.5, which is aimed at calculating the time-dependent travel time on a link, as the resulting reliability index values are a combination of the individual earliness and lateness values corresponding to each time interval.

As one would expect, the travel time values on a link between different time intervals are not statistically independent from each other (i.e. the link exhibits autocorrelation), as the travel time at a certain interval depends on the travel time on the same link in previous intervals. Hence, the concept behind the method developed is, as in the case of the calculation of the reliability of a path, to empirically compensate for the loss in accuracy due to the presence of autocorrelation by computing the expected coefficient of variation as the conditional expectation over all time intervals involved in the travel of a vehicle on a specific link, and thus assume that the coefficient of variation experienced by the vehicle is the mean coefficient of variation over all time intervals the trip on the link covers. A modification of equation 3.3-23 is thus used.

More specifically, as was described in Section 2.5, if Z is a set of time intervals $[\tau_k, \tau_{k+1})$, $k = 0, 1, \ldots, m-1$ separated by the time points $\tau_0 < \tau_1 < \tau_2 < \ldots < \tau_{m-1} < \tau_m$ and covering one day, then for every time interval $\Delta\tau_k : [\tau_k, \tau_{k+1})$, every link l has a space-mean speed value $\upsilon_k(l)$, a lateness index $r_{Lk}(l)$ and an earliness index $r_{Ek}(l)$, while every movement (a,b) has a mean delay value $\delta_k(a,b)$, a lateness index $r_{Lk}(a,b)$ and an earliness index $r_{Ek}(a,b)$. Using equation 3.3-24, the travel time variation logarithms $T_{logk}(l)$ and $T_{logk}(a,b)$, for every link l and junction movement (a,b) in each time interval is obtained; these are then going to be input into equation 3.3-23, so as to calculate the travel time variation logarithm over all time intervals of the vehicle's trip on the link or junction movement.

When performing the forward time-dependent expected travel time $t(l)$ calculation of link l for a specific entry time $\tau_{ent}(l) \in Z$ with $\tau_0 \leq \tau_{ent}(l) < \tau_1$, the travel time variation logarithm is computed as follows:

$$T_{log}(l) = T_{log0}(l) \qquad \text{if} \quad \lambda(l) / \upsilon_0(l) < \tau_1 - \tau_{ent}(l)$$

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{2}\cdot\left(T_0 + \sqrt{\exp[T_{log1}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_0) / \upsilon_1(l) < \tau_2 - \tau_1$$

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{3}\cdot\left(T_1 + \sqrt{\exp[T_{log2}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_1) / \upsilon_2(l) < \tau_3 - \tau_2$$

:
:

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{k+1}\cdot\left(T_{k-1} + \sqrt{\exp[T_{logk}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{k-1}) / \upsilon_k(l) < \tau_{k+1} - \tau_k$$

:
:

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{m}\cdot\left(T_{m-2} + \sqrt{\exp[T_{logm-1}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{m-2}) / \upsilon_{m-1}(l) < \tau_m - \tau_{m-1}$$

where:

$\lambda_0 = \upsilon_0(l)\,(\tau_1 - \tau_{ent}(l))$ and $T_0 = \sqrt{\exp[T_{log0}(l)] - 1}$

$\lambda_1 = \lambda_0 + \upsilon_1(l)\,(\tau_2 - \tau_1)$ and $T_1 = T_0 + \sqrt{\exp[T_{log1}(l)] - 1}$

:
:

$\lambda_{k-1} = \lambda_{k-2} + \upsilon_{k-1}(l)\,(\tau_k - \tau_{k-1})$ and $T_{k-1} = T_{k-2} + \sqrt{\exp[T_{logk-1}(l)] - 1}$

:
:

$\lambda_{m-2} = \lambda_{m-3} + \upsilon_{m-2}(l)\,(\tau_{m-1} - \tau_{m-2})$ and $T_{m-2} = T_{k-3} + \sqrt{\exp[T_{logk-2}(l)] - 1}$

On the other hand, when a reverse time-dependent travel time calculation is carried out, i.e. for a known exit time $\tau_{ex}(l)$, where $\tau_{ex}(l) \in Z$ and $\tau_{m-1} \leq \tau_{ex}(l) < \tau_m$, the travel time variation logarithm of link l is computed as follows:

$$T_{log}(l) = T_{logm\text{-}1}(l) \qquad \text{if} \quad \lambda(l) \,/\, \upsilon_{m\text{-}1}(l) < \tau_{ex}(l) - \tau_{m\text{-}1}$$

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{2}\cdot\left(T_{m\text{-}1} + \sqrt{\exp[T_{logm\text{-}2}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{m\text{-}1}) \,/\, \upsilon_{m\text{-}2}(l) < \tau_{m\text{-}1} - \tau_{m\text{-}2}$$

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{3}\cdot\left(T_{m\text{-}2} + \sqrt{\exp[T_{logm\text{-}3}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{m\text{-}2}) \,/\, \upsilon_{m\text{-}3}(l) < \tau_{m\text{-}2} - \tau_{m\text{-}3}$$

:
:

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{\text{m-k}}\cdot\left(T_{k+1} + \sqrt{\exp[T_{logk}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{k+1}) \,/\, \upsilon_{k}(l) < \tau_{k+1} - \tau_{k}$$

:
:

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{\text{m-1}}\cdot\left(T_{2} + \sqrt{\exp[T_{log1}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{2}) \,/\, \upsilon_{1}(l) < \tau_{2} - \tau_{1}$$

$$\text{else } T_{log}(l) = \ln\left(1 + \left(\frac{1}{\text{m}}\cdot\left(T_{1} + \sqrt{\exp[T_{log0}(l)] - 1}\right)\right)^2\right) \qquad \text{if} \quad (\lambda(l) - \lambda_{1}) \,/\, \upsilon_{0}(l) < \tau_{1} - \tau_{0}$$

where:

$\lambda_{m\text{-}1} = \upsilon_{m\text{-}1}(l)\,(\tau_{ex}(l) - \tau_{m\text{-}1})$ and $T_{m\text{-}1} = \sqrt{\exp[T_{logm\text{-}1}(l)] - 1}$

$\lambda_{m\text{-}2} = \lambda_{m\text{-}1} + \upsilon_{m\text{-}2}(l)\,(\tau_{m\text{-}1} - \tau_{m\text{-}2})$ and $T_{m\text{-}2} = T_{m\text{-}1} + \sqrt{\exp[T_{logm\text{-}2}(l)] - 1}$

:
:

$\lambda_{k+1} = \lambda_{k+2} + \upsilon_{k+1}(l)\,(\tau_{k+2} - \tau_{k+1})$ and $T_{k+1} = T_{k+2} + \sqrt{\exp[T_{logk+1}(l)] - 1}$

:
:

$\lambda_{2} = \lambda_{3} + \upsilon_{2}(l)\,(\tau_{3} - \tau_{2})$ and $T_{2} = T_{3} + \sqrt{\exp[T_{log2}(l)] - 1}$

$\lambda_{1} = \lambda_{2} + \upsilon_{1}(l)\,(\tau_{2} - \tau_{1})$ and $T_{1} = T_{2} + \sqrt{\exp[T_{log1}(l)] - 1}$

Following the calculation of the travel time variation logarithm, this can be next substituted into equations 3.3-6 and 3.3-7 so as to obtain the lateness and earliness indices for link l. Naturally, the same procedure is followed for each junction movement (a,b).

3.3.7 Numerical examples

With the help of three numerical examples, the theoretical findings derived in the previous sub-sections are demonstrated and verified here. The first example aims at demonstrating the calculation of the reliability indices and travel time intervals of individual links and of a path; the second example compares the accuracy of the calculation of reliability from a travel time distribution and a speed distribution; finally, the third example performs a calculation of reliability index values for a link under time-dependent conditions.

Example 1: Calculation of link and path reliability indices

For path p shown on Figure 3.3-4, consisting of five elements (three links, connected with each other by two junction movements), the reliability indices and travel time intervals for each element separately, as well as for the entire path, at a confidence level of 95%, are required. The mean travel time and standard deviation of each element are given on Table 3.3-1.

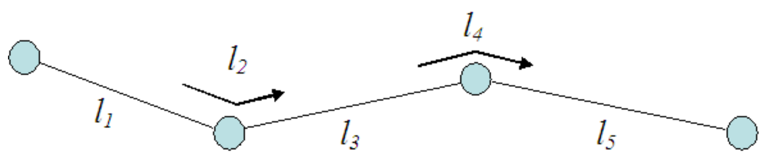


Figure 3.3-4: Path p, consisting of five elements

Table 3.3-1: Mean travel time and standard deviation for the five elements of path p

Element	Mean travel time (minutes)	Standard deviation (minutes)
l_1	5	1
l_2	1	0
l_3	3	1
l_4	2	0.2
l_5	6	3

The travel time variation logarithms are calculated first:

$$T_{log}(l_1) = \ln\left(1 + \frac{var[t(l_1)]}{[\bar{t}(l_1)]^2}\right) = \ln\left(1 + \frac{1^2}{5^2}\right) = 0.039$$

$$T_{log}(l_2) = \ln\left(1 + \frac{var[t(l_2)]}{[\bar{t}(l_2)]^2}\right) = \ln\left(1 + \frac{0^2}{1^2}\right) = 0$$

$$T_{log}(l_3) = \ln\left(1 + \frac{var[t(l_3)]}{[\bar{t}(l_3)]^2}\right) = \ln\left(1 + \frac{1^2}{3^2}\right) = 0.105$$

$$T_{log}(l_4) = \ln\left(1 + \frac{var[t(l_4)]}{[\bar{t}(l_4)]^2}\right) = \ln\left(1 + \frac{0.2^2}{2^2}\right) = 0.01$$

$$T_{log}(l_5) = \ln\left(1 + \frac{var[t(l_5)]}{[\bar{t}(l_5)]^2}\right) = \ln\left(1 + \frac{3^2}{6^2}\right) = 0.223$$

For 95% confidence, $\alpha = 0.05$ and $z_{0.025} = 1.96$. Then the lateness and earliness indices for each element can be calculated:

$$r_L(l_1) = \exp\left[½ \cdot T_{log}(l_1) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_1)}\right] = \exp\left[½ \cdot 0.039 - 1.96 \cdot \sqrt{0.039}\right] = 0.69$$

$$r_E(l_1) = \exp\left[-½ \cdot T_{log}(l_1) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_1)}\right] = \exp\left[-½ \cdot 0.039 - 1.96 \cdot \sqrt{0.039}\right] = 0.67$$

$$r_L(l_2) = \exp\left[½ \cdot T_{log}(l_2) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_2)}\right] = \exp\left[½ \cdot 0 - 1.96 \cdot \sqrt{0}\right] = 1$$

$$r_L(l_2) = \exp\left[-½ \cdot T_{log}(l_2) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_2)}\right] = \exp\left[-½ \cdot 0 - 1.96 \cdot \sqrt{0}\right] = 1$$

$$r_L(l_3) = \exp\left[½ \cdot T_{log}(l_3) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_3)}\right] = \exp\left[½ \cdot 0.105 - 1.96 \cdot \sqrt{0.105}\right] = 0.56$$

$$r_E(l_3) = \exp\left[-½ \cdot T_{log}(l_3) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_3)}\right] = \exp\left[-½ \cdot 0.105 - 1.96 \cdot \sqrt{0.105}\right] = 0.50$$

$$r_L(l_4) = \exp\left[½ \cdot T_{log}(l_4) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_4)}\right] = \exp\left[½ \cdot 0.01 - 1.96 \cdot \sqrt{0.01}\right] = 0.83$$

$$r_E(l_4) = \exp\left[-½ \cdot T_{log}(l_4) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_4)}\right] = \exp\left[-½ \cdot 0.01 - 1.96 \cdot \sqrt{0.01}\right] = 0.82$$

$$r_L(l_5) = \exp\left[½ \cdot T_{log}(l_5) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_5)}\right] = \exp\left[½ \cdot 0.223 - 1.96 \cdot \sqrt{0.223}\right] = 0.44$$

$$r_E(l_5) = \exp\left[-½ \cdot T_{log}(l_5) - z_{\alpha/2} \cdot \sqrt{T_{log}(l_5)}\right] = \exp\left[-½ \cdot 0.223 - 1.96 \cdot \sqrt{0.223}\right] = 0.35$$

Demonstrating the inverse procedure, i.e. the derivation of the travel time variation logarithm from the reliability indices, the example of l_5 is taken. Employing equation 3.3-24:

$$T_{log}(l_5) = \left(-z_{\alpha/2} + \sqrt{(z_{\alpha/2})^2 - 2\ln(r_E(l_5))}\right)^2 = \left(-1.96 + \sqrt{(1.96)^2 - 2\ln(0.35)}\right)^2 = 0.228$$

Travel time intervals can be determined for each link:

$t(l_1)_{0.025} = \bar{t}(l_1) \cdot r_E(l_1) = 5 \cdot 0.67 = 3.35$ min and $t(l_1)_{0.975} = \frac{\bar{t}(l_1)}{r_L(l_1)} = \frac{5}{0.69} = 7.25$ min

$t(l_2)_{0.025} = \bar{t}(l_2) \cdot r_E(l_2) = 0 \cdot 1 = 0$ min and $t(l_2)_{0.975} = \frac{\bar{t}(l_2)}{r_L(l_2)} = \frac{0}{1} = 0$ min

$t(l_3)_{0.025} = \bar{t}(l_3) \cdot r_E(l_3) = 3 \cdot 0.50 = 1.5$ min and $t(l_3)_{0.975} = \frac{\bar{t}(l_3)}{r_L(l_3)} = \frac{3}{0.56} = 5.36$ min

$t(l_4)_{0.025} = \bar{t}(l_4) \cdot r_E(l_4) = 2 \cdot 0.82 = 1.64$ min and $t(l_4)_{0.975} = \frac{\bar{t}(l_4)}{r_L(l_4)} = \frac{2}{0.83} = 2.41$ min

$t(l_5)_{0.025} = \bar{t}(l_5) \cdot r_E(l_5) = 6 \cdot 0.35 = 2.1$ min and $t(l_5)_{0.975} = \frac{\bar{t}(l_5)}{r_L(l_5)} = \frac{6}{0.44} = 13.64$ min

It can be thus seen, that, as the indices show, the most reliable element is the junction movement, whose standard deviation from the mean is zero. Conversely, the most unreliable element is the last link of the path, on which the travel time experienced ranges from 2.1 minutes to 13.64 minutes. Compared with the mean, which is 6 minutes, there is a possible +7.64/-3.9 minute variation, which is also expressed by the lower reliability index values.

To calculate the reliability indices of path p, the travel time variation logarithm of the entire path is computed, using equation 3.3-23:

$$T_{log}(p) = \ln\left(1 + \left(\frac{1}{n} \cdot \sum_{i=1}^{n}\sqrt{\exp[T_{log}(l_i)] - 1}\right)^2\right)$$

$$= \ln\left(1 + \left(\frac{1}{5} \cdot \left(\sqrt{\exp[0.039]-1} + \sqrt{\exp[0]-1} + \sqrt{\exp[0.105]-1} + \sqrt{\exp[0.01]-1} + \sqrt{\exp[0.223]-1}\right)\right)^2\right)$$

$$= \ln\left(1 + \left(\frac{1}{5} \cdot (0.199 + 0 + 0.333 + 0.1 + 0.5)\right)^2\right) = 0.05$$

The lateness and earliness indices for the path are:

$$R_L(p) = \exp\left[\tfrac{1}{2} \cdot T_{\log}(p) - z_{\alpha/2} \cdot \sqrt{T_{\log}(p)}\right] = \exp\left[\tfrac{1}{2} \cdot 0.05 - 1.96 \cdot \sqrt{0.05}\right] = 0.66$$

$$R_E(p) = \exp\left[-\tfrac{1}{2} \cdot T_{\log}(p) - z_{\alpha/2} \cdot \sqrt{T_{\log}(p)}\right] = \exp\left[-\tfrac{1}{2} \cdot 0.05 - 1.96 \cdot \sqrt{0.05}\right] = 0.63$$

The mean total travel time for the path is:

$$\overline{T}(p) = \sum_{i=1}^{n} \overline{t}(l_i) = 5 + 1 + 3 + 2 + 6 = 17 \text{ min}$$

Hence, the travel time interval for the entire path at the 95% confidence level is:

$$T(p)_{0.025} = \overline{T}(p) \cdot R_E(p) = 17 \cdot 0.63 = 10.71 \text{ min} \quad \text{and} \quad T(p)_{0.975} = \frac{\overline{T}(p)}{R_L(p)} = \frac{17}{0.66} = 25.76 \text{ min}$$

which implies a possible +8.76/-6.29 min deviation from the mean. As the indices show, the path is not very reliable, as an additional 50% of its mean travel time is possible to be encountered.

Example 2: Calculation of link reliability from travel time and speed distributions

Table 3.3-2: Descriptive statistics for the speed and travel time distributions

	Mean	**Standard deviation**
Speed Distribution	15.21 m/s	3.11 m/s
Travel Time Distribution	6.9 s	1.68 s

In this example, from a normal distribution of 200 speed measurements on link l, which is $\lambda(l)$ = 100 m long, the log-normal distribution of travel times is extracted. The reliability indices of the link at a confidence level of 95% are computed from both the travel time distribution and from the speed distribution directly, using the formulae in section 3.3.3, and the results are compared, so as to demonstrate the accuracy of the calculation from the speed distribution. The descriptive statistics of the two distributions are given on Table 3.3-2, while the entire range of the distribution values is given in Appendix B.

Starting from the calculation of reliability from the travel time distribution, similarly to Example 1, the following results are obtained:

$$T_{\log}(l) = \ln\left(1 + \frac{\mathrm{var}[t(l)]}{[\bar{t}(l)]^2}\right) = \ln\left(1 + \frac{1.68^2}{6.9^2}\right) = 0.058$$

$$r_L(l) = \exp\left[½ \cdot T_{\log}(l) - z_{\alpha/2} \cdot \sqrt{T_{\log}(l)}\right] = \exp\left[½ \cdot 0.058 - 1.96 \cdot \sqrt{0.058}\right] = 0.64$$

$$r_E(l) = \exp\left[-½ \cdot T_{\log}(l) - z_{\alpha/2} \cdot \sqrt{T_{\log}(l)}\right] = \exp\left[-½ \cdot 0.058 - 1.96 \cdot \sqrt{0.058}\right] = 0.61$$

and thus

$t(l)_{0.025} = \bar{t}(l) \cdot r_E(l) = 6.9 \cdot 0.61 = 4.21$ s and $t(l)_{0.975} = \frac{\bar{t}(l)}{r_L(l)} = \frac{6.9}{0.64} = 10.78$ s

Moving onto the speed distribution, the space mean speed needs to be calculated. Computing the harmonic mean of the speed measurements, this turns out to be 14.49 s. The variation logarithm is then calculated using equation 3.3-15:

$$\omega(l) = \bar{\upsilon}_s(l)/\bar{\upsilon}(l) = 14.49/15.21 = 0.953$$

$$T_{\log}(l) = \ln\left(1 + \frac{[\omega(l)]^2 \cdot \mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^2}\right) = \ln\left(1 + \frac{0.953^2 \cdot 3.11^2}{15.21^2}\right) = 0.037$$

Similarly though, attempting to calculate $T_{\log}(l)$ through equation 3.3-16:

$$T_{\log}(l) = \ln\left(1 + \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^2} \cdot \left(1 - \frac{\mathrm{var}[\upsilon(l)]}{[\bar{\upsilon}(l)]^2}\right)^2\right) = \ln\left(1 + \frac{3.11^2}{15.21^2} \cdot \left(1 - \frac{3.11^2}{15.21^2}\right)^2\right) = 0.037$$

which indicates that the approximation of equation 3.3-15 by equation 3.3-16 is fairly accurate.

Thus:

$$r_L(l) = \exp\left[½ \cdot T_{\log}(l) - z_{\alpha/2} \cdot \sqrt{T_{\log}(l)}\right] = \exp\left[½ \cdot 0.037 - 1.96 \cdot \sqrt{0.037}\right] = 0.7$$

$$r_E(l) = \exp\left[-½\cdot T_{log}(l) - z_{\alpha/2}\cdot\sqrt{T_{log}(l)}\right] = \exp\left[-½\cdot 0.037 - 1.96\cdot\sqrt{0.037}\right] = 0.67$$

Calculating the mean travel time as length over space mean speed, $\bar{t}(l) = \lambda(l)/\bar{\upsilon}_s(l)$ = 100/14.49 = 6.9 s (as obtained from the travel time distribution), and hence:

$t(l)_{0.025} = \bar{t}(l)\cdot r_E(l) = 6.9\cdot 0.67 = 4.62$ s and $t(l)_{0.975} = \dfrac{\bar{t}(l)}{r_L(l)} = \dfrac{6.9}{0.7} = 9.86$ s

It can be thus seen, that there is a discrepancy in the index values calculated from the two distributions, and consequently, a discrepancy in the travel time interval computed. More specifically, the direct calculation from the speed distribution tends to overestimate the actual reliability index values, resulting in an underestimation of the maximum and minimum possible travel time values. This is due to the Taylor series approximation that was carried out in order to express the travel time variation logarithm as a function of speed. Namely, as a first order expansion was performed, the higher order terms of the series were ignored, resulting thus in an underestimation of the travel time variation logarithm. Including higher order terms in the Taylor series expansion would improve the accuracy of the result, however, the present approximation can be accepted, because without significantly underestimating the actual values (difference of less than 1 second for a mean of 7 seconds), it has the very important advantage of simplicity.

Example 3: Calculation of time-dependent link reliability indices

Using the data from the first example of Section 2.5.3, link l of length $\lambda(l)$ = 2500 m is considered. Also, the time points τ_0 = 0 s, τ_1 = 300 s, τ_2 = 600 s and τ_3 = 900 s are given, specifying three 5-minute time intervals $\Delta\tau_0 = [\tau_0, \tau_1)$, $\Delta\tau_1 = [\tau_1, \tau_2)$ and $\Delta\tau_2 = [\tau_2, \tau_3)$. The space-mean speed on the link for each time interval is $\upsilon_0(l)$ = 55 km/h (15.27 m/s), $\upsilon_1(l)$ = 10 km/h (2.78 m/s) and $\upsilon_2(l)$ = 45 km/h (12.5 m/s), while the corresponding earliness and lateness values at the 90% confidence level are $r_{E0}(l)$ = 0.58 and $r_{L0}(l)$ = 0.64, $r_{E1}(l)$ = 0.35 and $r_{L1}(l)$ = 0.47, and $r_{E2}(l)$ = 0.55 and $r_{L2}(l)$ = 0.61. Setting an entry time of a vehicle into the link at $\tau_{ent}(l)$ = 211 s, the reliability indices of the link are sought.

First the variation logarithm for each time interval is derived using equation 3.3-24. The values are $T_{log0}(l)$ = 0.091, $T_{log1}(l)$ = 0.298 and $T_{log2}(l)$ = 0.109. As the entry time into the link is given, the forward calculation procedure described in Section 3.3.6 is used.

The values of $\lambda(l) / \upsilon_0(l)$ and $\tau_1 - \tau_{ent}(l)$ are calculated:

$$\lambda(l) / \upsilon_0(l) = 2500 / 15.27 = 163.72 \text{ s}$$
$$\tau_1 - \tau_{ent}(l) = 300 - 211 = 89 \text{ s}$$

Since $\lambda(l) / \upsilon_0(l) > \tau_1 - \tau_{ent}(l)$, the procedure goes into the next iteration and the values of λ_0 and T_0 need to be computed:

$$\lambda_0 = \upsilon_0(l)\,(\tau_1 - \tau_{ent}(l)) = 15.27 \cdot 89 = 1359.03 \text{ m}$$
$$T_0 = \sqrt{\exp[T_{log0}(l)] - 1} = \sqrt{\exp[0.091] - 1} = 0.309$$

The values of $(\lambda(l) - \lambda_0) / \upsilon_1(l)$ and $\tau_2 - \tau_1$ are calculated next:

$$(\lambda(l) - \lambda_0) / \upsilon_1(l) = (2500 - 1359.03) / 2.78 = 410.42 \text{ s}$$
$$\tau_2 - \tau_1 = 600 - 300 = 300 \text{ s}$$

Since $(\lambda(l) - \lambda_0) / \upsilon_1(l) > \tau_2 - \tau_1$, the procedure goes into the next iteration and the values of λ_1 and T_1 need to be computed:

$$\lambda_1 = \lambda_0 + \upsilon_0(l)\,(\tau_2 - \tau_1) = 1359.03 + 2.78 \cdot 300 = 2193.03 \text{ m}$$
$$T_1 = T_0 + \sqrt{\exp[T_{log1}(l)] - 1} = 0.309 + \sqrt{\exp[0.298] - 1} = 0.898$$

The values of $(\lambda(l) - \lambda_1) / \upsilon_2(l)$ and $\tau_3 - \tau_2$ are calculated next:

$$(\lambda(l) - \lambda_1) / \upsilon_2(l) = (2500 - 2193.03) / 12.5 = 24.56 \text{ s}$$
$$\tau_3 - \tau_2 = 900 - 600 = 300 \text{ s}$$

Since $(\lambda(l) - \lambda_1) / \upsilon_2(l) < \tau_3 - \tau_2$, the procedure stops and the total travel time t(l) and variation logarithm $T_{log}(l)$ are computed, using the appropriate formulas:

$\tau_{ex}(l) = \tau_2 + (\lambda(l) - \lambda_1) / \upsilon_2(l) = 600 + 24.56 = 624.56$ s

and

$t(l) = \tau_{ex}(l) - \tau_{ent}(l) = 624.56 - 211 = 413.56$ s $= 6.89$ min

$$T_{log}(l) = \ln\left(1 + \left(\frac{1}{3}\cdot\left(T_1 + \sqrt{\exp[T_{log2}(l)] - 1}\right)\right)^2\right) =$$

$$= \ln\left(1 + \left(\frac{1}{3}\cdot\left(0.898 + \sqrt{\exp[0.109] - 1}\right)\right)^2\right) = 0.157$$

Hence the reliability indices are calculated using equations 3.3-6 and 3.3-7:

$$r_L(l) = \exp\left[½\cdot T_{log}(l) - z_{\alpha/2}\cdot\sqrt{T_{log}(l)}\right] = \exp\left[½\cdot 0.157 - 1.65\cdot\sqrt{0.157}\right] = 0.56$$

$$r_E(l) = \exp\left[-½\cdot T_{log}(l) - z_{\alpha/2}\cdot\sqrt{T_{log}(l)}\right] = \exp\left[-½\cdot 0.157 - 1.65\cdot\sqrt{0.157}\right] = 0.48$$

The travel time interval for the link is therefore:

$t(l)_{0.05} = t(l)\cdot r_E(l) = 6.89\cdot 0.48 = 3.31$ min and $t(l)_{0.95} = \dfrac{t(l)}{r_L(l)} = \dfrac{6.89}{0.56} = 12.31$ min

3.4 Concluding remarks

In this chapter, the topic of travel time variability and reliability was initially reviewed and a new reliability measure, to be used in in-vehicle navigation, was developed. Initially, the importance of travel time variability to travellers was identified and it was derived from the literature that a function considering the reliability of links and routes in in-vehicle navigation would be extremely useful. Then, it was determined from the literature that travel times follow a log-normal distribution; however, from reviewing existing reliability measures adopted by various studies in the past, it was found that most of them either assume that travel times are normally distributed or are difficult to understand by the travellers, hence pointing out the need of a new measure, to be implemented in in-vehicle navigation.

A new reliability measure, consisting of two reliability indices based on the log-normal distribution, was defined, meeting a number of imposed requirements, such as comprehensibility by the traveller and ‘dimensionlessness’. It was then expressed in terms of the distribution of speeds, such that it could be possible to evaluate it when the given data consists of speed measurements rather than travel time measurements. A method to compute the reliability of a path consisting of a series of links and junction movements was derived, the inverse procedure to obtain statistical values from reliability index values was deduced, and a procedure for calculating the reliability of links under time-dependence was developed. Numerical examples were provided for demonstration purposes.

In the next chapter, the reliability measure defined here will be implemented and incorporated into a newly developed algorithm for dynamic in-vehicle navigation.

CHAPTER 4

Reliable dynamic in-vehicle navigation methodology

4.1 Introduction

Having introduced route finding in road networks and the new reliability measure, this chapter deals with the actual reliable dynamic in-vehicle navigation algorithm, which is intended to make use of the two concepts so as to provide good route recommendations to the driver. A good route in this context is not necessarily the fastest or the shortest one, but one which satisfies ex post a set of acceptability criteria, as these are preset by the driver. The value of a navigation system depends on the quality of the route guidance it provides, such that a good route suggestion characterises the system as precise and reliable, whereas a bad route suggestion (e.g. one guiding the driver onto congested roads or containing infeasible turns) has a negative effect on its appreciation. A system, constantly outputting unacceptable routes to the user may significantly lose in market value, as it may develop a bad reputation. Thus, in order to devise a strategy for an advanced system, the algorithm employed should be carefully designed so as to output good route recommendations.

Considering the requirements that a new advanced in-vehicle navigation algorithm should have, the most important is the inclusion of travel time uncertainty and reliability, which is the primary objective of this study. Making use of the concepts described in the previous chapter, an algorithm addressing the minimisation of travel time uncertainty in route guidance is required, such that the user of the system can get route recommendations, while at the same time minimising the amount of unpredictable delays encountered as much as possible. It should be noted, that only autonomous navigation systems are considered here, i.e. systems that do not have access to real-time traffic data apart from the limited information on traffic incidents provided by the so-called 'Traffic Message Channel' (TMC). The algorithm developed should, based solely on past travel time data, be able to identify possible 'black spots', i.e. areas with a high probability of encountering unpredictable delays and avoid them as much as possible. Furthermore, the system should also be able to perform this procedure en-route, i.e. should re-route in case actual delays are reported on the original route recommendation.

Apart from the incorporation of travel time uncertainty, however, there are also many additional requirements that an advanced algorithm for in-vehicle navigation should meet. The inclusion of time-dependence is one of them, i.e. the consideration of the fact that travel time is not constant on all roads of the road network but varies with time. It is essential for an advanced navigation system that its path finding algorithm considers the travel time experienced at the time of travel and computes routes accordingly, as introduced in Chapter 2.

Furthermore, the adoption of a multi-path strategy, so as to offer the driver a set of routes to choose from rather than a single route to follow, is a useful feature. Apart from the fact that the driver is able to choose a route best fitting his/her individual preferences, multi-routing approaches have also been proven to be superior, as they reduce the negative effects that a system may have by guiding a large number of vehicles along the same route, thus causing congestion (Lee, 1994). Finally, the application of constraints and multiple objectives, so as to ensure that the route guidance offered satisfies a number of acceptability requirements, is a feature of high practical value. An algorithm that is able to optimise in terms of additional objectives rather than just travel time or travel distance is therefore needed.

This chapter presents the in-vehicle navigation algorithm developed in this study, which combines the path finding algorithms of Chapter 2 with the reliability measure defined in Chapter 3. The chapter is structured as follows; Section 4.2 gives a comprehensive literature review of advanced route finding topics, such as path finding under uncertainty, multiple routes algorithms, multiple-objective path finding algorithms and previous dynamic route guidance strategies, existing in the literature. Following that, Section 4.3 presents Chen's link penalty method, which is the basis of the approach developed in this study, and the new reliable dynamic in-vehicle navigation algorithm (RDIN) is described and formulated subsequently. Finally, the topic of re-routing is examined in Section 4.4, including a brief review of its background, followed by the description and formulation of the new re-routing algorithm developed here (RDIN-R) which is an extension of RDIN.

4.2 Background

This section reports on previous research on topics, relevant to the route guidance algorithms described in this chapter, and more specifically relevant to the requirements that this needs to meet, as mentioned in Section 4.1. Firstly, route finding under uncertainty is reviewed, covering algorithms attempting to solve the so-called 'stochastic time-dependent least-time path problem' and the so-called 'robust shortest path' problem. Then, a review of algorithms computing more than one path is given, followed by solutions to the problems of routing using multiple objectives and routing subject to constraints. A brief review of existing route guidance approaches and algorithms in the literature is presented next, before moving onto the presentation of the algorithm implemented in this study, in subsequent sections.

4.2.1 Routing under uncertainty

A field that is relevant to transport applications, and more specifically to in-vehicle navigation, is the problem of route finding under uncertainty conditions with respect to link travel times, which is usually the case in congested urban road networks. Three main approaches are mentioned here: the stochastic time-dependent least-time path, the robust shortest path and Chen's link penalty method. In the first one, travel times on links are represented as random variables, while in the second one, travel times are represented by ranges, that is by an upper and a lower bound. Finally, in the third approach, which forms the basis of the RDIN algorithm described in Section 4.3, links are characterised as reliable or unreliable.

Stochastic time-dependent shortest path

The uncertainty exhibited by link travel times in congested urban road networks can be expressed by a model, where travel times are represented as random variables, with probability distribution functions that vary with time. The result is a stochastic time-dependent network, in which the shortest (least-time) path is to be determined. Unlike deterministic networks, in which a single shortest path can be found between an origin and a destination, in stochastic time-dependent networks several paths may each have a high probability of being the shortest one under a certain realisation of link travel times. Therefore, the solution of the stochastic time-dependent least-time path problem is usually a set of Pareto-optimal (also referred to as non-dominated) paths.

Some researchers attempt to solve the problem by finding the least expected travel time path (Cerulli et al, 2001). This approach is suitable for time-invariant stochastic networks (i.e. where the distributions are constant), as the least expected travel time path can be determined simply by setting each random link weight to its expected value and solving the equivalent deterministic problem; however, it cannot be applied to real networks, where link weights are time-dependent. An approach proposed by Hall (1986) involves combining the branch-and-bound and K-shortest paths methods, in order to determine a priori least expected travel time paths. Additionally, a dynamic programming approach with an adaptive decision rule is presented in the same work, according to which an optimal successor node is defined as a function of the arrival time at the node.

A significant contribution to the field has been made by Miller-Hooks and Mahmassani (1998; 2000; 2003) and by Miller-Hooks (2001). Namely, considering stochastic time-varying transport networks, a methodology for finding a set of a priori least possible time paths is developed (Miller-Hooks and Mahmassani, 1998). As a continuation, a procedure for determining a priori least expected time paths is presented (Miller-Hooks and Mahmassani, 2000), as well as a method for computing adaptive least expected time paths (Miller-Hooks, 2001), for the case where the driver is able to make decisions and change his/her route during the journey. Additionally, measures of assessing the Pareto-optimality of paths are defined (Miller-Hooks and Mahmassani, 2003). An algorithm incorporating delays due to traffic lights has also been provided (Yang and Miller-Hooks, 2004).

Miller-Hooks' approach has been tested and proved to be accurate on a real road network by Polenta and Hartley (2003). Polenta (2005) further extends Miller-Hooks' approach, by developing algorithms for the stochastic time-dependent least-time path problem, based upon different optimisation criteria and accommodating different levels of travel time uncertainty.

The main drawback of the stochastic time-dependent least-time path approach is that it may cause very large computation times (Meyer, 2003), a phenomenon frequently referred to as the "curse of dimensionality". Therefore, heuristic algorithms seeking a trade-off between precision and efficiency have been developed (Fu and Rilett, 1998; Polenta, 2005). The lack of efficiency and tractability (the processing of complete travel time distributions cannot be efficiently manipulated by processors of on-board devices, whose computed power is limited to 300-600 MHz, as stated in Section 1.3.1) however make such approaches very impractical to implement in in-vehicle navigation.

Robust shortest path

The robust shortest path approach is another method, which can be used to find the fastest path under uncertainty. As opposed to the stochastic time-dependent shortest path approach, link weights are not reflected by random variables with probability distribution functions varying with time, but are represented by ranges. Namely, it is assumed that the travel time on a link can take any value between a given lower and a given upper bound. The approach proceeds by investigating all possible routes under all possible scenarios, a scenario being defined as a specific realisation of link weights (travel times) in the entire network. A path selection procedure follows, according to the so-called 'relative robustness criterion' (also referred to as 'minmax robust deviation criterion' or 'minmax regret criterion'), as defined by Kouvelis and Yu (1997). This is why the approach is also referred to as the 'relative robustness shortest path' approach.

Considering every path on the network, the critical scenario for each one is the one in which its total travel time minus the travel time of the fastest path in this scenario (robust deviation) is maximum. By examining all possible paths in the network and obtaining all the corresponding robust deviation values, the best path is the one, whose robust deviation is minimal. In a study by Karasan et al (2002) it is proved that the critical scenario of a path is the one, in which all of its links are set to their upper bounds, while all other links are set to their lower bounds. This means that only a finite number of scenarios need to be considered, and more specifically, the number of scenarios to be examined is the number of possible paths in the network.

However, Karasan et al (2002) also conjecture that the relative robustness shortest path problem is NP-hard, which means that the computation time required for its solution grows exponentially with the size of the problem; this is subsequently proven and confirmed by Zielinski (2004) and by Averbakh and Lebedev (2004), even for specially structured networks (Kasperski and Zielinski, 2006). An initial heuristic solution for solving the problem is suggested by Yu and Yang (1998), while a mixed integer programming formulation and a pre-processing technique involving the elimination of a number of links, in order to reduce computation time is presented later (Karasan et al, 2002). Unfortunately this method is only applicable to layered networks, which are usually found in telecommunications.

An application of the problem and a proposed exact solution algorithm is suggested by Montemanni and Gambardella (2004). The proposed algorithm yields either an exact solution, or a fairly good heuristic solution. An alternative branch and bound algorithm with similar properties developed concurrently is presented next by Montemanni et al (2004). Finally, heuristic solutions using adaptations of Floyd's algorithm and Dijkstra's algorithm in the context of route guidance applications are given by Bell et al (2005).

Chen's link penalty method

A further method to overcome the problem of travel time uncertainty in route finding is presented by Chen et al (Chen et al, 2005a; 2005b; 2006; Chen and Wang, 2007), and involves penalising those links in the network, which exhibit greater uncertainty. This is done by increasing and progressively reducing their weights such that they are avoided but not completely excluded from the path that is returned by the path finding algorithm. This method is suitable for in-vehicle navigation as it is both fairly precise and efficient. As it is the approach adopted in this study as the basis of both the RDIN and RDIN-R algorithms, it is discussed in more detail in Section 4.3.1.

4.2.2 Multiple routes

A multi-routing strategy for in-vehicle navigation systems has been proven to be superior to single path solutions, especially with increasing market penetration (proportion of equipped vehicles in the entire vehicle fleet) (Lee, 1994). More specifically, suggesting only one path to all drivers of equipped vehicles increases the risk of occurrence of the so-called 'congestion feedback' phenomenon, which results in all vehicles being guided along a single path and consequently causing congestion on a previously uncongested road. Moreover, supplying the driver with more than one route enables him/her to choose the route best matching his/her individual preferences. Therefore, it is beneficial that a route guidance algorithm has the ability of producing more than one path.

A considerable amount of literature concentrates on solving the problem of finding the K-shortest paths in a network. In an early review, Dreyfus (1969) presents the algorithms to that date for determining the second-, third-, fourth-shortest path etc. An important contribution is made two years later by Yen (1971), in whose study it is demonstrated that the second-shortest path is a deviation from the shortest path, the third-shortest path is a deviation from either the shortest or the second-shortest path and so on. Therefore, to find the second-shortest path, the algorithm presented performs multiple subsequent searches, each time excluding one link of the shortest path by setting its weight to infinity, and chooses the path with the lowest cost among the ones obtained. While many other algorithms for finding the K-shortest paths exist in the literature (Climaco and Martins, 1982; Martins, 1984a; Eppstein, 1997; Martins et al, 1999; Jimenez and Marzal, 1999; Liu and Ramakrishnan, 2001; Yang and Chen, 2005), Yen's algorithm is a very popular algorithm for finding the K-shortest paths and is still being used and modified in recent studies, such as the study by Jeong et al (2007), where it is applied in a multi-path strategy for the location-based shortest path problem.

A different approach developed for computing multiple routes involves finding a certain number of disjoint paths, that is, paths that do not share any nodes or links. The advantage of a disjoint path set over the K-shortest paths approach, is that the paths are independent from each other, such that an incident on one of them will not affect the other ones. Nonetheless, the routes computed are not optimal, that is they are not the second-, third-, fourth-shortest etc paths any more. Algorithms achieving this are presented by Tanenbaum (1988) and Torrieri (1992); both approaches follow a similar procedure, which includes computing a path, then removing its links and their intermediate nodes and computing the next path in the reduced network.

Relatively recently, several research studies have been carried out, in order to determine so-called partially disjoint paths in a network. An approach by Park et al (2002) makes use of the so-called 'efficient vector labelling approach' to compute alternative "reasonable" paths in a transport network, where "reasonable" means that each of the paths computed should satisfy some constraints relating to their length and overlapping. This is achieved by ruling out a number of potential candidate paths through pruning of the network and through exclusion of sub-paths not meeting the constraints. By setting the value of the appropriate constraint, one can determine the extent to which overlapping between the alternative paths is desired.

The approach usually adopted for computing partially disjoint paths though, involves applying link penalties to already used links, increasing their weights, so as to reduce their chances of being included in other paths. A strategy developed by Rouphail et al (1995) for providing tourists with alternative routes, involves increasing the weight of every link in the shortest path by 20%, 50% and 100% of its original value, so as to avoid including it in, but not completely excluding it from the next path. Based on this concept, Pu et al (2001) and Chen et al (2005a) (see Section 4.3.1) also present link penalising procedures to compute K partially disjoint paths.

Using any link penalty application procedure yields multiple routes, which are nevertheless not optimal (shortest). However, applying constraints (see next sub-section) can guarantee acceptability by the drivers, which in the case of in-vehicle navigation is satisfactory. Furthermore, an advantage of this concept is that multiple paths are computed without altering the network structure, which is a procedure imposing additional effort on the limited computing power of navigation devices (see Section 1.3.1). Hence, such approaches are suitable for in-vehicle navigation applications and are therefore adopted in this study.

4.2.3 Multiple objectives and constraints

It has been found from empirical studies, that drivers have individual preferences regarding route choice, so that the definition of the best path may vary between individuals (Chen et al, 2001). Criteria such as travel cost, road safety, comfort, driver's familiarity, road characteristics etc, affecting the acceptability of a route by the driver, have a significant impact on route choice and increase the perceived value of a navigation system, which can more accurately reflect them. For example, a system using an algorithm that only optimises the total travel time may lead to some drivers being guided along very long routes (in terms of distance) or simply along routes they dislike, either because these appear to be marginally faster, or because they are part of a multiple-path strategy and are not optimal. Research in this field in terms of the development of algorithms for learning driver preferences is carried out and presented by Park et al (2007b; 2007c) and includes extracting individual preferences from the usage of navigation systems.

The multiple-objective shortest path problem can be mathematically formulated as a multiple criteria optimisation problem. Although optimising with respect to a single criterion is a relatively simple task, the problem becomes much more complex when trying to optimise with respect to different and often conflicting criteria. In earlier studies, Climaco and Martins (1982) and Martins et al (1984b) conjecture that, similarly to the stochastic time-dependent shortest path problem, in most cases no single path optimising all objectives at the same time may be available, and that the optimal solution of the problem is usually a set of Pareto-optimal paths. Solution algorithms outputting Pareto-optimal paths are presented in both studies, as well as in a later study by Skriver and Andersen (2000).

Other algorithms for solving this problem have been developed by numerous researchers, a few examples of which are mentioned here. In a study by Wahle et al (2001) fuzzy set theory is applied to the routing problem so as to optimise according to the individual criteria of the driver. Chen and Yang (2000; 2003) propose an algorithm for a bi-criterion shortest path problem, where the two objectives to be optimised are the total travel time and the number of stops at traffic lights, corresponding to the total delay at junctions. An attempt to compute the 'minimum time and minimum cost' path considering stops at traffic lights is also made by Ahuja et al (2002).

Defining a composite cost function giving appropriate weights to each of the objectives and aiming to minimise the total cost is a way of dealing with multiple-objective routing. In a study by Wang and Zhang (1992), describing a route planning algorithm for navigation systems of robots, a weighted function of travel time and number of turns on a route is used. Similarly, in a study by Blue et al (1997) the route finding algorithm is executed based on a cost function composed of trip travel time and trip complexity. In a different study by Kang et al (2006) addressing the so-called 'concierge service problem' in transport applications, which involves travelling to a number of shops and purchasing a number of products at the lowest possible cost and travel time, a total cost function is defined, containing three types of costs: shoprelated costs (prices of the products, parking costs etc), travel time-related costs (value of time spent on the road) and travel-related costs (petrol, tolls, vehicle wear and tear etc). The objective is to minimise the total cost. The approach is taken forward in a further study by Jeong et al (2007).

A different approach for solving the multi-objective shortest path problem is to optimise with respect to a criterion defined as most dominant and applying the other criteria as constraints, thus solving the so-called ‘constrained shortest path’ problem. Unfortunately, even for the simplest cases of constrained route search, the constrained shortest path problem has been found to be NP-hard (Garey and Johnson, 1979). Past research attempting to solve it mainly relates to the safe transport of hazardous materials. Examples are the work of Erkut and Verter (1998), who attempt to quantify transport risk and incorporate it in a routing model, and the work of Sivakumar and Batta (1994), who present a solution algorithm for the so-called ‘variance-constrained’ shortest path problem.

A further solution algorithm for the constrained shortest path problem is presented by Jahn et al (1999), who, in the context of route guidance, apply constraints to the path computation so as to ensure that drivers will not perceive the route computed as too long. In a more advanced version, the algorithm by Park et al (2002) introduces constraints to ensure that the paths computed will be “reasonable”, i.e. they will not be too long. Finally, Chen et al (Chen et al, 2005a; 2005b; 2006; Chen and Wang, 2007) develop a route guidance strategy, which includes a path-checking procedure, i.e. each path computed is checked against a number of acceptability constraints before being output by the route guidance algorithm (see Section 4.3.1).

4.2.4 Existing in-vehicle navigation approaches and strategies

Having reviewed work on routing under uncertainty, multi-routing and multi-objective and constrained routing, this sub-section introduces some existing dynamic route guidance concepts and describes existing approaches and strategies. Dynamic route guidance can be classified into multi-vehicle dynamic route guidance, where the objective is to minimise the system-wide total travel cost, i.e. the travel cost of all vehicles in the network, or single-vehicle dynamic route guidance, in which the aim is to minimise only the travel cost of a single guided vehicle (Zhao, 1997).

Multi-vehicle route guidance has been the subject of numerous research studies, not only in terms of in-vehicle navigation systems, but also in terms of route guidance communicated to the driver using variable message signs (e.g. Tsavachidis (2000)). Considering in-vehicle route guidance, it has been found that an important factor affecting its quality with respect to the performance of the network is the so-called 'market penetration' (Yang et al, 1993; Yang, 1998; 1999), which expresses the proportion of the vehicles that are equipped with a navigation device. Namely, with increasing market penetration, phenomena such as 'congestion feedback' (Arnott et al, 1991), where a large number of vehicles are guided along the same route thus causing congestion on a previously uncongested road, or 'overreaction' (Ben-Akiva et al, 1991), where the number of drivers responding to traffic information is so high that the traffic situation case used to generate the guidance becomes invalid, may occur.

Three categories of multi-vehicle route guidance strategies can be defined according to the traffic information that the system makes use of: reactive guidance, anticipatory guidance and feedback guidance. In reactive guidance (Deflorio, 2003), historical and current traffic data are used so as to perform an a priori traffic assignment and suggest a route to each participating driver prior to departure. In anticipatory guidance (Crittin and Bierlaire, 2001; 2003; Dong et al, 2006) the traffic assignment is also carried out a priori, however this is done using predicted traffic data rather than historic and current. In feedback guidance (Pavlis and Papageorgiou, 1999) on the other hand, equipped vehicles constantly report their traffic observations back to a centre, such that a real-time traffic assignment can be carried out at all times based on the traffic situation as this has been formed by compliance or non-compliance to the route guidance system by individual drivers.

While there are numerous other research studies tackling various issues of multi-vehicle dynamic route guidance, reviewing them is beyond the scope of this work, as this study is concerned with single-vehicle route guidance and with ways of avoiding delays as much as possible under the assumption that no real-time information, apart from TMC reports on incidents, is available to the system. In any case, any in-car multi-vehicle route guidance approach is difficult to implement, as it requires static or dynamic traffic assignment to be carried out, which is a very demanding task that is constrained by the computational ability of on-board devices. The advantage of multi-vehicle strategies, however, is that they consider the effect of their own recommendations and can therefore prevent congestion feedback and overreaction. In single-vehicle route guidance a detailed calculation of this is not possible; what can be done is to minimise the probability of occurrence of such phenomena by introducing a multi-routing strategy, as is demonstrated by Lee (1994) and adopted in this study.

Regarding previous work on single-vehicle dynamic route guidance algorithms, a number of approaches are worth reviewing here. In a study by Whitsitt and Travis (1996), an algorithm aiming to accelerate the search for a route and at the same time match the individual preferences of the driver, is proposed. This is achieved by employing an artificial intelligence technique called 'case-based reasoning', according to which a database of pre-computed routes is kept, such that when a route is required, the algorithm first searches whether a previously computed route, or part thereof, exists in the database, and only if that is not the case does it proceed to the computation of a new route. The study also presents some encouraging results, demonstrating an advantage in computation time compared to a simple version of Dijkstra's algorithm.

In another study by Ericsson et al (2006), an in-vehicle navigation algorithm is presented, optimising in terms of fuel consumption and aiming to minimise CO_2 emissions. The procedure adopted involves classifying the roads of the network into categories according to attributes such as their surrounding function and their speed limit, and defining so-called 'fuel consumption factors' for each road type. Then, a version of Dijkstra's algorithm is employed, optimising in terms of the fuel consumption factor of a route rather than travel time. The algorithm is field tested in a small network and the results not only indicate a clear fuel saving for the guided vehicles, but also a travel time saving compared to the routes that unguided drivers take. It is acknowledged though, that the findings may only refer to that particular network and the need of carrying out further tests on a larger network is identified.

In a study by Jeong et al (2007), a solution approach to the problem of 'location-based services' in an in-vehicle navigation system is presented. The study describes an algorithm for computing multiple paths considering specific points of interest that these should go through; points of interest are defined according to location-based activities. An example of a problem that this algorithm provides a solution to is the problem of needing to go home, but also needing to buy petrol and three bottles of wine on the way there, at the lowest possible price. The algorithm defines a cost function, according to all the costs that are included in such a trip, such as the price of petrol and wine at various petrol stations and supermarkets en route, but also the costs associated with the trip itself, i.e. vehicle wear and tear, value of travel time etc, and then attempts to find a set of alternative routes minimising the total cost. Nevertheless, while the simulation results reported in the study are fairly encouraging, it should be noted that the approach is based on the assumption that all the required information (range of products at shops, prices of products etc) is available, which is not the case in reality, as such a database does not exist and its creation would be an extremely time-consuming and demanding task, as not only its creation is required, but also its continuous updating.

An in-vehicle navigation algorithm is further presented by Park et al (2007a). Using a centralised system architecture in Korea, the route guidance requests of the driver are transmitted to a server, which computes a route for the driver in question and transmits it back to him/her in the vehicle. A multi-routing strategy is adopted in this approach by making use of a previously developed multiple paths finding algorithm (Park et al, 2002). The main characteristic of the procedure is that it uses forecast travel times, based on data collected from inductive loop detectors and the neural network forecasting procedure described by Park and Rilett (1998) and Park et al (1999). The results obtained from a pilot test carried out were quite promising and a full-scale implementation is currently being prepared. Nonetheless, the drawbacks of the approach are the lack of provision for time-dependence of travel times, which is considered to be unimportant, and the fact that the approach only covers the major roads of the Korean road network, as no data for minor roads is available.

Other work on in-vehicle route guidance includes the algorithm presented by Fu (2001), which aims to reflect real-time information more accurately, by not computing an entire route in advance and supplying it to the driver, but by forming it progressively in parts instead. The procedure, termed ‘closed-loop shortest path’ involves identifying only the immediate next link at each step, rather than the whole path to the destination, so as to account for the future availability of travel time information on individual links. The approach is based on the assumption that travel time on links is a random variable following a probability distribution with a certain mean and standard deviation. By making use of this at each step, the expected travel time to the destination can be estimated.

Considering route guidance approaches incorporating travel time uncertainty and variability, the works of Sen et al (2001), Polenta (2005) and Lu et al (2005) are worth describing. Starting from the first one, Sen et al (2001) formulate a multi-objective model which aims at optimising in terms of both the mean and variance of the travel time distribution, thus taking into account travel time variability. The algorithm proposed consequently solves the stochastic time-dependent least time path problem, which is reviewed in Section 4.2.1. In Polenta's study (Polenta, 2005), the same stochastic time-dependent shortest path concept is adopted and a number of algorithms for its solution, using different optimisation criteria with respect to in-vehicle dynamic route guidance, are developed. Seeking a trade-off between travel time and uncertainty, the algorithms are tested on the major road network of Nottingham, using travel time data obtained from SCOOT detectors. Finally, in the study by Lu et al (2005), a robust optimisation model is proposed for route guidance, in which travel time is treated as a random variable, whose distribution is obtained from historical data. By defining the so-called 'time at risk', the study aims at considering the distribution's skewness and kurtosis; using this, a multi-objective model is formulated, allowing a trade-off between travel time and time at risk.

One of the key in-vehicle dynamic route guidance approaches is Chen's link penalty method (Chen et al, 2005a; 2005b; 2006; Chen and Wang, 2007), which is the approach on which this study is based. This is presented in detail the next section.

4.2.5 Summary

From the review of the topics relevant to the dynamic route guidance approach developed in this study, it can be concluded that:

- Incorporating travel time uncertainty in route finding algorithms can be an extremely tedious task. While a significant number of studies have focussed on solving the stochastic time-dependent shortest path problem, the solutions derived are not easily implementable to route guidance due to their complexity. The robust shortest path approach is more promising, however it can also be computationally demanding and does not have the tractability and efficiency advantages of Chen's link penalty method, which is chosen and described in more detail in Section 4.3.1.

- When computing multiple routes, the most efficient approach is to apply link penalties to avoid already used links in subsequent computations, without ruling them out completely though, thus obtaining partially disjoint paths. Acceptability of the routes by the drivers can be ensured by the application of constraints on them.

- Solving the multiple-objective shortest path problem involves computationally demanding processes. Applying constraints is a more straightforward and acceptable approach.

- In-vehicle route guidance approaches have been extensively researched for the purpose of their integration in a system-wide multi-vehicle optimisation strategy, involving performing static or dynamic traffic assignment, and thus having heavy computational requirements, which are generally not available in on-board navigation devices. Existing single-vehicle route guidance approaches, on the other hand, are rather complex and inefficient procedures. An efficient and tractable algorithm is thus needed, such as Chen's link penalty method (Section 4.3.1).

4.3 The reliable dynamic in-vehicle navigation (RDIN) algorithm

This section presents the autonomous in-vehicle navigation algorithm developed in this study, termed reliable dynamic in-vehicle navigation (RDIN) algorithm. This is based on a previously developed method by Chen et al (Chen et al, 2005a; 2005b; 2006; Chen and Wang, 2007), whose aim is to incorporate travel time uncertainty in route guidance. This study takes this approach forward by first identifying its shortcomings and then modifying it accordingly. The concepts presented in Chapters 2 and 3 are employed here, that is the time-dependent forward and reverse A* algorithm and the new reliability measure. At first, a description of Chen's method is given, followed by the proposed improvements to it in this study and its mathematical formulation.

4.3.1 Chen's link penalty method

Chen's link penalty method (Chen et al, 2005a; 2005b; 2006; Chen and Wang, 2007) has recently been developed as an efficient route plan for in-vehicle navigation systems, which, under the provision of only minimal real-time traffic information, prevents travel time uncertainty from affecting the driver's trip as much as possible. Using only basic principles, such as the measure of reliability defined by Bell and Iida (1997) (see Section 3.2.5) and the A* algorithm (Hart et al, 1968) (see Chapter 2), the method attempts to compute a set of routes for the driver, which will prove themselves ex post to have been good. A good route is defined as one, which, while not necessarily being the fastest or the shortest, meets some pre-defined driver acceptability requirements, but most importantly, entails little travel time uncertainty, such that it is unlikely to encounter unpredictable delays along it.

As was reported in Section 3.2.5, the reliability measure defined by Bell and Iida is expressed as the probability of not encountering congestion, under the assumption that a link's or route's traffic status is binary: 'congested' or 'uncongested'. In order to define the congested state of a link, it is assumed in Chen's link penalty method, that a link is congested if its travel time is longer than a pre-set threshold value, which depends on the particular link. Links having a high probability of this occurring are thus specified as 'unreliable', as opposed to the ones on which this is unlikely to occur, which are termed 'reliable'. The main idea of the method is to compute routes, which avoid unreliable links as much as possible, while at the same time being acceptable to the driver.

The main characteristic of the algorithm is the use of link penalties, which are applied to unreliable links so as to discourage their inclusion in a route. This is achieved by increasing their weight, which in this case is their travel time, by a value large enough to force the route finding algorithm to avoid them and find an alternative route, even if this is not optimal. The link penalties also have an additional function, as they enable the computation of multiple routes. As mentioned in Section 4.2.2, an efficient multi-routing strategy is to prevent the alternative paths computed from being similar and attempt to make them partially disjoint. Applying link penalties and hence increasing the weights of links already included in a previously computed route prevents the link from being included in the computation of further alternative routes. The penalty for each link is inversely proportional to its reliability value, such that more unreliable links get higher penalties than less unreliable ones.

However, avoiding specific links may cause the route finding algorithm to compute completely illogical routes and divert the driver along very long detours in order to avoid a road classified as unreliable or a road that is included in another route. This is where the acceptability of the routes computed is introduced. Namely, in order to ensure that the routes computed are still acceptable to the driver, acceptability criteria are applied as constraints to the route finding algorithm, with which the routes computed by it should comply.

The constraints employed include a maximum path travel time constraint, a minimum path reliability constraint and a maximum number of computed routes constraint. In order to determine the threshold travel time value, above which a route is considered unacceptable, the fastest path p_0 from the origin to the destination is calculated; if its travel time is $T(p_0)$, the path travel time $T(p_i)$ of any path p_i should be $T(p_i)<\beta T(p_0)$, where β is the 'travel time permission parameter' ($\beta>1$). The minimum reliability threshold, on the other hand, is a preset value, a path having a reliability value smaller than which, is considered unacceptable. It should be noted, that the reliability of a path is defined as the product of the reliability values of the links forming the path, assuming statistical independence between them (though this is generally not the case in reality). Finally, the maximum number of computed routes constraint is introduced, so as to ensure that the algorithm will terminate, if an infinite number of acceptable routes are available.

While endeavouring to exclude unreliable and used links from the route search, it may well be the case that following the link penalty application procedure the path found does not meet the constraints applied, which means that some of the excluded links will have to be re-included in the search. In this situation the path is discarded and penalties are re-applied to the penalised links, this time though lower. Using this technique, the least unreliable links are re-included in the search and a new path search is carried out with the new penalties; if the path found does still not meet the constraints, penalties are further reduced and the same procedure is run again.

The link penalty application procedure in Chen's link penalty method occurs as follows: in the first iteration, the weight w_i of link i with reliability r_i is equal to its travel time t_i. In subsequent iterations though, where the link is penalised if it is unreliable, $w_i' = t_i + \Delta w_i$, where

$$\Delta w_i = \alpha^m (1-r_i)^q W_0 \qquad (4.3\text{-}1)$$

with $0 < \alpha < 1$, m = number of iterations, q = 0 if m = 0 otherwise q = 1, and W_0 = a value large enough to achieve link exclusion (suggested as $W_0 = \gamma T(p_0)$), where γ is a parameter with $1.5 \leq \gamma \leq 3$). Hence, as the number of iterations increases the value of α decreases and the penalty applied is therefore reduced.

The steps involved in Chen's link penalty method are thus as follows: an initial run of the A* algorithm is run, such that the fastest route is computed. Then, unreliable links are penalised through equation (4.3-1) and a new run of the A* algorithm is carried out, using the new link travel time values. It should be noted here, that although the increased travel times are used to guide the search procedure, the real travel times (prior to the increase) are used to calculate the travel time of the resulting route. If the route computed does not meet the constraints imposed, it is discarded, the penalties are reduced proportionally to their reliability values and a further A* run is conducted. If the resulting route meets the constraints, it is added to the alternative path set, which is to be output to the driver, and its links are penalised for further A* runs. The algorithm terminates when the required number of acceptable routes have been computed.

An interesting feature of Chen's link penalty method is the fact that for the first run of the A* algorithm, where the fastest path is computed, the algorithm runs in reverse order, that is from the destination to the origin. The advantage of running the A* from the destination to the origin is that after the first run, the computed cost from the destination to every visited node can be used as an estimate for subsequent forward A* runs. The search space of the algorithm is thus significantly reduced in subsequent runs and consequently this is reflected in the computation time.

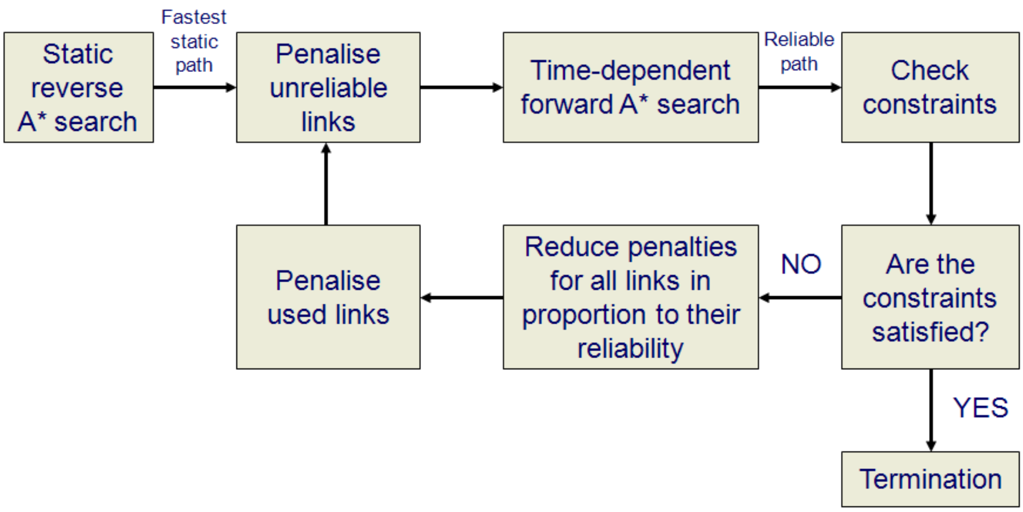


Figure 4.3-1: Chen's link penalty method

Chen's link penalty method also accounts for time-dependence in path finding (Chen et al, 2006). Making use of the technique by Sung et al (2000), termed 'flow speed model' (see Section 2.2.3), a time-dependent version of the A* is derived for the forward runs of the algorithm. Its application to the reverse runs, however, is more complex, as the arrival time at the destination is not known at the start of the search and the arrival time at each link is also not known; hence it is not possible to calculate the experienced travel time by a vehicle on any link, and as a consequence, static rather than time-dependent travel time values are used for the reverse A* runs.

A flowchart representation of Chen's link penalty method is shown in Figure 4.3-1.

4.3.2 Description of the RDIN algorithm

While Chen's link penalty method is a very efficient and precise approach to incorporating uncertainty in in-vehicle navigation, adopting a multiple routing strategy and ensuring driver acceptability, it has a number of shortcomings that need to be fixed so as to become implementable in an actual in-vehicle route guidance system. This is why a number of modifications are made in this study, such that, based on the original Chen's link penalty method, the new RDIN algorithm is derived, whose objective is to compute and suggest to the driver a set of equivalently reliable acceptable routes.

Measure of reliability

The first problem that can be identified is the use of the reliability measure defined by Bell and Iida. Namely, as is discussed in Chapter 3, this measure is a very good index of the reliability of an entire transport network; however, it is not suitable in route guidance, as the binary assumption of the traffic status on a link (congested or uncongested) is not enough to describe to the driver the situation that he/she may encounter. The driver is more interested in the amount of delay that he/she may encounter rather than just in the probability of encountering a delay of unknown duration, so that he/she can make a decision on whether to follow a route or not.

Furthermore, in the calculation of reliability of an entire route statistical independence between link travel times is assumed, which, as empirically shown by Rakha et al (2006), results in great overestimation of the actual reliability of the route. This is acknowledged in Chen's link penalty method and the concept of link failure dependence is proposed, according to which relationships between links are derived, such that an increase in travel time on one link affects travel time on a number of other links, either positively (i.e. their travel times also increase) or even negatively (i.e. their travel times decrease). Nonetheless, no methodology on how to derive these relationships is suggested, the only solution proposed being microscopic traffic simulation. Additionally, the measure employed has the disadvantage that the reliability of a route is dependent on the number of links forming the route, such that if a single link is split into two, the joint reliability of the two resulting links will not be equal to the reliability of the two links treated separately.

To address these issues, the reliability measure defined in Chapter 3 is used in the RDIN algorithm, which accounts for the fairly plausible assumption that travel times are log-normally distributed, considering their extreme values given a specific confidence level by introducing lateness and earliness indices. These indices can be simply converted to actual possible delay or time gain values, such that a time window can be given to the driver as concerns his/her arrival at the desired destination. Also, the assumption of statistical independence in the route reliability calculation is avoided using the method derived in Section 3.3.4, which empirically compensates for the losses caused by the presence of statistical dependence. The two reliability indices suggested in that section have the additional advantage of being independent from the length of the link or route in question.

In order to incorporate the new reliability measure into the algorithm, a modification to the penalty function needs to be made, since there are now two indices rather than one. The same concept of link penalty application is used, such that links that have lateness or earliness or both reliability indices below some pre-set thresholds, have their travel times increased in route search algorithm runs. The new link penalty function for a link l with earliness and lateness reliability indices $r_E(l)$ and $r_L(l)$ becomes thus:

$$\Delta t(l) = \alpha^{m}\left(1 - r_E(l) \cdot r_L(l)\right)^{q} W_0 \qquad (4.3\text{-}2)$$

where, as in Chen's link penalty method, $0 < \alpha < 1$, m = number of iterations, $q = 0$ if $m = 0$ otherwise $q = 1$, and $W_0 =$ a value large enough to achieve link exclusion ($W_0 = \gamma T(p_0)$, where γ is a parameter with $1.5 \leq \gamma \leq 3$, depending on how large one wants W_0 to be). Hence, in the first iteration of the algorithm the weight of link l is $t(l)$; in subsequent iterations though, the weight of the link becomes $t'(l) = t(l) + \Delta t(l)$.

Constraints

A further issue needing to be addressed in the implementation of Chen's link penalty method is the nature and number of the constraints used, as the existing constraints need to be modified and additional ones are introduced. The maximum route travel time constraint is kept as it is and along the same lines, the maximum route length constraint is introduced. The minimum reliability constraint on the other hand is replaced by two new constraints, which are the minimum path earliness and the minimum path lateness constraints. Additionally, a maximum path overlapping constraint is applied, so as to ensure that the alternative paths will be adequately disjoint. Finally, the maximum number of computed routes constraint is kept as in Chen's method.

Starting from the maximum route travel time constraint, already existing in Chen's model, it has to be ensured that the alternative routes computed are not too long, such that they can be considered as alternatives to the fastest route by the driver. Hence, as mentioned in Section 4.3.1, the travel time $T(p_i)$ of an acceptable route p_i must not be greater than a value $T_{max} = \beta T(p_0)$, where β is the travel time permission parameter and $T(p_0)$ is the travel time of the fastest path p_0. The value of β depends mainly on the travel time of the fastest route and on how much extra travel time drivers are willing to accept. For example, if $T(p_0)$ = 10 minutes, a value of β = 2 would mean that a route twice as long as the fastest route (20 minutes) would be acceptable. However, if $T(p_0)$ = 30 minutes, such a value would result in routes up to one hour being acceptable, which is clearly not realistic. Values of β of 1.1 or 1.2 might be applicable in that case.

It is possible that even though a route may be acceptable to the driver in terms of travel time, it is not be acceptable in terms of length. Namely, many drivers may not be willing to be guided along very long detours in order to take advantage of small travel time savings. This is why the maximum route length constraint is introduced, using the same concept as in the maximum route travel time constraint. More specifically, for an alternative route p_i to be considered acceptable, its length $\Lambda(p_i)$ must not be longer than a value $\Lambda_{max} = \zeta\Lambda(p_0)$, where ζ is the length permission parameter and $\Lambda(p_0)$ is the length of the fastest path. The range of ζ is the same as the range of β, i.e. it depends on the length of the fastest route. For example, if $\Lambda(p_0)$ = 5 km, then ζ = 2 would make a 10km-long route acceptable; however, if $\Lambda(p_0)$ = 30km, ζ = 2 would mean that 60 km-long routes would be acceptable, which is unrealistic. As in the case of β, ζ values of 1.1 or 1.2 would be more appropriate.

Replacing the minimum reliability constraint in the original Chen's link penalty method, two new constraints are introduced, namely the minimum earliness and minimum lateness constraints. The reason of their existence is that, although unreliable links are excluded as much as possible from the route search, it is possible that no acceptable path is found during the first iterations, which means that the penalties applied to the links would be reduced so as to progressively re-include the least unreliable among the previously excluded links. In that case and in the absence of the appropriate constraints, there would not be any check on whether the resulting route is reliable or not, and as such, a route with a low earliness or lateness index would be regarded as acceptable, as it would be satisfying the imposed acceptability constraints. Since the key objective of the RDIN approach is to ensure that the routes computed are reliable, the introduction of a minimum earliness and a minimum lateness threshold value, R_{Emin} and R_{Lmin}, is necessary, such that for the reliability indices of an acceptable route p_i, it is $R_E(p_i) > R_{Emin}$ and $R_L(p_i) > R_{Lmin}$. It is suggested in this study, following parameter fine-tuning (Chapter 5), that $0.4 < R_{Emin} < 0.6$ and $0.45 < R_{Lmin} < 0.65$, depending on the confidence level used in the calculation of reliability (see Chapter 3).

For the same reason that it cannot be guaranteed that excluding unreliable links will result in a reliable route unless a minimum route reliability constraint is present, it cannot be guaranteed either, that penalising links already included in an alternative route will result in an adequately disjoint path set. Also, it has been concluded from observation, that great path overlapping is likely to occur to the extent that it is possible that two routes share all of their links but one; although this would classify them as partially disjoint, these paths have obviously none of the advantages of this property. This is why the maximum path overlapping constraint is introduced.

Considering two paths p_i and p_j of lengths $\Lambda(p_i)$ and $\Lambda(p_j)$, the total length of the shared links between the two paths is $\Lambda(p_i \cap p_j)$, while the total lengths of the non-shared links of path p_i

and p_j are $\Lambda'(p_i) = \Lambda(p_i) - \Lambda(p_i \cap p_j)$ and $\Lambda'(p_j) = \Lambda(p_j) - \Lambda(p_i \cap p_j)$ respectively. It should be noted that $\Lambda(\varnothing) = 0$. The path overlapping ratio $\varepsilon(p_i,p_j)$, defined as

$$\varepsilon(p_i,p_j) = \frac{\Lambda(p_i \cap p_j)}{\sqrt{\Lambda'(p_i) \cdot \Lambda'(p_j)}} \tag{4.3-3}$$

is a dimensionless parameter. As can be derived, values of $\varepsilon(p_i,p_j)$ close to 0 would indicate little overlapping between two paths, whereas the larger $\varepsilon(p_i,p_j)$ is, the more the two paths overlap. Considering extreme cases, if the paths are completely disjoint, they do not share any links, meaning that $\Lambda(p_i \cap p_j) = 0$ and therefore $\varepsilon(p_i,p_j) = 0$. On the other hand, if the paths completely overlap (i.e. they share all of their links), then at least one of $\Lambda'(p_i)$ and $\Lambda'(p_j)$ is 0, meaning that $\varepsilon(p_i,p_j) = \infty$.

The respective constraint that is introduced is that every newly computed route should satisfy a maximum path overlapping ratio constraint value, when compared with all the other computed routes so far. In other words, the largest $\varepsilon(p_i,p_j)$ value for path i when compared with any other already computed path j should be lower than a preset threshold ε_{max}. The value of ε_{max} depends on the extent that path overlapping is desirable. Threshold values of 2 – 2.5, as obtained from parameter fine-tuning (Chapter 5), are acceptable; however, if very little overlapping is allowed, then lower values of ε_{max} are chosen, with a value of 0 resulting in totally disjoint paths.

The last constraint applied is the maximum number of routes that are to be computed by the algorithm. Namely, the number of equivalently reliable alternative routes computed should be less than N_{max}, where a value of $N_{max} = 3$ is suggested. This constraint also acts as a termination condition for the algorithm, as the computation of N_{max} routes indicates that the desired number of alternative reliable routes has been obtained and no more routes are to be sought.

Time-dependence

A further issue that is dealt with in the RDIN algorithm is time-dependence. In Chen's link penalty method, the so-called 'flow speed model' technique is used to reflect time-dependence in the forward A* runs (see Section 2.2.3), however its application to the reverse A* run is more complex, because the arrival time at the destination is not known at the start of the search and because the arrival time at each link is also not known. Since the only data available at the start of the procedure is the departure time from the origin, it is not possible to run the time-dependent A* algorithm backwards, unless there is at least some indication of the arrival time at the destination. This is why static travel times are used in the reverse A* run of the original Chen's link penalty method, thus not considering time-dependence in the fastest path calculation.

A contribution of the RDIN algorithm is the fact that time-dependence is also applied to the backward version of the A* search, using the RA* algorithm described in Chapter 2. Namely, Chen's link penalty method is modified, so that, using the same technique, it becomes possible to derive the time-dependent fastest path from the initial A* run and then compute further partially disjoint time-dependent reliable acceptable paths using subsequent A* runs. For this purpose, the order in which the various A* runs are carried out is reversed.

Unlike the original Chen's link penalty approach, where an initial reverse A* run takes place, in the procedure developed here an initial run of the FA* algorithm (see Sections 2.4.1 and 2.5.1) is carried out, yielding the time-dependent fastest path p_0. From this, taking into account the departure time from the origin, the arrival time (AT) at the destination is obtained. Then, applying the maximum path travel time acceptability criterion, i.e. that any acceptable path's travel time may not be longer than the fastest path's travel time multiplied by the travel time permission parameter β, and adding the maximum acceptable path travel time to the departure time (DT), the latest acceptable arrival time (AAT) at the destination is obtained, that is AAT = DT + $\beta T(p_0)$.

Following that and applying penalties on all unreliable links, the RA* algorithm is run, as presented in Sections 2.4.3 and 2.5.2, starting from the AAT, in order to obtain a reliable path, along with an acceptable departure time (ADT), which is the time point at which one would need to depart from the origin so as to arrive at the destination before the AAT, using the newly computed reliable path. Having obtained the path, its actual travel time is computed, i.e. the travel time when departing from the origin at time DT rather than ADT, along with its AT. The path is then checked against the constraints, in addition to the condition that the ADT must be later than the DT. If the constraints are met, the path is kept; otherwise, link penalties are reduced and a further RA* run is carried out to compute a new path.

It should be noted that by carrying out the initial FA* search, optimality can be guaranteed for the resulting path; this means that the initial path obtained is actually the fastest available path. However, as optimality is not needed when subsequently computing partially disjoint paths since some links are purposely excluded from the search, the heuristic estimates used in the RA* algorithm runs do not necessarily need to underestimate the actual travel time from the origin to any point. The values on the g-labels, representing the actual travel time from the origin to each link, set by the initial FA* search (see Chapter 2), may not underestimate the actual travel time from any point to the origin, however they are very close approximations to it. Thus, it is acceptable to use them as heuristic estimates in the RA* runs, as this reduces computation time.

Other issues

Other issues needing to be improved in the original Chen's link penalty method relate to the inclusion of the fastest route to the set of reliable alternative acceptable paths, in case this meets the imposed constraints, and to the termination conditions of the algorithm, so as to ensure that this will not run for a long time seeking for a route that is only marginally better than the ones obtained or maybe does not even exist.

Regarding the first issue, it is implicitly assumed in Chen's method that the fastest route is by default unreliable and hence unacceptable to the driver, and therefore, following its calculation, it is omitted and a more reliable route is sought by penalising all the unreliable links in the network. However, this assumption is in general wrong, as it is very likely that the fastest route is both reliable and satisfies the driver's acceptability criteria, i.e., that it is a perfectly eligible route to be included in the path set suggested to the driver. Using the original technique, the fastest route is discarded, as it is very rare that after the application of penalties the same route is re-computed, and in the situation described, an acceptable route is omitted.

In order to prevent this from happening, a path checking procedure is added immediately after the initial A* run, so as to check whether the fastest route computed meets the constraints. If it does, it is directly added to the path set suggested to the driver and the algorithm proceeds as normal with the link penalisation step, seeking the second alternative route. If it does not, then it is discarded and the algorithm seeks for a more reliable route, as originally planned.

Considering the termination condition of the algorithm, this is defined as the stage when N_{max} routes have been computed, where N_{max} is the desired number of alternative routes. Nevertheless, it is possible that such a number of alternative acceptable routes between the origin and destination do not exist in the network. This means that the algorithm in this situation will keep on searching for a route that does not exist and will therefore enter an infinite loop. An additional termination condition is therefore required.

A sensible criterion to indicate that the algorithm has searched "enough" and that it should terminate and just output the routes it has computed so far, is the situation where a route already included in the path set is re-computed. This would mean that despite having penalised the links already included in a computed and accepted path, the algorithm has not found another acceptable path and the same path has been found. No provision is made for this in Chen's method, so an additional path check is added to the RDIN algorithm, such that for every path computed, it is checked that it is not the same as a previously computed one. This can be immediately identified by the occurrence of a path overlapping ratio of ∞.

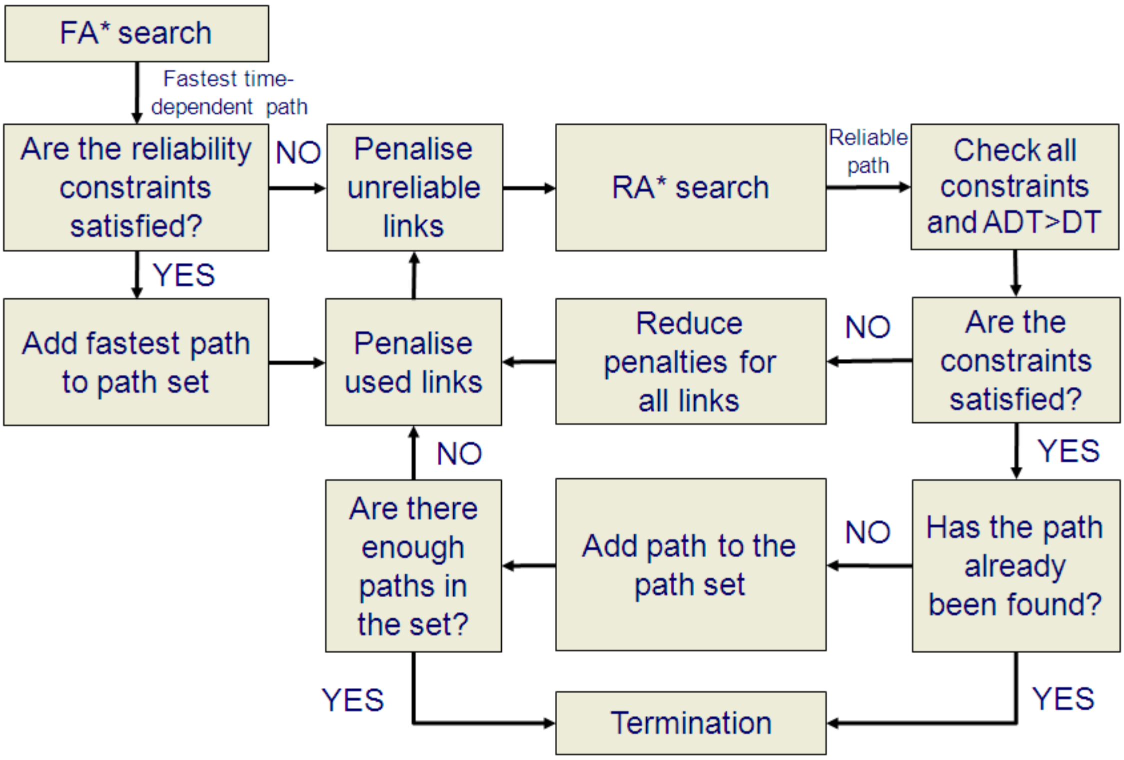


Figure 4.3-2: The RDIN algorithm

Finally, in order to ensure that even in the case where the above termination condition can only be reached after a very large number of iterations, a maximum number of iterations is imposed on the algorithm. Basically, to prevent very long running times from occurring (certainly not desirable in in-vehicle navigation systems), the algorithm terminates after 100 iterations and outputs the routes that it has calculated thus far. In the case where no reliable routes have been found, the fastest route is output and the driver is informed that no acceptable reliable routes have been found and that only the fastest route is available to him/her.

A flowchart representation of the RDIN algorithm is shown in Figure 4.3-2.

4.3.3 Formulation of the RDIN algorithm

Following the description of the concepts involved in the RDIN algorithm, this sub-section sets out to present a mathematical formulation. The notation used is given first, followed by the presentation of the procedure. The notation used in previous chapters also applies here, however variables are re-defined for the purpose of the reader's ease.

Notation

Consider a network G, consisting of a set of nodes N, a set of directed links V and a set of movements M, specifying the allowed turning movements between links. Movements are represented by their start and end links, such that movement (a,b) connects links a and b.

o:	The origin link
d:	The destination link
l_x:	Part x of link l, $x=s$ for start, $x=e$ for end
$t(l)$:	The travel time of link l, calculated using the methods of Section 2.5
$\lambda(l)$:	The length of link l
$r_E(l)$:	The earliness index of link l, calculated using the method of Section 3.3.6
$r_L(l)$:	The lateness index of link l, calculated using the method of Section 3.3.6
r_{Emin}:	Link earliness index threshold for defining a link as 'earliness-unreliable'
r_{Lmin}:	Link lateness index threshold for defining a link as 'lateness-unreliable'

$\delta(a,b)$:	The delay of movement (a,b)
$CL_s \subset V$:	Closed-start list of the RA* algorithm
$CL_e \subset V$:	Closed-end list of the RA* algorithm
i:	Paths counter
p_i:	Path i
$L(p_i)$:	The list of links in path p_i
$T(p_i)$:	The travel time of path p_i
$\Lambda(p_i)$:	The length of path p_i
$R_E(p_i)$:	The earliness reliability index of path p_i
$R_L(p_i)$:	The lateness reliability index of path p_i
$\varepsilon(p_i,p_j)$:	Path overlapping ratio of paths p_i and p_j
T_{max}:	Path travel time threshold
β:	Travel time permission parameter
Λ_{max}:	Path length threshold
ζ:	Length permission parameter
R_{Emin}:	Path earliness threshold
R_{Lmin}:	Path earliness threshold
ε_{max}:	Maximum path overlapping ratio threshold
N_{max}:	Maximum number of paths
DT:	The departure time
AAT:	The acceptable arrival time
$AT(p_i)$:	The arrival time of path p_i
$ADT(p_i)$:	The acceptable departure time for path p_i
m:	Iterations counter
PS:	List of i computed paths

Procedure

Algorithm RDIN

Step 0 (Initialisation): FA* run for o and d → Fastest path p_0 with $L(p_0)$, $T(p_0)$, $\Lambda(p_0)$, $R_E(p_0)$, $R_L(p_0)$, $AT(p_0)$.

$T_{max} = \beta T(p_0)$, $\Lambda_{max} = \zeta\Lambda(p_0)$, $AAT = DT + T_{max}$.

$\forall$ $l_x \in CL_x$ set $h(l_x)=g(l_x)$. $\forall$ $l_x \notin CL_x$ assign $h(l_x)$.

Set $W_0 = \gamma T(p_0)$, where $1.5 \leq \gamma \leq 3$.

Set $m = 0$.

If $R_E(p_0) \geq R_{Emin}$ and $R_L(p_0) \geq R_{Lmin}$

$p_1 = p_0$, $PS = \{p_1\}$, $i = 2$.

Else

Set $i = 1$, $PS = \varnothing$.

Step 1 (Link penalty application): $\forall$ $l \in V$

If $r_E(l) < r_{Emin}$ or $r_L(l) < r_{Lmin}$ or $l \in PS$:

$t'(l) = t(l) + \alpha^m \left(1 - r_E(l) \cdot r_L(l)\right)^q W_0$, $0 < \alpha < 1$,

$q = 0$ for $m = 0$, $q = 1$ otherwise.

Else

$t'(l) = t(l)$.

$\forall$ $(a,b) \in M$

If $r_E(a,b) < r_{Emin}$ or $r_L(a,b) < r_{Lmin}$ or $a,b \in PS$:

$\delta'(a,b) = \delta(a,b) + \alpha^m \left(1 - r_E(a,b) \cdot r_L(a,b)\right)^q W_0$,

$0 < \alpha < 1$, $q = 0$ for $m = 0$, $q = 1$ otherwise.

Else

$\delta'(a,b) = \delta(a,b)$.

$m = m + 1$.

Step 2 (Reliable route calculation): RA* run using $t'(l)$ and $\delta'(a,b)$ for path searching, starting from the AAT_s → Path p_i, with $L(p_i)$, $\Lambda(p_i)$, $ADT(p_i)$.

If $ADT(p_i) < DT$

If $m < 100$

Go back to Step 1.

Else

Go to Step 4.

Else

Calculate $T(p_i)$, $R_E(p_i)$, $R_L(p_i)$, $AT(p_i)$ based on

$L(p_i)$, starting from the DT, using the method of Section 2.5.1.

Step 3 (Check constraints): If $p_i \in PS$

Go to Step 4.

Else

$$\varepsilon = \max_{p_j \in PS}\left(\varepsilon(p_i,p_j)\right) = \max_{p_j \in PS}\left(\frac{\Lambda(p_i \cap p_j)}{\sqrt{\Lambda'(p_i) \cdot \Lambda'(p_j)}}\right).$$

If $T(p_i) < T_{max}$, $\Lambda(p_i) < \Lambda_{max}$, $R_E(p_i) > R_{Emin}$, $R_L(p_i) > R_{Lmin}$ and $\varepsilon < \varepsilon_{max}$

$PS = PS + \{p_i\}$. $i = i + 1$.

If $i \leq N_{max}$

Go back to Step 1.

Else

Go to Step 4.

Else

Go back to Step 1.

Step 4 (Termination): Output PS

4.4 The reliable re-routing (RDIN-R) algorithm

In an in-vehicle navigation system it is often the case that after having selected a route and started driving along it, the driver needs to re-route, either because he/she has simply made a mistake in following the directions provided by the system, or because a traffic incident is reported on the route currently being followed, such that a number of links become unusable and create delays. In this case an algorithm re-computing a path from the vehicle's current position to the driver's destination is required, and as it needs to output the new path as soon as possible since the driver is awaiting instructions whilst en route, it needs to be very efficient. Re-routing is a time-critical procedure because it needs to be run on the spot and output a solution as quickly as possible.

This section complements the RDIN algorithm described in the previous section, by presenting RDIN-R, an algorithm for re-routing in the RDIN scheme. Initially a brief review of existing re-routing algorithms in the literature is given, including the provision made in Chen's link penalty method. Following that, the RDIN-R algorithm is described and formulated.

4.4.1 Background on re-routing algorithms

While much research has been conducted on routing, as was shown in Sections 2.2 and 4.2, the amount of literature on re-routing is considerably smaller. As reported by Kim and Jung (2002), two re-routing techniques have been used in practice in the case where a part of the original route becomes unusable due to an incident: either a complete path re-computation from scratch or a re-computation of the path between the two ends of the unusable part, such that a diversion is created which re-joins the original route as close as possible "downstream" of the incident. The disadvantage of these techniques is that they can become highly inefficient, both in terms of computation times and in terms of the result obtained.

As re-routing is a time-critical procedure, existing literature is devoted to minimising the computation time by making use of previously computed information as much as possible. In particular the A* algorithm is employed as the most efficient and accurate path finding algorithm. Examples include the work of Stentz (1994; 1995), whose D* algorithm dynamically generates optimal paths for a robot operating with a sensor by re-planning locally and is therefore more efficient when the change in the network situation (incident) occurs close to the current position of the robot, and the study by Kim and Jung (2002), which presents a method employing previously computed shortest path information to compute an approximate path without having to re-compute it from scratch. A further example is the work of Koenig et al (2004), whose 'lifelong planning A*' also has the ability of re-using data from previous searches but can become greatly inefficient if the incident is close to the start of the trip, as it may have to re-compute most of the original route.

The above algorithms are advantageous in terms of efficiency, however they do not consider the driver's acceptability criteria and may thus route them along unreliable or unacceptable routes. Chen's link penalty method accounts for re-routing (Chen and Bell, 2005) considering reliability and driver acceptability, using the method described in Section 4.3.1. Namely, if an incident occurs on one of the alternative routes computed by the algorithm, the corresponding link is to be avoided and has therefore its reliability value set to zero. A link penalisation step is then applied, where the travel times of unreliable links are increased so to prevent them from being included in the route computation, according to equation 4.3-1. A route from the vehicle's current position to the destination is then computed using the forward A* algorithm, making use of the data obtained during the initial reverse A* run. The resulting partial route is checked against the driver acceptability constraints and if it does not meet them, it is discarded and a new one is sought. The incident link is always avoided, even at stages where the link penalties have been significantly reduced, as its reliability value, which has been set to zero, results in the penalty function (expression 4.3-1) to be always equal to a large value.

4.4.2 Description of the RDIN-R algorithm

Chen's re-routing method nicely fits with the RDIN framework described in the previous section; nevertheless it is only an initial suggestion, and as such, it has some unclear issues that need to be resolved. First of all, the method concentrates entirely on the case where re-routing is initiated by the occurrence of a traffic incident along the route followed by the driver and no consideration is made with respect to the case where the driver accidentally (or intentionally) deviates from it. Then, similarly to the routing algorithm, it is assumed that an acceptable route always exists and no provision is made for the case where no alternative acceptable route can be found, in which case the algorithm should terminate by keeping the original route. Also, the acceptability constraints used in Chen's routing algorithm are also used intact in re-routing, which is nonetheless not appropriate as not a complete route is sought, but only a partial route. Especially in the case of minimum reliability, the measure used in Chen's method is dependent on the length of the route, and therefore the reliability of a partial route will be greater, since less links are involved. An adaptation of the constraints should thus take place.

Taking Chen's suggested method forward, the RDIN-R algorithm is presented here, designed to be implemented together with the RDIN algorithm. Chen's method suggests to run the re-route searching algorithm (i.e. the A*) in the same direction as the routing algorithm, i.e. forwards. As in the RDIN algorithm the reliable route search step implements the RA* algorithm, this means that RA* should also be used in re-routing. However, the main problem that arises is that RA* requires the acceptable arrival time (AAT), defined in the RDIN algorithm, as benchmark, which cannot be employed in the re-routing process because it is possible that the delays experienced by the vehicle so far result in all possible re-routes exceeding it. On the other hand, the only benchmark available is the time at which the re-routing algorithm is initiated, which is termed re-routing departure time (DT-R). Consequently, only a forward search can be carried out, meaning that the FA* algorithm has to be used.

The fact that the FA* run of the RDIN-R algorithm is preceded by a series of RA* runs in the framework of the RDIN algorithm can be useful in order to follow the same concept as in RDIN and make use of previously computed information. Namely, using the minimum g-label, assigned to every link part during the series of RA* runs and representing the actual travel time from that link part to the destination, as an h-label (estimate of the travel time to the destination) in the FA* run of RDIN-R, provides the algorithm an adequately informed environment, thus contributing to meeting RDIN-R's very tight computation speed requirements. Though the heuristic estimates used by RDIN-R's FA* search are not guaranteed to underestimate the actual travel time and hence ensure optimality, they will be close estimates to it, thus providing adequately accurate solutions. Therefore, a task that is added to RDIN's step 2 (RA* run) is the storage of the lowest g-value of every link part, so that it can be later used in RDIN-R's FA* run.

When it comes to the link penalty application process, a similar concept as in Chen's method is used and the incident links (if any) have their lateness reliability index values set to zero (worst-case scenario, as it indicates that the links are severely blocked and will be for a considerable amount of time). Equation 4.3-2 is then used for the penalty application. It should be noted, that in contrast to RDIN, only unreliable links (i.e. links having $r_E < r_{Emin}$ and $r_L < r_{Lmin}$) are penalised and used links are not. The reason for this is that RDIN-R is not seeking for alternative partially-disjoint routes, but only for a reliable acceptable route from the vehicle's current position to the destination and therefore links lying on other alternative routes do not need to be avoided.

Considering the constraints imposed on the part-routes computed by RDIN-R, these are in concept similar to the ones applied in RDIN. Starting from the maximum travel time constraint and bearing in mind that in re-routing the driver is more interested in finding a way of getting to his/her destination as smoothly as possible (by avoiding the traffic incident, if there is one, or by finding his/her way quickly after being "lost") than ensuring that this route is not too long, a higher travel time permission parameter than in RDIN can be employed. Thus, the re-routing travel time permission parameter β_R is introduced, where $\beta_R \geq \beta$. This is combined with the travel time $T(p_S^{c,d})$ from the current position link of the vehicle c to the destination d on the original selected path by the driver p_S, such that the maximum allowed travel time of the re-route $T(p_R^{c,d})$ is $T_{max\text{-}R} = \beta_R T(p_S^{c,d})$.

In the same way and using the same assumption, the maximum path length constraint is defined using the re-routing length permission parameter ζ_R, where $\zeta_R \geq \zeta$. Combining it with the length $\Lambda(p_S^{c,d})$ of the part $p_S^{c,d}$ of the original selected route p_S from the current position link c to the destination d, the maximum allowed length of the re-route $\Lambda(p_R^{c,d})$ is $\Lambda_{max\text{-}R} = \zeta_R \Lambda(p_S^{c,d})$. On the other hand, as the measure or reliability introduced in Chapter 3 and used in the RDIN algorithm is independent of the length of the route and the number of links it is formed by, the two minimum reliability constraints can be transferred intact from the RDIN algorithm, as they also relate to part-routes. Hence, the re-routing path earliness and lateness thresholds R_{Emin} and R_{Lmin} used in the RDIN-R algorithm are the same as in the RDIN algorithm.

Finally, regarding termination of the algorithm if no acceptable re-route path can be found, the criterion is the maximum number of iterations. Namely, if the algorithm exceeds 100 iterations, then it can be concluded that no re-route satisfying the imposed acceptability constraints can be found. In that case, if re-routing was activated by an incident reported on the selected route so as to avoid the affected area, a message is returned that no acceptable re-route can be found and that the only route is to follow the original route and go through the affected area. If on the other hand re-routing occurs due to the driver missing a turn and no acceptable path can be found, then a fastest path calculation is carried out from the current point c to the destination d, using the FA* algorithm and the resulting route, not bearing any acceptability requirements, is returned to the driver, along with a message stating this. The latter situation, however, is rather unlikely to occur, as it would mean that the driver missed a very important turn, after which the only alternative is a very long route. This can be the case on motorways in rural areas, where one would need to drive to the next exit (which may be quite far), but not in urban environments, where the road network is much denser.

4.4.3 Formulation of the RDIN-R algorithm

After the description of the RDIN-R algorithm, a mathematical formulation is given here. The notation used in the algorithm is listed first, followed by the formulation of the stepwise procedure involved. It should be noted that the notation used in the formulation of the RDIN algorithm in the previous section also applies here and that only additional notation not already defined is listed.

Notation

c:	The current position link
$h_R(l_x)$:	Heuristic estimate of travel time from l to d used in re-routing
p_S:	The selected path by the driver, among the paths in PS
$p_S^{c,d}$:	The part-path on p_s between c and d
$p_R^{c,d}$:	The re-route path from c to d
DT-R:	The re-route departure time
$T_{max\text{-}R}$:	Maximum re-route path travel time threshold
β_R:	Re-route travel time permission parameter
$\Lambda_{max\text{-}R}$:	Maximum re-route path length threshold
ζ_R:	Length permission parameter
$IL \subset V$:	The set of links affected by traffic incidents

Procedure

Algorithm RDIN-R

Step 0a (Pre-requisites): RDIN run for o and d, and run for o and d, and during the series of RA* runs, $\forall\ l_x \in CL_x$, $h_R(l_x) = \min(g(l_x))$ → Alternative path set PS. Selection by the driver of $p_S \in PS$ with $L(p_S)$, $T(p_S)$, $\Lambda(p_S)$, $R_E(p_S)$, $R_L(p_S)$, $AT(p_S)$.

Step 0b (Initialisation): Calculate $p_S^{c,d}$.

$T_{max\text{-}R} = \beta_R T(p_S^{c,d})$, $\Lambda_{max\text{-}R} = \zeta_R \Lambda(p_S^{c,d})$.

Set $W_0 = \gamma T(p_0)$, where $1.5 \leq \gamma \leq 3$.

Set $m = 0$.

$\forall\ l \in IL$, $r_L(l) = 0$.

Step 1 (Link penalty application): $\forall\ l \in V$

If $r_E(l) < r_{Emin}$ or $r_L(l) < r_{Lmin}$:

$$t'(l) = t(l) + \alpha^m\left(1 - r_E(l) \cdot r_L(l)\right)^q W_0,\ 0 < \alpha < 1,$$

$q = 0$ for $m = 0$, $q = 1$ otherwise.

Else

$t'(l) = t(l)$.

$\forall\ (a,b) \in M$

If $r_E(a,b) < r_{Emin}$ or $r_L(a,b) < r_{Lmin}$:

$\delta'(a,b) = \delta(a,b) + \alpha^m \left(1 - r_E(a,b) \cdot r_L(a,b)\right)^q W_0$,

$0 < \alpha < 1$, $q = 0$ for $m = 0$, $q = 1$ otherwise.

Else

$\delta'(a,b) = \delta(a,b)$.

$m = m + 1$.

Step 2 (Re-route calculation): FA* run between c and d using $t'(l)$ and $\delta'(a,b)$ for path searching, starting from the DT-R → Part-path $p_R^{c,d}$, with $L(p_R^{c,d})$, $T(p_R^{c,d})$, $R_E(p_R^{c,d})$, $R_L(p_R^{c,d})$, $\Lambda(p_R^{c,d})$, $AT(p_R^{c,d})$.

Step 3 (Check constraints): If $T(p_R^{c,d}) < T_{max\text{-}R}$, $\Lambda(p_R^{c,d}) < \Lambda_{max\text{-}R}$, $R_E(p_R^{c,d}) > R_{Emin}$, $R_L(p_R^{c,d}) > R_{Lmin}$

Go to Step 4.

Else

If $m < 100$

Go back to Step 1.

Else

$p_R^{c,d} = p_S^{c,d}$. Go to Step 4.

Step 4 (Termination): If $\exists\, l \in \left(IL \cup L(p_S^{c,d})\right)$

Output $p_R^{c,d}$.

Else

If $m < 100$

Output $p_R^{c,d}$.

Else

FA* run between c and d using $t(l)$ and $\delta(a,b)$ → $p_R^{c,d}$. Output $p_R^{c,d}$.

4.5 Concluding remarks

This chapter presented the reliable dynamic in-vehicle navigation methodology developed in this study. At first a review of relevant literature was presented, including topics of routing under uncertainty, multi-routing strategies, multi-objective routing and previous dynamic route guidance schemes. Then, the methodology adopted in this study, based on Chen's link penalty method, was described and the corresponding algorithm, termed RDIN, was formulated. A variation of the RDIN algorithm for re-routing, the RDIN-R algorithm was also presented and formulated.

While the RDIN approach has many innovations regarding its efficiency and accuracy so as to offer better quality route guidance in a more advanced system than state-of-the-art, it still has some weaknesses. For example, a possible improvement of the link penalty application function (equation 4.3-2) is the fact that reducing the penalties by a constant value and consequently re-including previously excluded links may result in a potential more reliable route, satisfying the constraints, to be ignored. To determine whether such a route exists, the link penalties would have to follow a "dither-changing" procedure (i.e. progressively increased and reduced), so as to search for penalties that are as large as possible without infringing the constraints, and to finally converge to the route.

Another possible improvement that can be identified is the fact that the constraints are based on constant values, whether directly (such as the minimum reliability indices) or even indirectly (travel time and length constraints, depending on constant permission parameters). This means that these values have to be changed depending on the road network the algorithm is run on. In order to create a more universal approach, an adaptive procedure examining the network and determining the values of the parameters should be carried out prior to running the RDIN and RDIN-R algorithms. This would also be the case for the threshold values characterising links as reliable or unreliable.

Additionally, a topic that needs further investigation is the inter-link congestion dependence relationships, so as to be included in the RDIN approach. Currently only links characterised as unreliable are penalised in the algorithm; however, the occurrence of congestion on a link will almost certainly affect other links in the network, either positively or negatively. This should be reflected in the RDIN and RDIN-R procedures, such that links dependent on unreliable links are also penalised. Ideally, congestion patterns should be identified and incorporated into the algorithm. Current research in this direction includes the work of Hu et al (2007), who attempt to derive congestion patterns through microscopic simulation, and the work of Hu and Bell (2007), using spatial econometrics to predict traffic incidents.

In order to address such issues, appropriate solution procedures should be incorporated in the RDIN and RDIN-R algorithms, which would however significantly increase their running times, making them very inefficient. Ways of improving the accuracy of the approach while at the same time reducing its effect on efficiency should be sought. However, this is beyond the scope of this study and can be therefore identified as an area of future research.

Having presented the theoretical framework of the in-vehicle navigation approach developed in this study, which includes the basic features of path finding in road networks and the newly developed reliability measure, as well as the RDIN and RDIN-R algorithms, the practical application of the approach is reported next. The next chapter introduces ARIAdNE, a new purpose-developed software tool for the implementation of the algorithms, and reports on the results from two simulation experiments, aiming to demonstrate it and give a preliminary indication of the advantages of the approach, prior to its actual implementation in the field, described in Chapter 6.

CHAPTER 5

Software implementation and simulation experiment

5.1 Introduction

After introducing the RDIN and RDIN-R algorithms in the previous chapter, this chapter describes the actual implementation of the approach in in-vehicle navigation. A new software tool, called ARIAdNE, is developed, providing an application platform for the new RDIN and RDIN-R algorithms, used in both laboratory and field experiments.

ARIAdNE is intended to be run in a Microsoft Windows™ environment. Though it is still prototype software not suitable for use on built-in devices, it needs to be user-friendly and efficient, especially when used in the field, as is the case of the experiments reported in the next chapter. The required functionalities included are, apart from the generation of reliable dynamic route guidance, the creation and modification of road networks, along with their display, the simulation and import of traffic data and the provision of further expansion of the program, so as to incorporate the outcomes of further research work, such as modelling of user preferences and congestion prediction.

In this chapter, ARIAdNE is used in a simulation experiment demonstrating the functionality of the RDIN and RDIN-R algorithms. Time-dependent traffic data is simulated for a road network in Munich, Germany, and route guidance is sought between an origin and a destination. To evaluate the result obtained, the opinion of local traffic experts is asked.

The chapter is structured as follows; ARIAdNE is introduced in the next section and its functions, structure and user interface are described. Then, the test network and the procedures behind the simulation of link speed, junction delay and reliability data are presented, followed by an account of the experimental conduct and of the results obtained.

5.2 Software implementation – ARIAdNE

This section introduces a new software tool, the Adaptive Reliable Imperial Advanced Navigation Engine (ARIAdNE), formerly known as the Imperial College Navigation Software (ICNavS), which implements the RDIN and RDIN-R algorithms presented in the previous chapter and also provides a visual interface for displaying the road network and the route guidance obtained. It is written in the Visual C#.NET object-oriented programming language. A short description of the program's abilities and modes is given first, followed by a presentation of its structure and user interface.

5.2.1 Modes of operation and abilities

ARIAdNE has three modes of operation, named 'Navigation', 'Design' and 'Analysis', each one having its own purposes and functions. 'Navigation' mode entails the main purpose of the software tool, which is its use as an advanced car navigation system making use of the RDIN and RDIN-R algorithms. More specifically, in ARIAdNE's 'Navigation' mode the user can load a previously created road network, input an origin and destination, and obtain a set of alternative equivalently reliable time-dependent routes from the RDIN algorithm. Then, by selecting a route according to his/her own criteria, he/she can obtain driving directions and follow them. Also, the user can simulate a traffic incident along the selected route and activate the RDIN-R algorithm to re-route and find a different route to the destination in real-time.

'Design' mode on the other hand is at a lower level, such that it is a pre-requisite to 'Navigation'. In this mode the user can create a new network from scratch by drawing nodes and links on a background map image and setting allowed junction movements between them, or he/she can modify a previously created network. A network can also be imported into ARIAdNE from a PTV VISUM^{TM} map database. Additionally, the user can superimpose a background map on a network, scale it and save it in one of the program's own formats, .ntw (text delimited) or .nwk (serialised), so that it can be further used in 'Navigation' mode. Traffic data for the links and junction movements of a network can also be simulated in 'Design' mode, as it will be explained in the next section, or it can be simply imported, as is the case of floating vehicle data (see next chapter).

Finally, 'Analysis' includes other useful functions offered by the system and provisions for extensions of the routing algorithms. Examples of those are the filtering of network elements and their appropriate colouring (e.g. the colouring of links with a lateness reliability value lower than a threshold so as to identify black spots); and the generation of sets of partially disjoint routes from a single origin to all possible destinations in a network. The latter will be used in a research project currently under way, aiming at modelling driver preferences so as to create an adaptive in-vehicle navigation system. It should be noted, that the resulting learning algorithm is to be incorporated in ARIAdNE's 'Navigation' mode, such that based on previous route selections, a route recommendation entailing the driver's preferences can be given, following the RDIN run. Nevertheless, this work is not complete at the time of compilation of this thesis.

Other abilities of ARIAdNE concern the support of both left-hand and right-hand traffic networks, enabling thus its use in both the UK, Continental Europe and North America, and also include a series of visual functions aiming at either the manipulation of the network display (e.g. zooming and panning the map), or at reducing the memory usage if a very large network is loaded causing the application to run slower (disabling of background and visual network elements).

5.2.2 Structure

As the programming language used is object-oriented, the structure of the code is based on a hierarchy of classes. Three main categories of classes can be identified in ARIAdNE: network classes (including all classes relating to the creation and storage of the road network), operation classes (including the classes relating to the execution of the algorithms) and visual classes (including the visual controls and everything relating to the display of the program).

Starting from the network structure (Figure 5.2-1), there is a network class for nodes (called 'Nodes'), another one for links (called 'Links') and another one for junction movements (called 'Movements'), each bearing a number of properties that are categorised as 'static' or 'runtime', depending on whether they are modified or not during the execution of the algorithms. Properties relating to network topology and geometry are static, since they are not altered during the execution of the program; on the other hand, properties such as the travel time of a link, computed upon request during the execution of the RDIN and RDIN-R algorithms, are runtime properties.

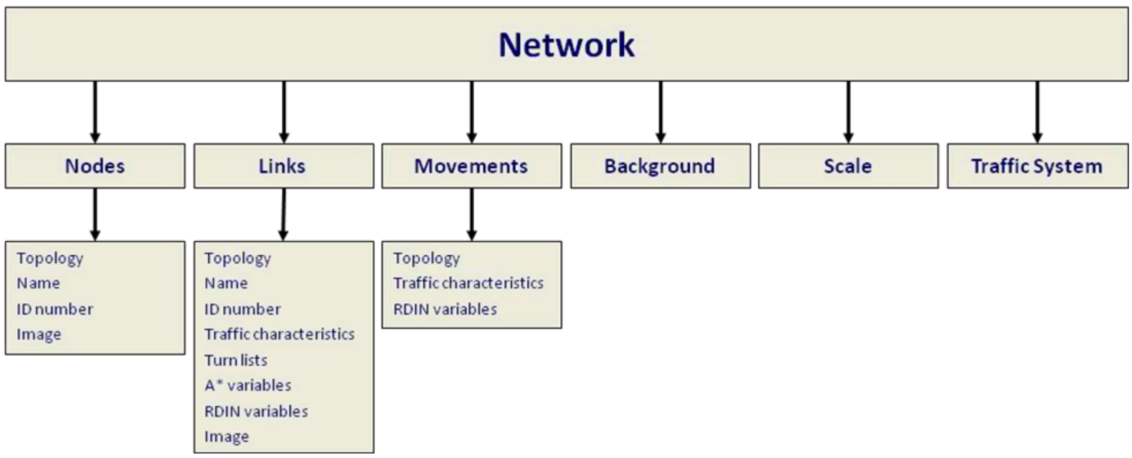


Figure 5.2-1: Network structure of ARIAdNE

'Nodes' (i.e. every node in the network) objects hold the following static properties: name, ID number and topology (X and Y co-ordinates with respect to a given reference point). Their main purpose is to define the start and end locations of 'Links' objects. Also, every 'Nodes' object is associated with the image of a circle representing the node in the visual interface of the software.

On the other hand, 'Links' objects hold the following static properties: name, ID number, topology (start node, end node, road type, length), traffic characteristics (free-flow speed, speed profile, earliness profile and lateness profile) and turn lists (two lists stating the predecessor and successor links of the link along with the corresponding connecting junction movements). Additionally, each 'Links' object holds some runtime properties, which can be grouped into A* variables (twelve labels for the execution of the A* search in the RDIN and RDIN-R algorithms – f, g and h for the start and end of the link, four Boolean variables indicating whether the link is in the open-start, open-end, closed-start or closed-end lists of the A* search and a pointer indicating the link where the search came from in case the link has been visited by the algorithm), and RDIN variables (time-dependent speed, travel time, earliness and lateness, a property holding the updated value of the travel time following the penalty application and a Boolean variable indicating whether an incident is reported on the link). Every 'Links' object is also associated with the image of a line, representing the link in the visual interface of the software.

'Movements' objects hold the following static properties: topology (start-link, end-link) and traffic characteristics (turn type – right, left or straight-on, delay, earliness and lateness profiles). They also possess runtime properties relating to the RDIN algorithm, which are: time-dependent delay, earliness and lateness values, and a property holding the updated value of the delay following the application of penalties.

The elements of the network of each of the three network classes are held in a single object of a class called 'Network', which also has some additional properties, namely a background image, a scale value for the background image and a property indicating whether right-hand or left-hand traffic applies. In each instance of the program there can only be one 'Network' object, as different 'Network' objects represent different networks loaded in the program.

Regarding the operation of ARIAdNE, the following operation classes exist beside 'Network': 'A-star', 'Paths' and 'Main'. Class 'A-star' does not have any properties, but contains all the functions involved in any way in the A* algorithm. Thus, it contains the code for the FA* and RA* algorithms, the functions evaluating time-dependent travel time and reliability, as well as the calculation method of the reliability of a route described in Section 3.3.

Class 'Paths' on the other hand expresses each route computed and has the following properties, all of which are runtime: ID number, list of links forming the route, list of movements connecting the links, travel time, length, earliness, lateness, maximum time gain, maximum delay, expected arrival time, earliest arrival time, latest arrival time, directness (ratio of airline distance to actual distance) and number of right, left and U-turns. A 'Paths' object is returned every time an A* search is completed, and the alternative path set output by the RDIN algorithm is a list of 'Paths' objects.

Finally, class 'Main' is the initiating class; it contains some general variables, such as the alternative route list, as well as some general functions, such as the code for the RDIN and RDIN-R algorithms, the functions for opening and saving an existing or new network file and the functions for importing or simulating traffic data for a loaded network.

Having presented the structure of ARIAdNE, a description of the user interface is given in the next sub-section.

5.2.3 User interface

Describing the user interface of ARIAdNE, a main menu is displayed whenever the program is run, offering a choice of mode to the user: 'Navigation', 'Design' or 'Analysis' (Figure 5.2-2). Depending on which one of the three is chosen by the user, the main window of the application comes up with different functions being available. The main window consists of a canvas, where the road network is displayed, a box on the right hand side, where the properties of any selected element on the canvas are shown (node, link or movement), and buttons enabling the execution of the program's functions.

Figure 5.2-2: ARIAdNE main menu

In 'Navigation' mode (Figure 5.2-3), the user is immediately prompted to load a previously created network so as to be able to run the RDIN algorithm on it. With a network loaded, the user can set an origin and a destination by pressing on the appropriate buttons, set the parameters of the algorithm (travel time and length permission parameters, reliability thresholds etc.) and obtain alternative reliable routes, whose properties are displayed in a list box located to the right of the canvas. By selecting a route in the list box, its links are highlighted on the map. If the route is selected to be followed by the user, driving directions are displayed in another list box, located to the right of the canvas. By selecting a particular row in this latter box, the corresponding manoeuvre is identified on the map by the appearance of a small vehicle.

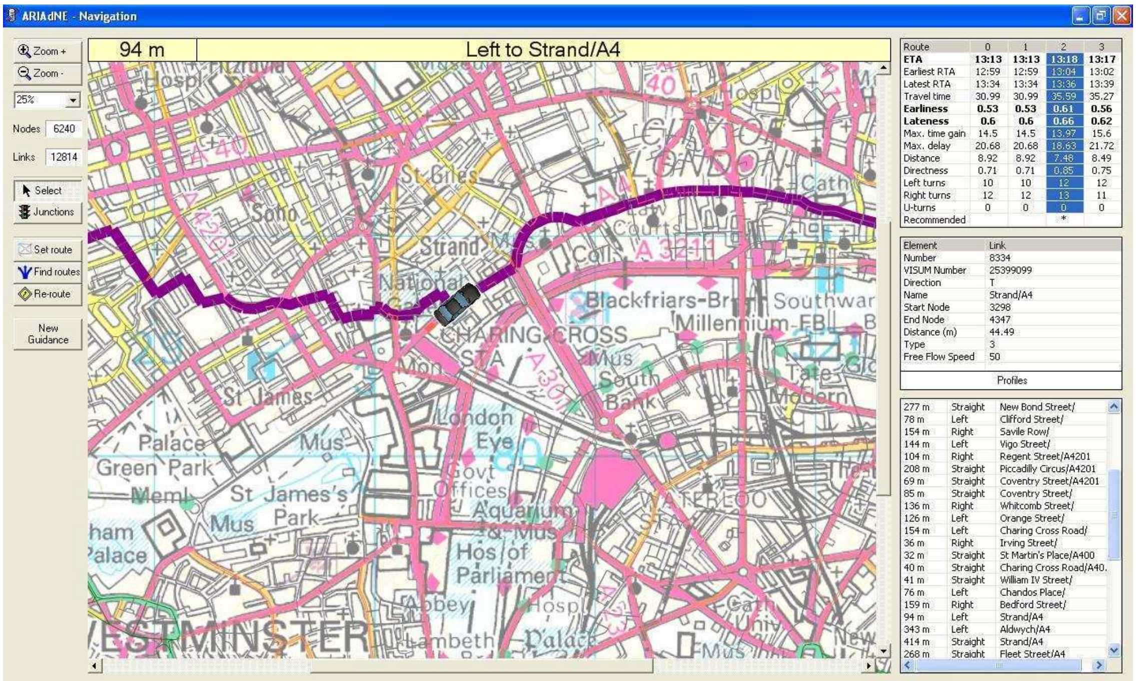


Figure 5.2-3: ARIAdNE user interface in 'Navigation' mode

As different functions are available in 'Design' and 'Analysis' modes different controls are available. For example, the route set and driving directions boxes are not available in 'Design' mode (Figure 5.2-4), and so are the buttons for obtaining route guidance. However, a range of buttons for designing a network are available, such as buttons for the creation of new nodes and links of different road types, and the setting of allowed and banned turns. Also, buttons for loading and saving a network are present, as well as buttons for the import of traffic data on a loaded network. Finally, a menu bar offering a number of additional functions is provided, containing items for setting a background and for simulating traffic data. As in 'Analysis' mode a mixture of the functions of 'Navigation' and 'Design' are available, certain buttons from both modes are displayed.

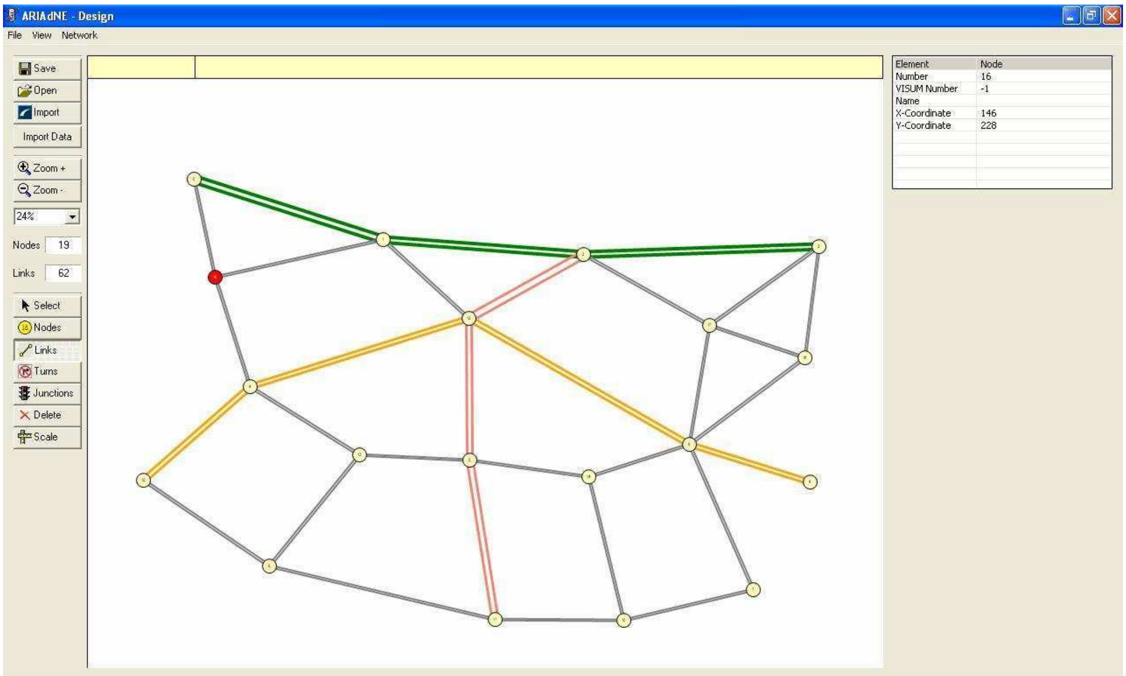


Figure 5.2-4: ARIAdNE user interface in 'Design' mode

Using ARIAdNE, a simulation experiment is carried out so as to demonstrate the RDIN and RDIN-R algorithms, described in the previous chapter, to conduct some preliminary validation and to fine-tune the parameters employed. Prior to reporting on this and the results obtained, the data used is presented, including the test network and the procedures involved in the simulation of traffic data in the next section.

5.3 Test network and traffic data simulation

This section presents the acquisition and simulation of the data required in the experiment reported in the next section. First, the test network is introduced; then, a description of the simulation processes of traffic data is given, and more specifically of link speed profiles, junction delay profiles, and earliness and lateness reliability index profiles for both links and junction movements.

5.3.1 Test network

The test network employed in the simulation experiment of this chapter is the area around the

'Forschungs- und Innovationszentrum' (Research and Innovation Centre) of BMW Group, located in the northern suburbs of the city of Munich in Germany. It is bounded by the A99 motorway to the North, the B11 national road to the East, Hohenzollernstrasse to the South and the B304 / Dachauer Strasse to the West, covering an area approximately 10 km long and 7.5 km wide, and containing 3506 nodes and 7130 links.

Diversity is one of the key characteristics of the network chosen, as this contains many different road types ranging from motorways to minor roads, and includes a wide range of land uses, spreading from residential areas to business districts. It is notable to point out that the main headquarters of BMW Group are located in the centre of the network. Regarding the importance of the roads contained by the network, the A9 motorway is one of the key arteries into and out of the city, connecting its centre with the airport, and so is the B13 national road (Ingolstädter Strasse), while the A99 is Munich's outer ring. The area covered by the network is shown in Figure 5.3-1.

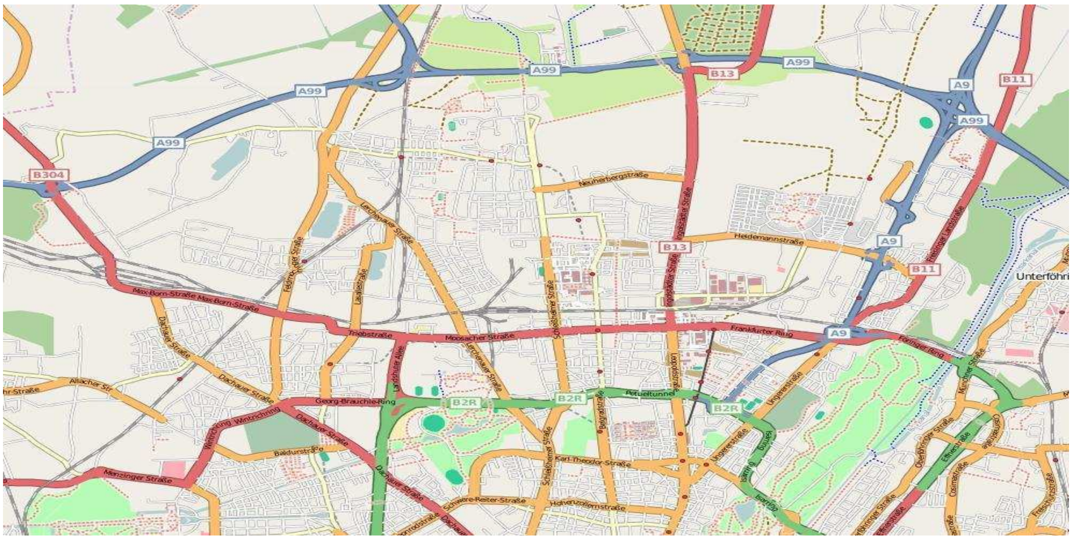


Figure 5.3-1: The Munich test network

(Source: www.openstreetmap.org)

The network is imported into ARIAdNE in PTV VISUM^{TM} format and then saved in the program's own format, .nwk, so as to be able to be re-loaded at any time. Following that, traffic data is simulated for it, according to the procedures described in the next three sub-sections.

5.3.2 Link speed data

Based on the fact that road networks are hierarchical, such that there are different types of road, each having different properties such as traffic flow speeds and number of lanes, daily speed profiles are simulated for each individual link of the test network. Namely, the classification employed by many digital map developers, such as NAVTEQ™, grouping roads into five categories (Motorway, Major A-road, A-road, B-road and Minor road) is adopted. Knowing the speed limit of each road type, it is assumed that vehicles travel at a constant speed at a given time interval of the day, and that the highest value that this speed can take is the free flow speed, set equal to the speed limit.

Table 5.3-1: Speed limits for each road type and speed values used

Road type	Speed limit (km/h)
Motorway	120
Major A-road	70
A-road	50
B-road	50
Minor road	30

The speed limits of the five road types are shown in Table 5.3-1. Naturally, as one would expect, such speed values occur during off-peak times, and the values encountered during peak and inter-peak hours are lower. Splitting the day into 15-minute intervals and distinguishing between weekdays and weekends, simulated coefficients are applied on the free flow speed values according to the time of the day, ranging from 1 during off-peak hours to 0.6 during the morning and afternoon peak of weekdays. A different pattern is used for weekends, where the corresponding factors take values from 1 during off-peak to 0.8 during weekend peak. An exception applies to minor roads, on which a factor of 1 is used during the entire day, whether peak or off-peak, as it is assumed that any traffic fluctuations are not noticeable due to the fact that the traffic flow is low anyway.

Figures 5.3-2 and 5.3-3 show the simulated daily speed profiles for weekdays and weekends corresponding to all five road types, based on the free flow speed values of Table 5.3-1. An interesting feature that should be noted is that during weekday peak hours, it is assumed that minor roads may have faster speeds than A- and B-roads, and during the top of the peak hour they may even exceed Major A-roads. This is likely to occur in reality and an evidence is the fact that professional drivers (such as taxi drivers) sometimes prefer driving along minor roads during peak hours to avoid the congestion on the main roads. Using the techniques described in Section 2.5 in the execution of the FA* and RA* runs in the RDIN and RDIN-R algorithms, time-dependent travel times are calculated and accounted for.

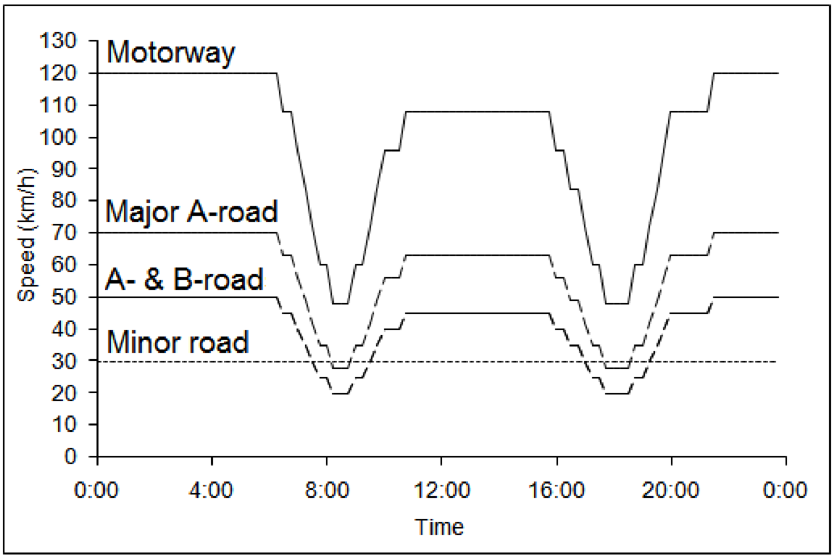


Figure 5.3-2: Simulated link speed profiles for weekdays

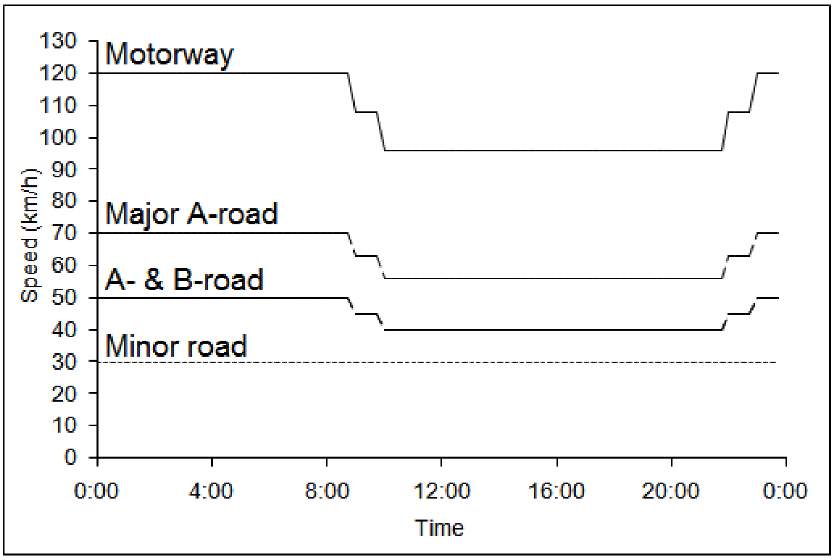


Figure 5.3-3: Simulated link speed profiles for weekends

Having presented the simulation procedure for link speed data, the next sub-section describes how daily profiles for junction delays are simulated.

5.3.3 Junction delay data

Adopting a similar method to the one used for link speed profiles, delay profiles are simulated for junction movements based on the type of the movement (right turn, left turn, straight-on movement, U-turn) and the roads meeting at the junction. Considering right-hand traffic, which is the case in the Munich network, it is taken into account that right turns usually cause lower delays than left turns and straight-on movements, as the vehicle does not come into conflict with other traffic streams, apart from pedestrian ones. In the case of left turns on the other hand, the vehicle often needs to cross the opposing traffic stream, which means that it may have to stop and wait for several seconds for an acceptable gap.

Initially, maximum average delay values are estimated for junction movements corresponding to peak hours. Right turns are assigned a delay of 0; for left turns the estimated delay value assigned is a function of the road types of the start- and end-links of the turn; finally, for straight-on movements the estimated delay depends on the road types of the links that are being crossed by the movement, and more specifically, on the road type of the link with the highest road type crossing the movement in question. U-turns are treated in the same way as left turns.

Table 5.3-2: Maximum simulated delay values for left turns

	Turn ending on link of type…				
		Major A	**A**	**B**	**Minor**
Turn starting from link of type…	**Major A**	10	10	15	15
	A	10	10	15	15
	B	15	10	5	5
	Minor	20	10	5	2

Table 5.3-3: Maximum simulated delay values for straight-on movements

	Movement crossing link of type…				
		Major A	**A**	**B**	**Minor**
Movement starting from link of type…	**Major A**	10	5	0	0
	A	10	10	5	0
	B	15	10	5	0
	Minor	15	10	5	2

Maximum average delay values for left turns and straight-on movements are shown in Tables 5.3-2 and 5.3-3. It should be noted that as no left turns are possible onto or from motorways, any left turn delays involving motorways are neglected and are set to 0. Also, as intersections between motorways are always grade-separated, ensuring an unobstructed traffic flow, delays of straight-on movements involving motorways are also set to 0.

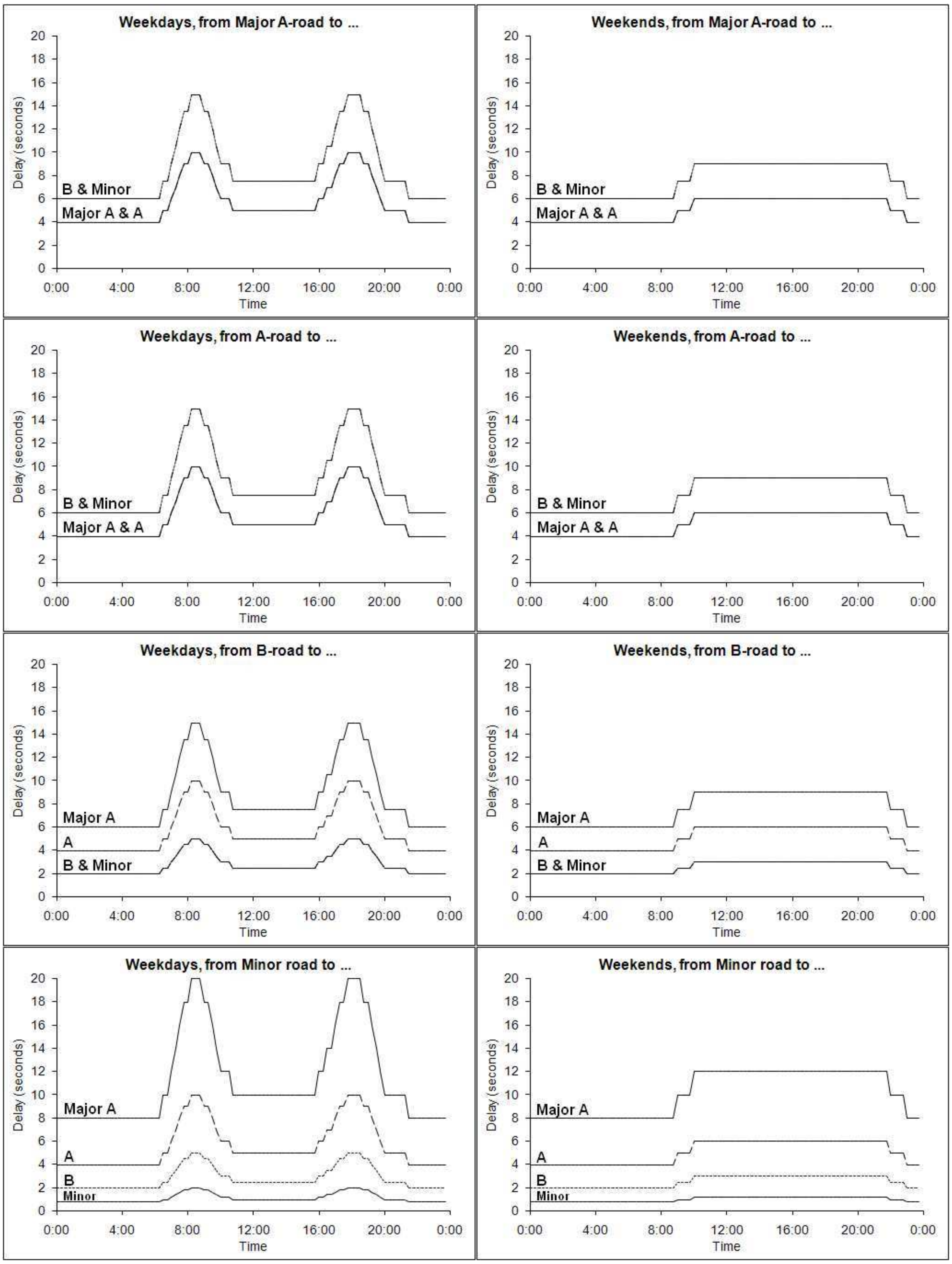


Figure 5.3-4: Simulated left turn delay profiles

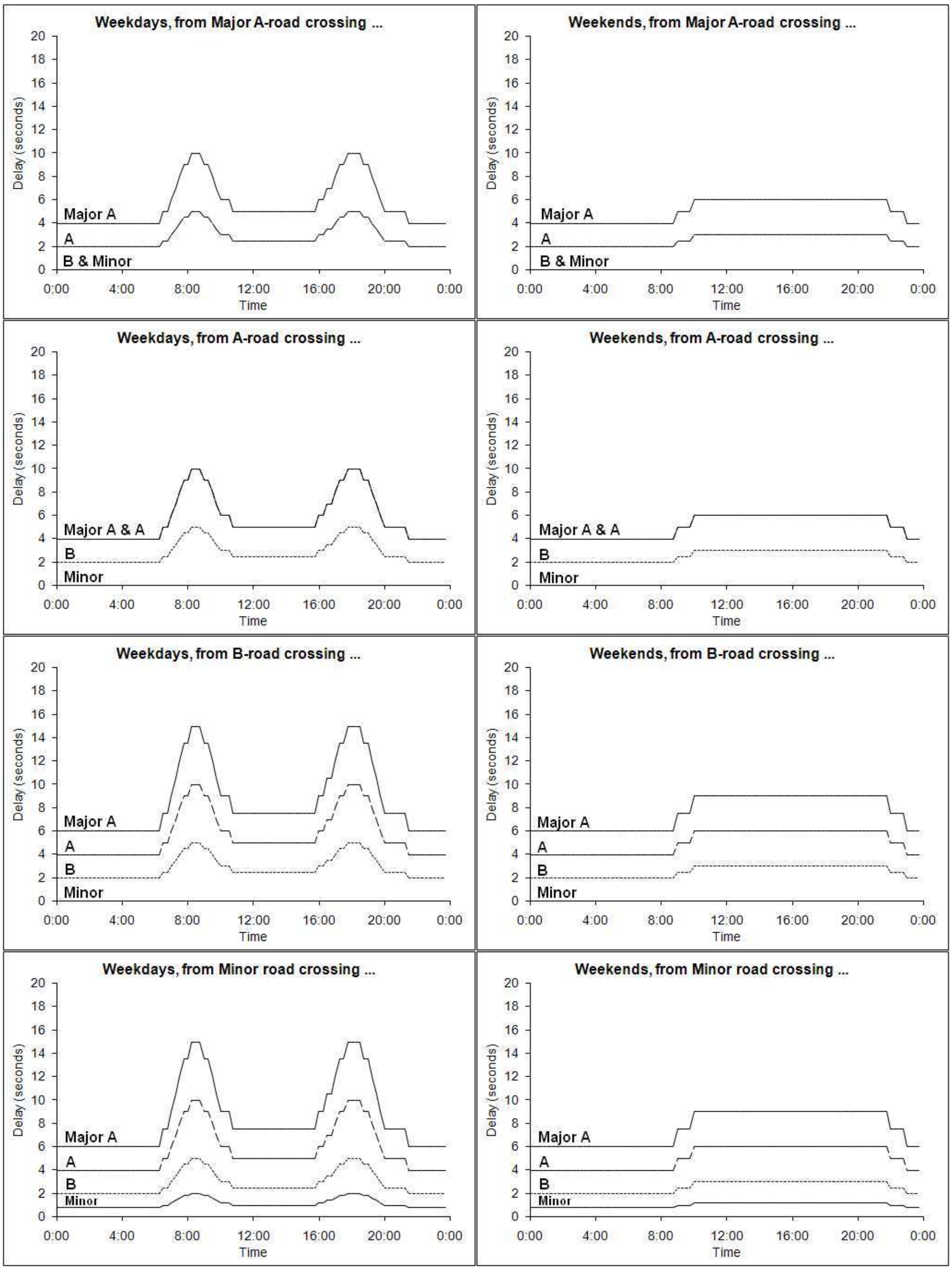


Figure 5.3-5: Simulated straight-on movements delay profiles

Having determined maximum delay values, these are multiplied by coefficients so as to obtain daily profiles. In the same way as is done for link speed values, days are split into 15-minute intervals and categorised into weekdays and weekends. The coefficients applied for each time interval depend on the time of the day, ranging from 0.4 during off-peak hours to 1 during the morning and afternoon peaks of weekdays. Weekend coefficients, on the other hand, range from 0.4 during off-peak hours to 0.6 at weekend peaks. Junction delay profiles for left turns and straight-on movements for each road type are shown in Figures 5.3-4 and 5.3-5.

After the presentation of the simulation procedure of junction delay values, the following sub-section deals with the simulation of earliness and lateness reliability profiles for links and junction movements.

5.3.4 Reliability data

To simulate reliability data for links and junction movements, a random function is adopted. Basically, a range of earliness values between 0.55 and 0.7 is selected after a fine-tuning procedure, corresponding to travel time variation logarithm values of 0.1085 and 0.0414 (see Section 3.3). Each link and junction movement is assigned a random travel time variation logarithm value from that range and the respective earliness and lateness index is derived as described in Section 3.3.

The values obtained are the maximum indices corresponding to the weekday peak hours. Employing the same process as in the link speed and junction delay data simulation, coefficients are applied to these values according to the time of the day, so as to deduce earliness and lateness daily profiles for all links and movements of the network. Nonetheless, the coefficients are not applied on the reliability index values themselves, but on the corresponding travel time variation logarithms. The coefficients range from 0.4 during off-peak times to 1 during weekday peaks; for weekends, the values range from 0.4 during off-peak to 0.6 during weekend peaks. An additional coefficient of 0.8 is applied on the variation logarithm values of minor roads, assuming that due to lower traffic flows, the amount of possible unpredictable delays is also lower, i.e. the reliability indices are higher.

The above randomly simulated reliability index values, however, result in all links and movements of the network being fairly reliable, as the range of values adopted ensures an adequate level of reliability. The earliness and lateness profiles of a reliable link are shown in Figure 5.3-6. To reflect the unreliability of “black spots” of the network in the corresponding network element reliability profiles, lower reliability values are assigned to specific links and movements for specific times of the day based on local expert knowledge, i.e. knowledge of individuals travelling on the network frequently. These may be professional drivers, bus drivers, taxi drivers or even simple commuters who drive along a route very often and happen to know that unpredictable delays frequently occur in a specific location at a specific time. To characterise a network element as ‘unreliable’ reliability values lower than the minimum earliness and lateness thresholds r_{Emin} and r_{Lmin} are applied (see Section 5.4.1).

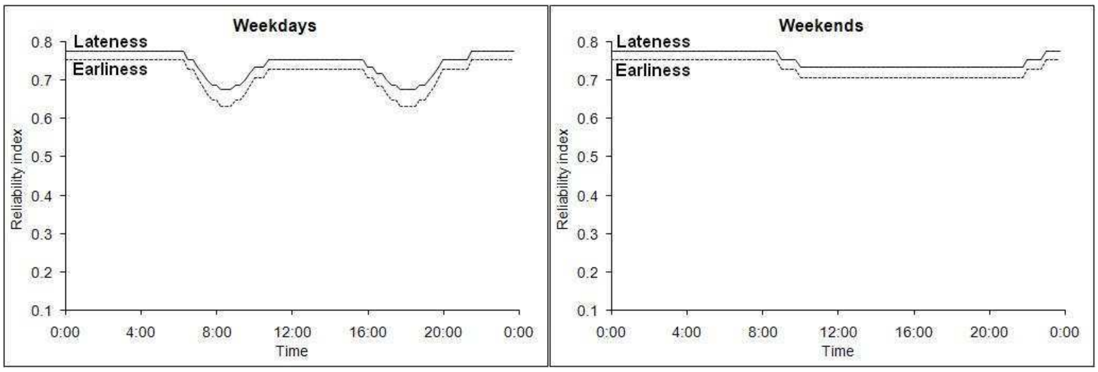


Figure 5.3-6: Simulated earliness and lateness profiles of a reliable link

In the Munich test network introduced in Section 5.3.1, a road stretch of the B13 (Ingolstädter Strasse) is set as unreliable during weekday afternoon peaks, based on expert knowledge of the network originating from BMW employees driving on the network during their daily commute. Namely, the earliness and lateness indices of 15 links are set to 0.2 and 0.37 respectively for a duration of eight 15-minute intervals between 17:00 and 19:00. The earliness and lateness profiles of these links are shown in Figure 5.3-7, while the links themselves are shown in the circled area of the map of Figure 5.3-8.

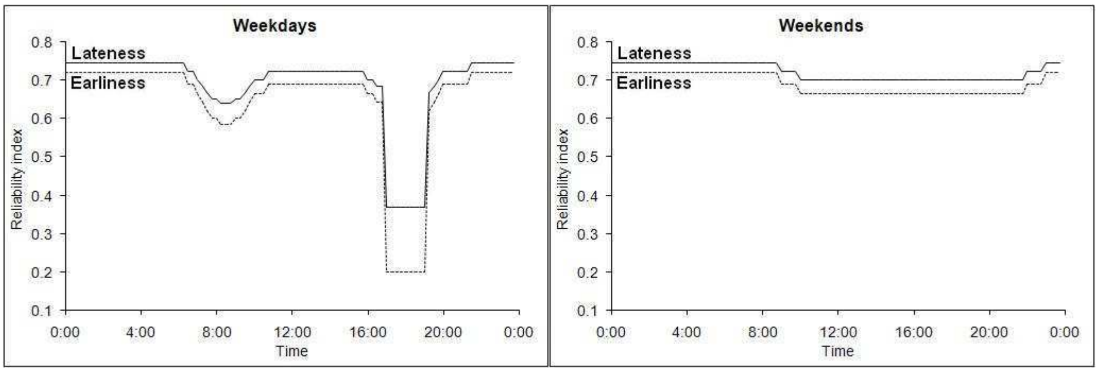


Figure 5.3-7: Earliness and lateness profiles of an unreliable link during weekday evening peaks

Following the simulation of traffic data, the next section reports on the simulation experiment and on the results obtained.

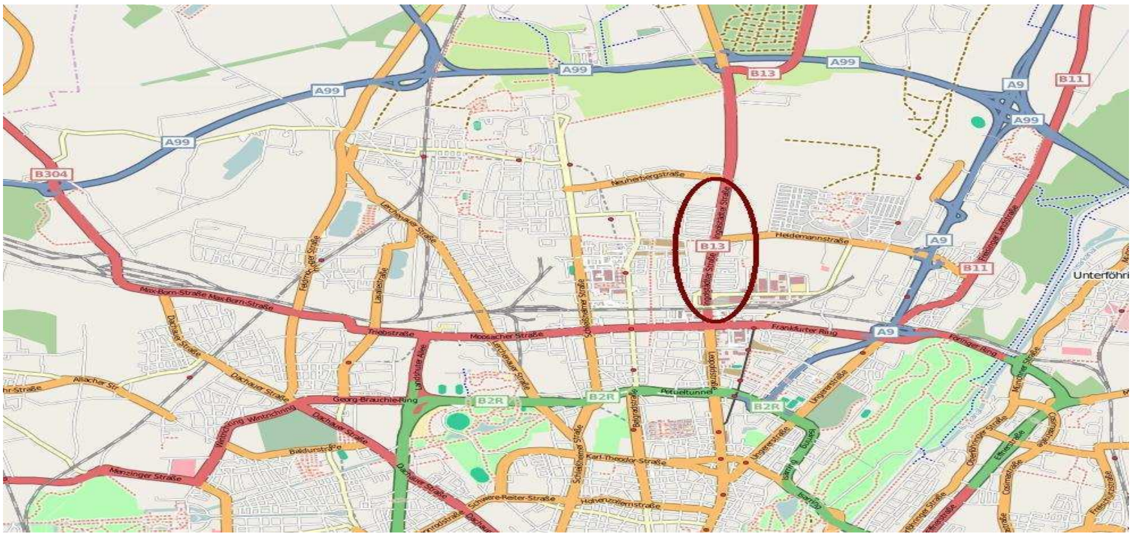


Figure 5.3-8: The unreliable part of the B13 during weekday afternoon peaks
(Source: www.openstreetmap.org)

5.4 Experimental conduct and results

This section describes the conduct of a simulation experiment using ARIAdNE, in order to demonstrate the RDIN and RDIN-R algorithms, presented in Chapter 4. The test network, link speed data and junction delay data described in the previous section are employed. The parameters employed and the procedure followed are reported here, followed by an account of the results obtained and their evaluation.

5.4.1 Parameters and procedure

Selecting the BMW headquarters (Dostlerstrasse) as the origin of the trip in ARIAdNE, and the northbound direction of the A92 motorway connecting the city with the airport as the destination, a set of equivalently reliable partially disjoint routes are sought. The departure time is 18:15 on a weekday, thus representing the afternoon peak and coinciding with the daily commute, resulting in high traffic flows out of the city centre.

The parameters input into ARIAdNE for the simulation experiment are presented here, following a fine-tuning procedure. These include the constraints employed by the RDIN and RDIN-R algorithms, as well as the threshold values imposed on the reliability indices, so as to characterise a link as ‘reliable’ or ‘unreliable’ and to penalise it accordingly.

Starting from the route travel time and length constraints, the values of the route travel time and length permission parameters β and ζ are set to 1.4 and 2 respectively for the RDIN algorithm. This implies that any route computed by ARIAdNE is only considered as acceptable if its travel time is up to 40% longer than the travel time of the fastest route and up to twice as long as the fastest route. The corresponding values for the RDIN-R algorithm are β_R = 1.7 and ζ_R = 2. Also, the route overlapping index constraint ε is set to 2. Finally, the maximum number of routes N_{max} computed by ARIAdNE is set to 3, that is up to three alternative reliable routes satisfying the imposed constraints can be supplied to the driver.

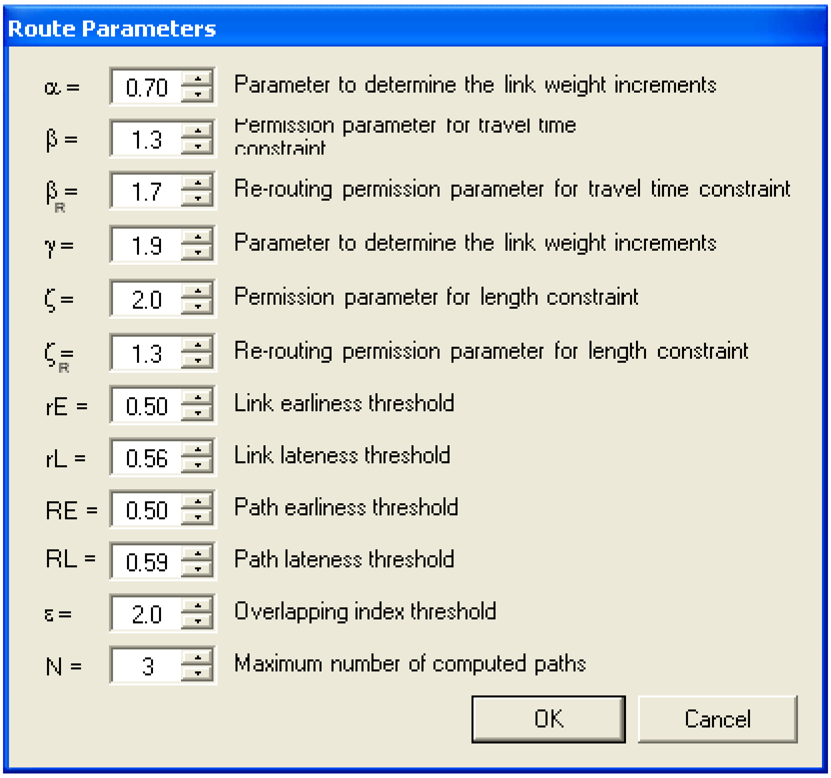


Figure 5.4-1: Experiment parameters, as shown in ARIAdNE

Regarding reliability values, the earliness and lateness thresholds specifying whether a link is reliable or unreliable are set to r_{Emin} = 0.5 and r_{Lmin} = 0.56 respectively, as a result of fine-tuning. Links that have earliness or lateness values lower than these thresholds are characterised as unreliable and are penalised in the RDIN and RDIN-R algorithms. For routes, the respective thresholds R_{Emin} and R_{Lmin} are set to 0.5 and 0.59, such that routes having lower reliability index values are considered as unacceptable by the driver and are therefore rejected by the RDIN algorithm. The confidence level for the calculation of the reliability indices is set to 90%, which results in a value of z = 1.65 to be used (see Chapter 3).

Last but not least, parameter α, which comes into the link penalty application function of the RDIN procedure and determines how large the penalty increments are, is set to 0.7. Equivalently, parameter γ, used in the calculation of the W_0 value for travel time penalties and also determining how large the penalty increments are, is set to 1.9.

The experiment is carried out as follows; the RDIN algorithm is run at first for the given origin-destination pair at 18:15 and a set of three acceptable routes is output. A route among those is selected as the one to be followed. Then, at 18:19 an incident is simulated on the selected route, thus causing the RDIN-R algorithm to be called and re-route from the vehicle’s current position to the destination, in order to avoid the affected area. The results of the experiment are presented next.

5.4.2 Results

The fastest route p_0 is computed first by the RDIN algorithm, not making any reliability or driver acceptability considerations; it is shown in Figure 5.4-2a and has an expected travel time of $T(p_0)$ = 19 minutes, a length of $\Lambda(p_0)$ = 14.7 km, and earliness and lateness indices of $R_E(p_0)$ = 0.44 and $R_L(p_0)$ = 0.54 respectively, implying a time window of -10.6/+16.5 minutes. Its reliability indices are lower than the acceptability threshold values R_{Emin} and R_{Lmin} and therefore the route is rejected and not added to the route set. This is normal as the route includes all the links of the B13 (Ingolstädter Strasse), that are set to be unreliable based on expert knowledge of the network. Nonetheless, despite not being acceptable, the fastest route computation enables the setting of the acceptability constraints for subsequent route computations.

In the next step of the RDIN algorithm reliability is taken into account and the unreliable links of the B13 are penalised. The resulting route, p_1, shown in Figure 5.4-2b is only slightly longer in terms of travel time than p_0, having an expected travel time of $T(p_1)$ = 19.4 minutes. Though significantly longer in terms of distance ($\Lambda(p_1)$ = 21.6 km), it is much more reliable, with earliness and lateness indices of $R_E(p_1)$ = 0.61 and $R_L(p_1)$ = 0.66 respectively, resulting in a time window of -7.5/+10 minutes. The route completely avoids all the unreliable links of the B13, following a longer but more reliable path along the A9 motorway and then along the A99 motorway. As the route meets all the acceptability criteria imposed including reliability it is accepted and added to the route set to be suggested to the driver.

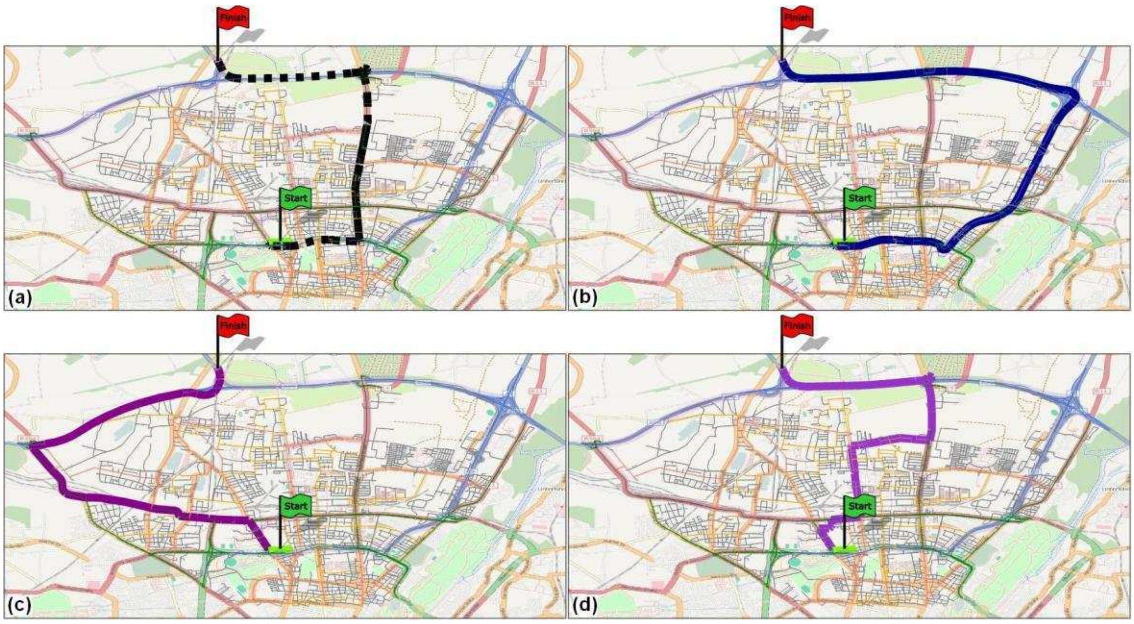


Figure 5.4-2: The fastest route (a) and the three reliable alternative routes (b-d)

In the next step of the RDIN algorithm an alternative route to p_1 is sought, this time not only avoiding unreliable links and junction movements, but also links and movements already included in p_1. Route p_2, shown on Figure 5.4-2c, being slightly longer than p_1 in terms of time ($T(p_2)$ = 21.5 minutes) and shorter in terms of distance ($\Lambda(p_2)$ = 17 km) is obtained. Its reliability indices are $R_E(p_2)$ = 0.63 and $R_L(p_2)$ = 0.68, representing a time window of -7.9/+10.3 minutes, thus making it more or less equivalently reliable to p_1. Nonetheless, the most important characteristic of p_2 is that it completely avoids not only the unreliable links of the B13 but also the links of p_1, guiding the driver towards the West instead of the East, along the B304 (Dachauer Strasse), and then along the A99, approaching the destination (A92) from the opposite direction to p_1.

For the third alternative route computed by the RDIN algorithm, the links and movements that need to be avoided as much as possible are more numerous than in the calculation of the other two routes, as they include the unreliable links of B13 and the links and movements of p_1 and p_2, i.e. the A9, the B304 and the A99. As completely avoiding the latter is not possible, since it is the only approach to the A92, the algorithm aims at using the A99 as little as possible. Route p_3, shown on Figure 5.4-2d, is obtained, with an expected travel time of $T(p_3)$ = 23.7 minutes, a length of $\Lambda(p_3)$ = 15.5 km and reliability indices of $R_E(p_3)$ = 0.64 and $R_L(p_3)$ = 0.68, representing a time window of -8.5/+11 minutes. As can be seen, the route avoids the unreliable parts of the B13, as well as most of the parts of p_1 and p_2, with the exception of the stretch of the A99 on p_1. The result is that the route is longer than the other two in terms of travel time, nonetheless it is acceptable (less than 40% longer than p_0) and is equivalently reliable, thus making it the third route of the set.

Figure 5.4-3: Re-routing on the first alternative route

It is up to the driver to choose a route to follow among paths p_1, p_2 and p_3, according to his/her individual preferences. In this simulation experiment route p_1 is selected and is thus denoted p_S. Four minutes after departure, at 18:19, an incident is reported further downstream of the vehicle's position, and more specifically, between the first and second junction of the A9 motorway. The RDIN-R algorithm is called to re-route and avoid the incident area by finding a new route to the destination.

The outcome of the RDIN-R algorithm is shown on Figure 5.4-3, where the incident link (not on route p_S any more) and the current position link are indicated and the part-route $p_R^{c,d}$ from the current position c to the destination d is drawn. Its travel time is $T(p_R^{c,d})$ = 20.2 minutes, i.e. slightly longer than the original part-route $p_S^{c,d}$ from c to d ($T(p_S^{c,d})$ = 15.4 minutes), while its length is $\Lambda(p_R^{c,d})$ = 19.6 km, and its reliability indices are $R_E(p_R^{c,d})$ = 0.61 and $R_L(p_R^{c,d})$ = 0.66, corresponding to a time window of -7.8/+10.3 minutes.

5.4.3 Evaluation

To assess the quality of the results obtained from the experiment, a small survey is carried out with local network experts. The set of experts employed consists of five employees of BMW Group and residents of the city of Munich, who drive on the test network described in Section 5.3.1 for their daily commute. The experts are initially presented with the origin-destination pair and traffic situation introduced in Section 5.4.1 and are asked to recommend the route that they would take if they were driving. The result is that among the five experts, four propose route p_1, while the last expert recommends route p_3. It is interesting to note that none of the experts suggests route p_0, which is in theory the fastest route.

The experts are then shown the route set output by ARIAdNE and are asked to comment on it and evaluate the three routes. The outcome is that they all agree that all three alternative routes are plausible, that they have used all of them for their commute at times and that they would all be considered by them if travelling at the time the simulation experiment is referring to. On the other hand, the experts agree that they would not consider travelling on the fastest route, since they know that there is a high probability of encountering congestion.

Finally, the experts are asked to comment of the travel times that they would expect to encounter along the routes, as well as on the time windows output by ARIAdNE. The result is that they indicate that both the expected and the earliest and latest travel times given are what they would expect to encounter themselves in the given situation. The evaluation thus suggests that the routes and travel times output by ARIAdNE are fairly reasonable.

5.5 Concluding remarks

This chapter introduced ARIAdNE, a software tool implementing the RDIN and RDIN-R algorithms, and its functions, structure and user interface were described. To demonstrate the use of the program on real road networks and to draw some preliminary conclusions on the applicability of the algorithms, a simulation experiment was carried out on a road network in Munich, Germany. The procedures for the simulation of traffic data, including link speed, junction delay and reliability index daily profiles, were described. Then, the experiment was carried out and the results were presented.

The results from the experiment seem to be fairly plausible, as both the routes computed and the travel time and reliability values calculated are reasonable and correspond to the choices made by experts of the network when questioned. This suggests that the RDIN and RDIN-R algorithms seem to yield promising solutions, though more experiments are required, so as to draw more concrete conclusions. For this purpose, the approach is field tested in a vehicle in Central London using floating vehicle data, employing ARIAdNE as the platform. The experiments conducted and the results obtained are reported in the next chapter.

CHAPTER 6

Validation and field experiments

6.1 Introduction

The validation of a theoretical approach is a very important task, as it is the only way of demonstrating the characteristics of a new method and of identifying its advantages and disadvantages, so as to conclude on whether it can be applied in reality or not. The in-vehicle navigation algorithm developed earlier in this study was initially tested using a simulation experiment in the previous chapter, and a preliminary finding was that it is fairly precise, as it corresponded to the route choices of local network experts. Nonetheless, a better validation method is field experimentation, i.e. observation of its performance in real traffic conditions. Namely, by implementing the algorithm in an actual vehicle and by using it as a normal navigation system, i.e. by following the route directions output, taking note of the travel times and traffic conditions encountered, useful conclusions can be drawn regarding its efficiency and accuracy.

Field experimentation is thus used in this study to validate the RDIN algorithm, presented in Chapter 4. For the conduct of the experiments, ARIAdNE, introduced in Chapter 5, is used as an implementation platform of the algorithm; this is installed in a conventional laptop and placed in a vehicle, which is then driven according to the route directions suggested by it. This chapter presents the methods used and the results obtained from the experiments, and draws conclusions on the correctness of the RDIN approach and its advantages compared to existing navigation systems.

It should be noted here, that although an algorithm for reliable re-routing is developed in Chapter 4, RDIN-R, this is not field tested in this study. The reason is that, in order to use RDIN-R in the field, accurate positioning and real-time information on traffic incidents are needed. Regarding the former, an interface between a Global Positioning System (GPS) receiver and ARIAdNE is required, including the use of a map-matching algorithm. Concerning the latter, the only way of obtaining real-time traffic information is to interface ARIAdNE with the Traffic Message Channel. Both are fairly demanding tasks and are therefore beyond the scope of the research presented in this study. Hence, the field experiments reported only refer to the RDIN algorithm and the validation of the RDIN-R algorithm is only based on the result of the laboratory simulation experiment described in Chapter 5.

The chapter is structured as follows: Section 6.2 presents the methods employed for the acquisition and processing of the data required to conduct the experiments. Section 6.3 describes the experimental methods, as well as the parameters used, Section 6.4 reports and discusses the results obtained from the experiments, while Section 6.5 expresses some concluding remarks.

6.2 Data acquisition and processing

Prior to the description of the field experiments, the data sources used in this study are presented. Data is obtained with regard to the test network and the travel times on the links of the test network, so as to input the data needed in ARIAdNE for the execution of the RDIN algorithm. Travel time and travel time reliability data is derived from measurements of the Floating Vehicle Data (FVD) system developed by ITIS Holdings PLC, while the test network is obtained from Planung Transport Verkehr AG (PTV AG) in the form of a PTV VISUM™ map database, extracted from a Navigation Technologies (NAVTEQ™) digital map database.

The background of travel time estimation in general and FVD is given first, followed by the presentation of the test network. Then the collection and analysis of the data so that it can be used in ARIAdNE is described.

6.2.1 Background of travel time estimation and FVD

The estimation and prediction of travel time in an urban network is a topic that is of vital importance to many transport applications and has therefore been extensively researched in the past and continues to be analysed by many researchers, such as Park and Rilett (1998), Park et al (1999) and Awasthi et al (2003). This is also the case in route guidance, since a navigation system that has access to accurate travel time predictions can also provide accurate route suggestions. Important contributions to the field are the works of Hoffmann and Janko (1990) in the context of the ALI-SCOUT project, Boyce et al (1993) in the context of the ADVANCE project, Sen et al (1995; 1998; 1999) and Kerner et al (2005); these introduce methods, according to which default travel time profiles for a route guidance system are estimated, using data obtained from probe vehicles (vehicles equipped with a navigation system).

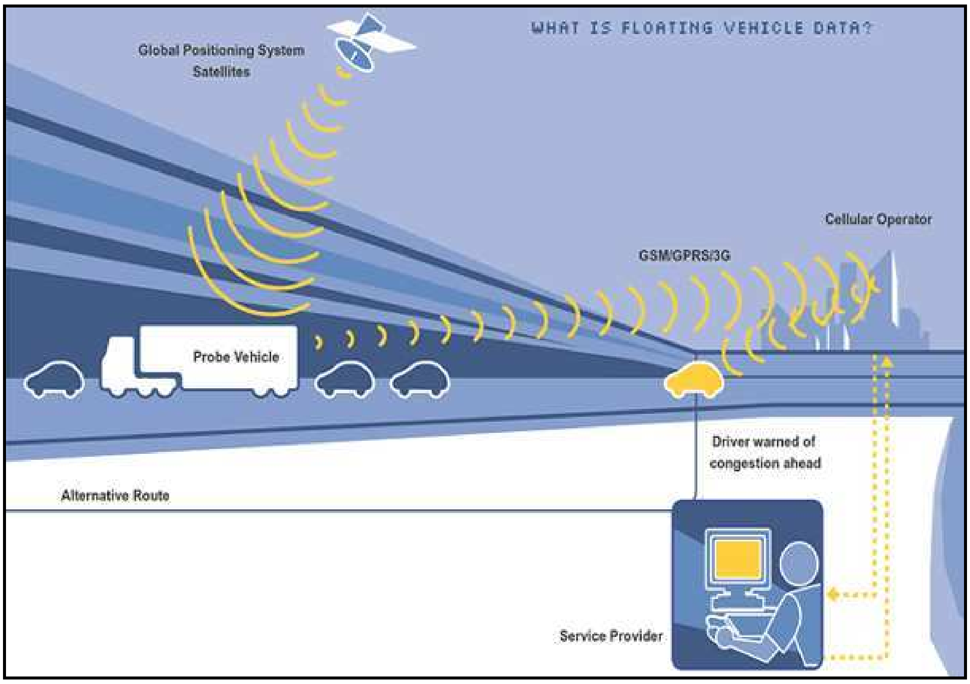


Figure 6.2-1: Operation of the ITIS FVD^{TM} system
(ITIS Holdings PLC Website, 2007)

In recent years a new method of estimating travel time has been developed; this is called Floating Vehicle Data (FVD) and its concept is based on the transmission of traffic data from a number of vehicles equipped with measuring devices actually "floating" in traffic to the service provider, the processing of this data to convert it to traffic information and the re-broadcast of this data to the vehicles, equipped with receivers. In the UK, the first fully operational FVD system has been developed by ITIS Holdings PLC. The ITIS FVD™ system has been collecting data since February 2000, while commercial provision of the data gathered has been in place since 2002. With more than 2.5 million measurements per day, the system is now considered the largest of its kind in the world (Cowan and Gates, 2003). A schematic of the operation of the ITIS FVD™ system is shown in Figure 6.2-1.

The main advantage of the FVD system, as opposed to other existing travel time data collection schemes, is the fact that data can be obtained wherever there is coverage by the GPS, whenever the ignition of the vehicle is on; this greatly minimises the cost arising from installing conventional measuring devices (such as inductive loop detectors) on many roads, while at the same time it enables the monitoring of roads that are not equipped with such devices, such that the UK strategic highways network (all but the minor roads) can be monitored (Simmons et al, 2002; Ilgaz et al, 2002). The collection of data occurs as follows: each vehicle from a fleet of about 50,000, consisting of private vehicles, business vehicles, trucks and coaches, is equipped with a Data Collection Unit (DCU), transmitting its location (longitude and latitude) at pre-determined intervals. Then speed is computed between two consecutive journey points and is assigned using a map-matching procedure to a specific link on a digital road network (see Section 6.2.2), thus giving an indication of the traffic situation on that link at the time of the measurement.

While the initial purpose of the ITIS FVD™ system was to obtain current traffic data and to broadcast it to vehicles, an additional function (or "by-product") that has recently gained value is the creation of a historical travel time database from past measurements. One of the reasons is that an important advantage of a FVD system is its ability to collect data 24 hours a day, 365 days a year. The resulting database of speed measurements enables the calculation of key statistical values for the travel time on each link of the network, such as the mean, the standard deviation and confidence intervals; those can be used in many transport planning applications, particularly in congestion monitoring, accessibility modelling (derivation of isochrones, based on real data rather than speed limits or default values, as is the current practice) and representation of tidality of traffic in town and city centres (Traffic Engineering and Control Product Review, 2003; Storey and Holtom, 2003).

In route guidance, ITIS FVD™ has only been used for the provision of current traffic data so far. In this study, ITIS FVD™ speed measurements are being utilised for the calculation of mean speed and reliability values (according to the methodology described in Chapter 3) for the links of a test network in London, so as to be used by the RDIN algorithm in ARIAdNE, described in Chapters 4 and 5. Before reporting on the extraction and processing of the data, which is described in Section 6.2.3, the test network is presented.

6.2.2 Test network

For the conduct of the field experiments, a test network meeting a number of requirements, so as to be suitable for that purpose, is to be defined. First of all, the network needs to be large enough to offer a route choice, such that alternative equivalently reliable partially disjoint routes can be computed for each origin-destination pair. Routes of different lengths and different characteristics should be available, so that the RDIN algorithm developed can be tested thoroughly.

Then, there needs to be adequate coverage of the ITIS FVD™ system on the network, such that enough data is available for the conduct of the experiments. As was mentioned in Section 6.2.1, the ITIS FVD™ system only covers the UK strategic highways network; since for route guidance a detailed network including all minor roads is needed, some travel time and reliability data will inevitably have to be simulated for the field experiments. However, as it would be better for the experiments to use as much real data as possible and to minimise the amount of simulated data, a network with a good ITIS FVD™ coverage (i.e. with a high density of main roads) would be more appropriate. Such are usually the city centres of major conurbations.

Another requirement is that the network needs to be representative of a random network used in route guidance. This requires the network to be as diverse as possible, that is, to contain areas with different land uses and road types. A network of an entirely residential or commercial area is not suitable because it is not representative of the situation encountered by a navigation system. Similarly, a network consisting solely of one or two road types, as these are defined by NAVTEQ™ (Motorway, Major A-Road, A-Road, B-Road, Minor Road) is inappropriate for the same reason.

The network that is selected for this study is the original London Congestion Charging Zone (LCCZ), consisting of 6240 nodes and 12891 links and spreading over an area around 7 km long and 5 km wide. As it covers most areas of Central London, it exhibits a great diversity of land uses and contains all road types but one (it does not contain a motorway), thus being representative of a random network used in route guidance. Also, it has good ITIS FVD^{TM} coverage (see Figure 6.2-2), because it has a high density of main roads.

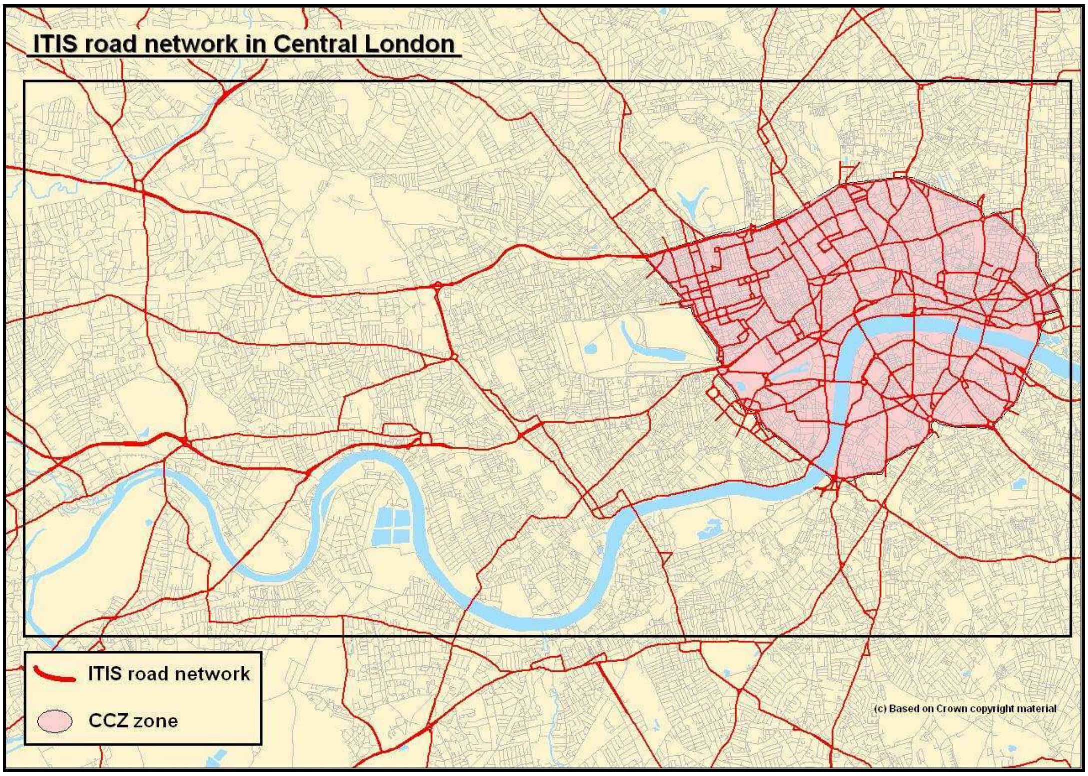


Figure 6.2-2: ITIS FVD^{TM} coverage in Central London

(dark area indicates the test network)

(Personal communication with Ian Forshaw, ITIS Holdings PLC, 2007)

It should be noted, however, that the test network excludes the western extension of the LCCZ, because the ITIS FVD^{TM} measurements acquired (see Section 6.2.3) refer to a period prior to the start of its operation (19 February 2007), while the experiments are carried out after that (July 2007). Since the extension of the LCCZ has resulted in a reduction of traffic in the affected areas and hence in an increase of the speeds, the available data is incorrect and therefore the extension of the zone is excluded from the test area.

The test network is shown on Figure 6.2-3 and is supplied in PTV VISUM^{TM} format. For each network element, a number of attributes are included. For nodes these are the ID number corresponding to the NAVTEQ™ database and the location with X and Y co-ordinates. For links the following are given: ID number in NAVTEQ™ database, ID of start node, ID of end node, co-ordinates of intermediate points indicating the curvature, road type, speed limit and direction. Finally, for movements the ID numbers of the three nodes specifying the turn (start node, via node, end node) and the type of the turn (left, right, straight-on or U-turn) are given.

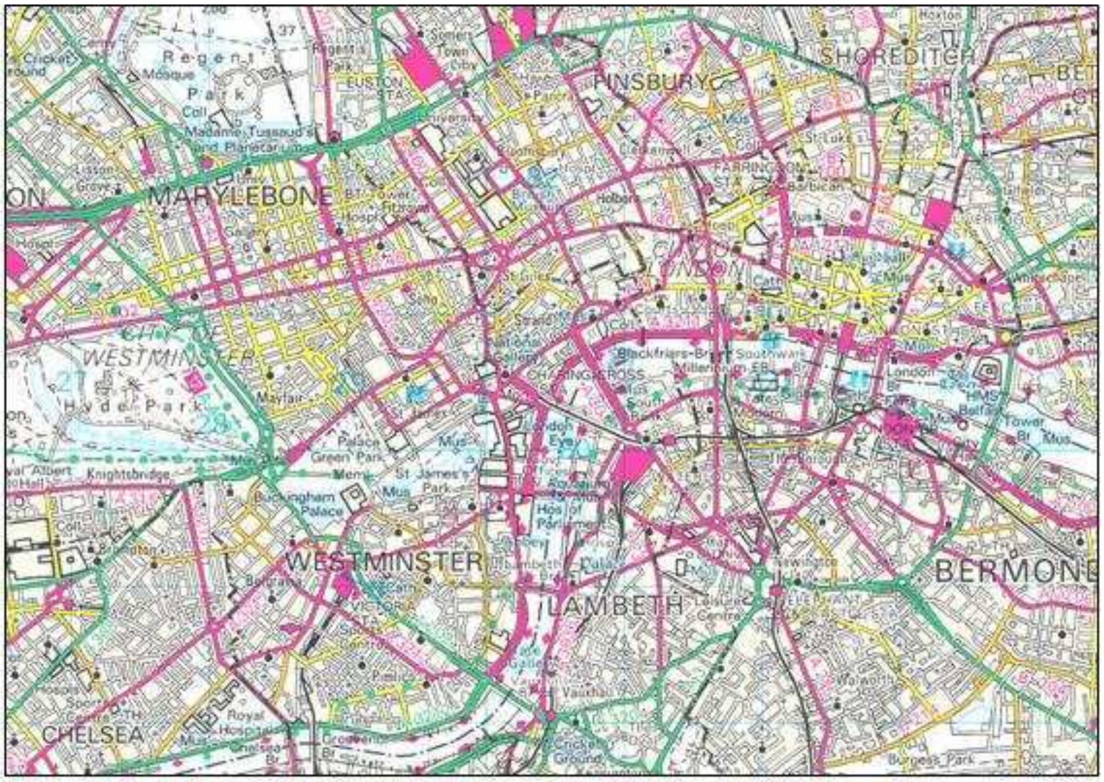


Figure 6.2-3: The test network

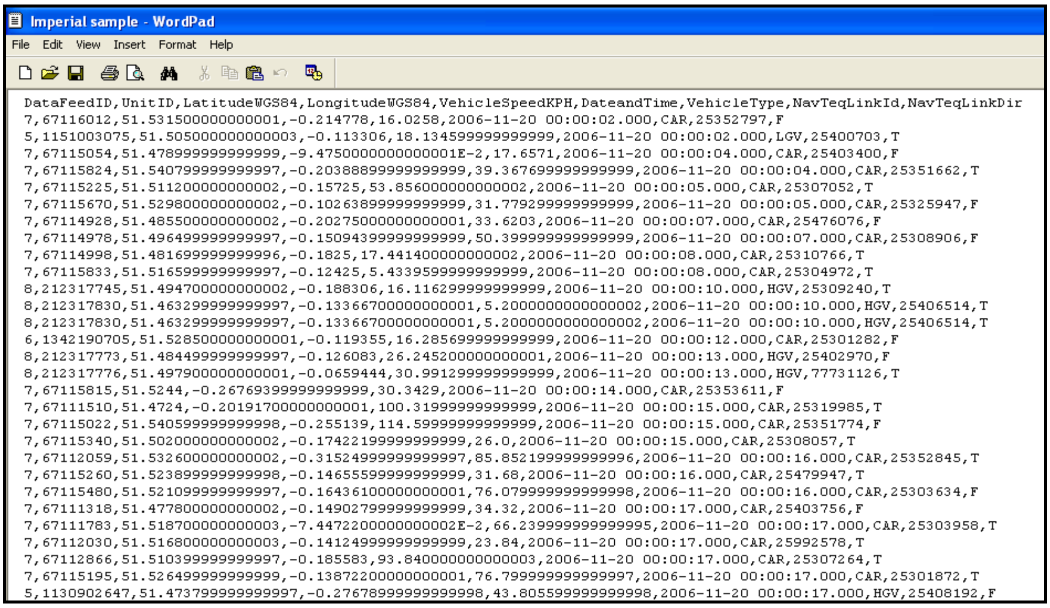
The network is subsequently imported into ARIAdNE, so that it can be used for the field experiments. The next two sections report on the acquisition and processing of the ITIS FVD™ measurements.

6.2.3 ITIS FVD™ set collection and processing

ITIS FVD™ measurements are supplied for the conduct of the field experiments in this study. For the test network introduced in the previous section, data collected by the ITIS FVD™ system during a three month period, between 1 September 2006 and 1 December 2006, is obtained. This corresponds to about 7.6 million speed measurements, provided in delimited text format, where each row represents a single measurement. The attributes of each measurement are the following:

- Unit ID, representing the unique vehicle ID supplying the measurement
- Latitude of the recorded position of the vehicle during the measurement
- Longitude of the recorded position of the vehicle during the measurement
- Speed of the vehicle in km/h at the time of the measurement
- Date and time of the measurement
- Vehicle type (car, bus, heavy goods vehicle (HGV) or light goods vehicle (LGV))
- Map-matched NAVTEQ™ link ID number
- Direction of the NAVTEQ™ link

A sample extract of the data is shown in Figure 6.2-4, where the individual attributes for each measurement can be seen.


Imperial sample - WordPad
File Edit View Insert Format Help

```
DataFeedID,UnitID,LatitudeWGS84,LongitudeWGS84,VehicleSpeedKPH,DateandTime,VehicleType,NavTeqLinkId,NavTeqLinkDir
7,67116012,51.531500000000001,-0.214778,16.0258,2006-11-20 00:00:02.000,CAR,25352797,F
5,1151003075,51.505000000000003,-0.113306,18.134599999999999,2006-11-20 00:00:02.000,LGV,25400703,T
7,67115054,51.478999999999999,-9.4750000000000001E-2,17.6571,2006-11-20 00:00:04.000,CAR,25403400,F
7,67115824,51.540799999999997,-0.20388899999999999,39.367699999999999,2006-11-20 00:00:04.000,CAR,25351662,T
7,67115225,51.511200000000002,-0.15725,53.856000000000002,2006-11-20 00:00:05.000,CAR,25307052,T
7,67115670,51.529800000000002,-0.10263899999999999,31.779299999999999,2006-11-20 00:00:05.000,CAR,25325947,F
7,67114928,51.485500000000002,-0.20275000000000001,33.6203,2006-11-20 00:00:07.000,CAR,25476076,F
7,67114978,51.496499999999997,-0.15094399999999999,50.399999999999999,2006-11-20 00:00:07.000,CAR,25308906,F
7,67114998,51.481699999999996,-0.1825,17.441400000000002,2006-11-20 00:00:08.000,CAR,25310766,T
7,67115833,51.516599999999997,-0.12425,5.4339599999999999,2006-11-20 00:00:08.000,CAR,25304972,T
8,212317745,51.494700000000002,-0.188306,16.116299999999999,2006-11-20 00:00:10.000,HGV,25309240,T
8,212317830,51.463299999999997,-0.13366700000000001,5.2000000000000002,2006-11-20 00:00:10.000,HGV,25406514,T
8,212317830,51.463299999999997,-0.13366700000000001,5.2000000000000002,2006-11-20 00:00:10.000,HGV,25406514,T
6,1342190705,51.528500000000001,-0.119355,16.285699999999999,2006-11-20 00:00:12.000,CAR,25301282,F
8,212317773,51.484499999999997,-0.126083,26.245200000000001,2006-11-20 00:00:13.000,HGV,25402970,F
8,212317776,51.497900000000001,-0.0659444,30.991299999999999,2006-11-20 00:00:13.000,HGV,77731126,T
7,67115815,51.5244,-0.26769399999999999,30.3429,2006-11-20 00:00:14.000,CAR,25353611,F
7,67111510,51.4724,-0.20191700000000001,100.31999999999999,2006-11-20 00:00:15.000,CAR,25319985,T
7,67115022,51.540599999999998,-0.255139,114.59999999999999,2006-11-20 00:00:15.000,CAR,25351774,F
7,67115340,51.502000000000002,-0.17422199999999999,26.0,2006-11-20 00:00:15.000,CAR,25308057,T
7,67112059,51.532600000000002,-0.31524999999999997,85.852199999999996,2006-11-20 00:00:16.000,CAR,25352845,T
7,67115260,51.523899999999998,-0.14655599999999999,31.68,2006-11-20 00:00:16.000,CAR,25479947,T
7,67115480,51.521099999999997,-0.16436100000000001,76.079999999999998,2006-11-20 00:00:16.000,CAR,25303634,F
7,67111318,51.477800000000002,-0.14902799999999999,34.32,2006-11-20 00:00:17.000,CAR,25403756,F
7,67111783,51.518700000000003,-7.4472200000000002E-2,66.239999999999995,2006-11-20 00:00:17.000,CAR,25303958,T
7,67112030,51.516800000000003,-0.14124999999999999,23.84,2006-11-20 00:00:17.000,CAR,25992578,T
7,67112866,51.510399999999997,-0.185583,93.840000000000003,2006-11-20 00:00:17.000,CAR,25307264,T
7,67115195,51.526499999999999,-0.13872200000000001,76.799999999999997,2006-11-20 00:00:17.000,CAR,25301872,T
5,1130902647,51.473799999999997,-0.27678999999999998,43.805599999999998,2006-11-20 00:00:17.000,HGV,25408192,F
```

Figure 6.2-4: Sample extract from the supplied ITIS FVD™ file

In order to use the ITIS FVD™ values, a data aggregation procedure needs to take place. Namely, the data needs to be aggregated temporally by day and time of day, such that it can be imported and used in ARIAdNE. As is the case in Chapter 5 with the simulation experiments, days are categorised into weekdays and weekends, while each day is split into 15-minute intervals. Based on the time and day of the observation of an individual measurement, the speed value is assigned to a link and to a specific day and time interval.

The individual measurements for a specific link and time interval form a distribution of speeds. Hence, by calculating the space-mean speed (harmonic mean of the distribution), average speed values are obtained and can be used in the place of the speeds typically used in in-vehicle navigation. Following that, the variance of the distribution is calculated and, by choosing a level of confidence, earliness and lateness reliability index values are computed for each link and time interval, based on the methodology described in Chapter 3. The result from the data aggregation is that for each link there are six time series: weekday speed, earliness and lateness, and weekend speed, earliness and lateness, each one of which has 96 values for each one of the 15-minute intervals. An example is shown in Figure 6.2-5. When plotted against time of day, the speed, earliness and lateness profiles are obtained.

Speed and Reliability of Link 403

Link: 403

Time of Day	Speed Weekday	Earliness Weekday	Lateness Weekday	Speed Weekend	Earliness Weekend	Lateness Weekend
08:30 - 08:45	13.27803	0.558979	0.6197443	20.9306	0.5534885	0.6156473
08:45 - 09:00	14.5736	0.4045069	0.5088315	25.10223	0.5683131	0.6267384
09:00 - 09:15	14.89784	0.5013449	0.5773436	27.40285	0.5939032	0.6461054
09:15 - 09:30	16.93915	0.5233178	0.5933536	25.28883	0.5332028	0.6006175
09:30 - 09:45	14.07513	0.5113037	0.5845768	18.31372	0.6138989	0.6614398
09:45 - 10:00	12.71176	0.4939333	0.5719846	20.68747	0.5394765	0.6052479
10:00 - 10:15	14.27702	0.4926167	0.5710348	22.64505	0.7451369	0.7668763
10:15 - 10:30	14.72771	0.4673081	0.5528988	16.98093	0.5701292	0.6281036
10:30 - 10:45	13.50363	0.5746697	0.6315228	21.73944	0.5311649	0.5991168
10:45 - 11:00	13.64653	0.5012096	0.5772457	20.84246	0.6333962	0.6765682
11:00 - 11:15	13.51178	0.4852415	0.565726	22.81372	0.5903146	0.6433722
11:15 - 11:30	13.81358	0.4924496	0.5709144	19.96125	0.4913941	0.5701534
11:30 - 11:45	13.75413	0.4996826	0.5761399	15.22243	0.5491977	0.6124542
11:45 - 12:00	14.43804	0.52873	0.597326	17.35081	0.5915919	0.6443444
12:00 - 12:15	13.99996	0.4965422	0.5738687	16.16674	0.4919347	0.5705431
12:15 - 12:30	12.59628	0.4613147	0.5486372	11.86899	0.4616694	0.548889
12:30 - 12:45	12.40587	0.4634326	0.5501417	13.61352	0.4872026	0.5671357

OK Cancel

Figure 6.2-5: The speed, earliness and lateness lists for a link

It should be noted here that a filter is applied to the data aggregation, such that a minimum number of observations is needed in order to be able to calculate representative statistical values. Namely, when very few observations exist for a specific link and time interval, it is considered that the resulting mean and variance are not statistically reliable and it is not safe to assume that they reflect the real speed distribution. Therefore, whenever less than three speed measurements are present, these are ignored and simulated data based on network aggregates (see Section 6.2.4) is used for the link and time interval in question.

Additionally it should be noted that a filter is also applied to the data aggregation when it comes to abnormally low values. When a value from an observation is too low (below 7 km/h), this indicates that it has taken longer than normal for the specific vehicle to traverse the link in question possibly due to the vehicle having stopped while the measurement was taken. The value of 7 km/h is determined based on the assumption, that on a 100m-long link, the worst-case scenario is that a vehicle travels at 50 km/h and then experiences a 45-second delay at the upcoming intersection, resulting in a total traversal time of roughly 50 seconds. Hence, observations below 7 km/h are discarded and are not included in mean and reliability calculations. However, as by excluding low-speed observations stationary traffic may be ignored, it is examined whether for any link and time interval there is a high proportion of low-speed measurements available. As such, if low speeds form more than 30% of the total number of measurements for a link and time interval combination, these are not ignored and are included in the mean and reliability calculations.

6.2.4 Simulation of missing data

For the links and time intervals, for which speed data is not provided by the ITIS FVD™ system (either because the system does not cover them or because not enough data points are available), mean speed and reliability data are simulated. The simulation procedure involves extrapolating values from the links for which data is available to the ones for which data is not available.

Regarding speed data and provided that using plain speed values would be incorrect due to the fact that on different road types different speeds are achievable, the chosen value to extrapolate is the ratio of the mean speed to the free-flow speed. Assuming that when no congestion is present the highest speed at which a vehicle would travel on a road is the speed limit, it can be assumed that the free-flow speed corresponds to the speed limit, which is provided for every link in the map database (see Section 6.2.2). In order to extrapolate the data, the ratio of the mean speed over the speed limit (speed ratio) is calculated for every link and time interval, for which data is available. For all those links and for each time interval individually, the average speed ratio is computed; the resulting value is assigned to all the remaining links of the network, for which no data is available. Speed values can then be derived for these links and time intervals by multiplying the speed ratio assigned to them with their speed limit value.

On the top two graphs of Figure 6.2-6, the average speed ratio profiles for all links of the network are shown. These are also an indication of the level of congestion of the network. As one would expect, there is a sharp drop in the speed of traffic on the links of the network in the morning hours for weekdays, as well as a smoother increase in the evening hours, leading to much higher speeds during the night.

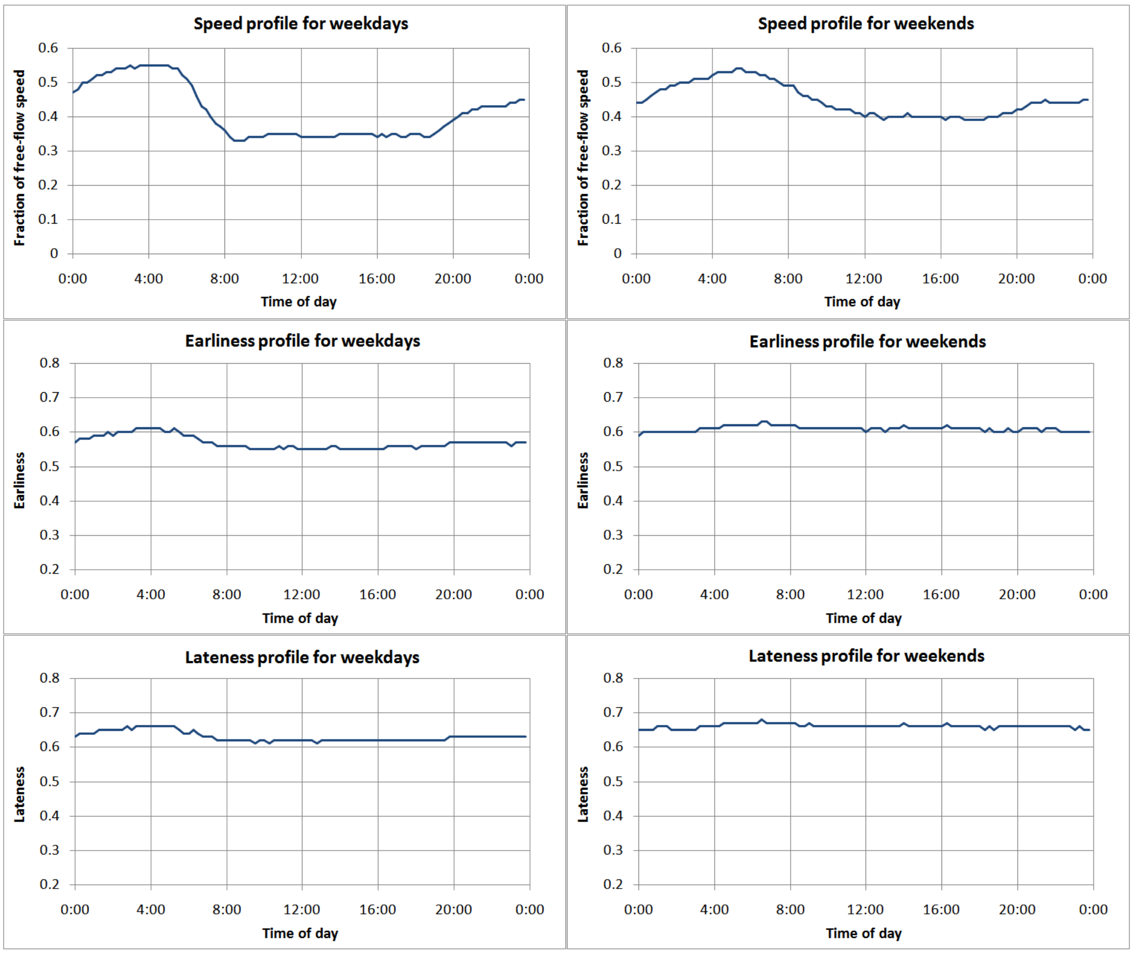


Figure 6.2-6: Simulated speed, earliness and lateness profiles

What would be expected but cannot be seen on the graph is that during the inter-peak period (between the morning and evening peaks) higher speeds than the ones during the peak periods should be observable; instead, it seems as though the speeds during that period remain more or less constant. The reason for this may lie in the fact that the network is from the city centre of a major conurbation, which means that greater traffic flows are present throughout the day rather than just during the peak periods. Also, it is conjectured that another reason may be the Congestion Charging scheme, which prevents commuters from driving to work, hence reducing the peak period flows. On the other hand, two observations can be made on the weekend graph: the smooth drop of speeds during the morning periods, leading to low speeds (high flows) in the afternoon, which are nevertheless higher than the weekday ones, and the lower speeds of the very early hours of weekends, resulting from the city's nightlife on Friday and Saturday night.

Concerning reliability data, the method of simulation is similar to the one used for mean speeds. Namely, the average earliness and lateness indices are computed over all links, for which data is available, and for each time interval individually. Then, the values computed are assigned to the links, for which data does not exist. In the bottom four graphs of Figure 6.2-6, the average earliness and lateness profiles for weekdays and weekends are shown. It can be seen that while reliability is more or less constant on weekends, there are slight variations on weekdays; also, both earliness and lateness are a bit higher on weekends. This can be attributed to the fact that unpredictable congestion is more likely to arise on weekdays, when higher congestion levels are present.

Finally, in the methodology presented in Chapter 4, provision was made for junction delays to be considered in the algorithm and to be included in the route finding procedure. Values were simulated in Chapter 5, so as to run the laboratory experiment of ARIAdNE, based on the turn type encountered and the road types meeting at the junction. For the field experiments of this chapter real junction delay values are not available. ITIS FVD™ values, however, are actual speed measurements and reflect the traversal time of links by specific vehicles, and therefore, they include the delay that a vehicle may have encountered at a junction.

Of course, different turning movements at the end of each link would introduce different delay values (e.g. right turns would induce longer delays than straight-on movements); the disaggregation of link travel time and subsequent junction delays, however, is a task that extends beyond the scope of this study. Therefore, for reasons of simplicity, junction delay and reliability are assumed to be the same for all turning movements encountered at the end of a specific link in these experiments and their values are thus considered as being included in the corresponding link travel time value.

6.3 Conduct of the field experiments

Having described the methodology behind the acquisition of the data needed to carry out the field experiments, the conduct of the experiments themselves is reported in this section. The parameters and setup employed are introduced first, leading to the results reported in the next section.

Two field experiments are carried out in this study. In the first one (single-vehicle experiment), one vehicle is used and is driven in the test network along routes suggested by ARIAdNE and observations are made; comparison of the experienced travel time with the travel times predicted by ARIAdNE and by a conventional system is also made. In the second one (double-vehicle experiment), two vehicles are used, such that one vehicle is driven along routes suggested by ARIAdNE, while the other follows routes suggested by a conventional system; comparison is made between the two systems as concerns the overall travel time experienced, so as to show that, in the absence of real-time traffic information, the guidance offered by ARIAdNE is superior to the one offered by the conventional system.

6.3.1 Conduct and analysis of the single-vehicle experiment

The apparatus of the single-vehicle experiment consists of a private car, a portable conventional car navigation system and a conventional laptop, on which ARIAdNE is implemented. The configuration is shown on Figure 6.3-1, that is the laptop is placed onto the passenger seat next to the driver, while the conventional system is placed on the windscreen. The driver manually sets the position of the vehicle in ARIAdNE (this is established through the GPS in the conventional system) as the origin link and determines a destination, which he/she inputs in both ARIAdNE and the conventional system. The conventional system only suggests one route, which is the fastest according to the information it has access to, i.e. only the speed limits. ARIAdNE on the other hand computes a maximum of three alternative reliable routes, based on the speed and reliability values derived from the ITIS FVD™ system. The driver selects the route he/she prefers from the set output by ARIAdNE, takes note of the departure time (DT), of the expected times of arrival of both the conventional system (CETA) and ARIAdNE (AETA), as well as of the earliest and latest reliable times of arrival of ARIAdNE (AERTA and ALRTA).

Having selected a route, the driver starts driving along it, by following the driving directions displayed in the respective box of ARIAdNE. When reaching the destination, the driver takes note of the actual time of arrival (ATA), compares it to the CETA and the AETA, and also checks whether it lies between the AERTA and the ALRTA. All the recorded values from each run are added to a table, from which the data is subsequently processed. Namely, from the recorded arrival time values, the following values can be derived: the conventional expected travel time (CETT = CETA – DT), the ARIAdNE expected travel time (AETT = AETA – DT), the ARIAdNE earliest and latest reliable travel times (AERTT = AERTA – DT and ALRTT = ALRTA – DT) and the actual travel time (ATT = ATA – DT).

Figure 6.3-1: Configuration of the single-vehicle field experiments

In order to make the derived travel times over all measurements comparable with each other, the fractions of those to the ATT are calculated; hence a new table containing the ratios of each travel time value (CETT, AERTT, AETT and ALRTT) to the ATT is deduced. For each of those ratios a distribution is obtained from the total of the runs of the experiment; by ranking the distribution and by dividing the rank of each value by the total number of runs, cumulative probabilities are obtained, which result in a cumulative distribution. Then, plotting the cumulative distributions of all four travel times in a single graph, a visual comparison between them becomes possible.

6.3.2 Conduct and analysis of the double-vehicle experiment

The apparatus of the double-vehicle experiment is very similar to the single-vehicle experiment, the main difference being the fact that two cars are used instead of one. Namely, the configuration of the first vehicle (called ARIAdNE vehicle) is the one shown in Figure 6.3-1, without the conventional system. The conventional system is in fact placed on the windscreen of the second vehicle (called conventional vehicle), "equipped" with a second driver. The drivers meet at the same point to start their trip and the driver of the ARIAdNE vehicle enters the current position of the vehicle as the origin. Then the two drivers agree on a destination, input it in their systems, which calculate route(s) for them, and take note of the DT. The conventional vehicle driver takes note of the CETA provided by his/her system, while the ARIAdNE vehicle driver selects a route among the set of reliable routes given and writes down the AETA, the AERTA and the ALRTA of this route.

The two drivers then begin to follow the route directions given by their systems. Upon arrival at the destination, the drivers observe which of the two arrived earlier and write down the conventional actual time of arrival (CATA) and the ARIAdNE actual time of arrival (AATA). All the recorded values are added to a table, from which they are subsequently processed. It should be noted here, that in order to ensure objectivity and eliminate the possible advantage of one system compared to the other arising from the driving style of the one or the other driver (i.e. if one driver is faster than the other thus always arriving earlier), the drivers are instructed to adopt a 'neutral' driving style (using the speed limit as an indication where possible), and they also swap systems for half the runs of the experiment. Namely, following a set of runs (usually two or three) the first driver takes the conventional system in his/her vehicle and gives ARIAdNE to the second driver and the experiments continue to be carried out as normal.

When it comes to the analysis of the results, similarly to the single-vehicle experiment, travel times are computed from the arrival times that have been recorded. A new table is thus constructed, having columns for each one of the CETT, the AETT, the AERTT, the ALRTT (as in Section 6.3.1), the conventional actual travel time (CATT = CATA – DT) and the ARIAdNE actual travel time (AATT = AATA – DT). Also, the difference in the actual travel time (DATT) between the two systems is computed, such that it can be seen which system "won" in each run (DATT = CATT – AATT). When DATT is positive, this means that the ARIAdNE vehicle arrived earlier than the conventional vehicle, while the opposite applies if DATT is negative; a DATT value of zero indicates that the vehicles arrived at the destination at the same time. A plot of DATT against the number of runs is created, so as to visualise the magnitude of DATT.

Since minor incidents on the route may cause short delays (crossing pedestrians, delivery HGVs, parking vehicles etc), values of DATT between -3 and 3 minutes are considered as ties between the two systems. This is applied as a filter to DATT, such that if DATT lies within those values, it is set as equal to zero. The new filtered DATT is called FDATT and a respective column is added to the results table. The filter is also applied to CATT and AATT, such that the values of FCATT and FAATT are obtained by setting them equal to the average value of CATT and AATT, if DATT lies between -3 and 3 minutes, or by adding/subtracting 1.5 minutes. In order to make DATT and FDATT values comparable between the two vehicles, the ratios DATT/CATT and FDATT/FCATT are computed (DATT/AATT and FDATT/FAATT would be equally appropriate). By ranking these values and by dividing by the total number of runs, cumulative distributions are obtained for both the filtered and the unfiltered data cases, showing the comparison of the results output by the two systems.

A by-product of the double-vehicle experiment is the acquisition of data for the single-vehicle experiment. Namely, as the configuration of the ARIAdNE vehicle is the same as the one used in the single-vehicle experiment and the CETA is obtained from the conventional vehicle, the measurements of the double-vehicle test are also used as additional data to the single-vehicle test.

6.3.3 Experiment parameters

The parameters of the experiments were fine-tuned and employed in the simulation experiment described in Chapter 5, and are therefore only mentioned briefly here: $\beta = 1.4$, $\zeta = 2$, $\varepsilon = 2$, $N_{max} = 3$, $r_{Emin} = 0.5$, $r_{Lmin} = 0.56$, $R_{Emin} = 0.5$, $R_{Lmin} = 0.59$, $\alpha = 0.7$ and $\gamma = 1.9$. The only difference in the parameters is the confidence level used in the calculation of the reliability indices. Namely, a confidence level of 90% is used in the simulation experiment of Chapter 5 and in the field experiments of this chapter; however, a confidence level of 95% is used in the field pre-experiment, described next. This means, that although a value of z = 1.65 is used in Chapter 5 and in the field experiments here, for the pre-experiment it is z = 1.96.

It should be mentioned here that most of the above parameters are network-specific, and the fact that they were fine-tuned for the Munich network in Chapter 5 does not guarantee that they are appropriate for the London network in the field experiments. Nevertheless, since carrying out a further fine-tuning procedure would be a tedious task and since the two networks have similar characteristics such that similar values would be expected, the same parameters used in the simulation experiments are also used here.

Having presented all the pre-requisites of the field experiments, the results of those are reported next.

6.4 Results

Following the description of the conduct of the experiments and the presentation of the analysis methods this section reports on the results obtained. Initially, a pre-experiment is carried out using the configuration of the single-vehicle experiment, followed by the actual trials.

6.4.1 Pre-experiment

In order to test the experimental configuration and analysis method so as to identify potential weaknesses prior to the conduct of the actual field trials, a pre-experiment is conducted. This involves four runs of the single-vehicle experiment in the test network; the departure times and times of arrival for each of these runs, as specified in Section 6.3.1 are shown in Table 6.4-1.

From the times recorded, travel times can be calculated (Table 6.4-2) and by dividing those by the ATT, the results become comparable (Table 6.4-3 and Figure 6.4-1a) such that cumulative distributions for each travel time are obtained and plotted on a graph (Figure 6.4-1b).

Table 6.4-1: Pre-experiment departure and arrival times

Route		DT	Conv.	ARIAdNE			ATA
Origin	Destination		CETA	AERTA	AETA	ALRTA	
Bayswater Rd	New Kent Rd	15:20	15:30	15:29	15:41	15:59	15:37
New Kent Rd	Knightsbridge	15:49	15:57	15:57	16:05	16:17	16:05
Bayswater Rd	Whitechapel High Str	16:37	16:51	16:55	17:16	17:48	17:10
Southwark Str	Knightsbridge	17:38	17:48	17:46	17:55	18:09	17:56

Table 6.4-2: Pre-experiment travel times

Route		Conv.	ARIAdNE			ATT
Origin	Destination	CETT	AERTT	AETT	ALRTT	
Bayswater Rd	New Kent Rd	10	9	21	39	17
New Kent Rd	Knightsbridge	8	8	16	28	16
Bayswater Rd	Whitechapel High Str	14	18	39	71	33
Southwark Str	Knightsbridge	10	8	17	31	18

Table 6.4-3: Pre-experiment travel time ratios

Route		Conv.	ARIAdNE		
Origin	Destination	CETT/ATT	AERTT/ATT	AETT/ATT	ALRTT/ATT
Bayswater Rd	New Kent Rd	0.5882	0.5294	1.2353	2.2941
New Kent Rd	Knightsbridge	0.5000	0.5000	1.0000	1.7500
Bayswater Rd	Whitechapel High Str	0.4242	0.5455	1.1818	2.1515
Southwark Str	Knightsbridge	0.5556	0.4444	0.9444	1.7222

As can be seen from the result of the pre-experiment the AETT is very close to the ATT, as opposed to the CETT, which is closer to the AERTT. From this it can be conjectured that ARIAdNE predicts the actual travel time fairly accurately, as opposed to the conventional system. Also, even when the ATT is not equal to the AETT, it still lies within the interval bounded by the AERTT and the ALRTT, showing that the AERTA and the ALRTA output by ARIAdNE reflect reality. An observation that can be made is that the intervals are very large compared to the travel times, i.e. that the estimates are rather conservative. The reason for this is that the confidence level used here (95%) is very high and therefore it is concluded that a lower level should be used. 90% confidence is used in the subsequent actual field experiments.

The results of the pre-experiment give an indication of the performance of ARIAdNE, however, they do not provide enough data and thus more tests are required to draw more concrete conclusions. The field experiments described in the next sections throw more light into this investigation. Finally, no hardware or software problems are discovered during the conduct of the pre-experiment, apart from a few minor issues relating to the user interface and the position of the laptop. These are fixed before the conduct of the single- and double-vehicle experiments, presented next.

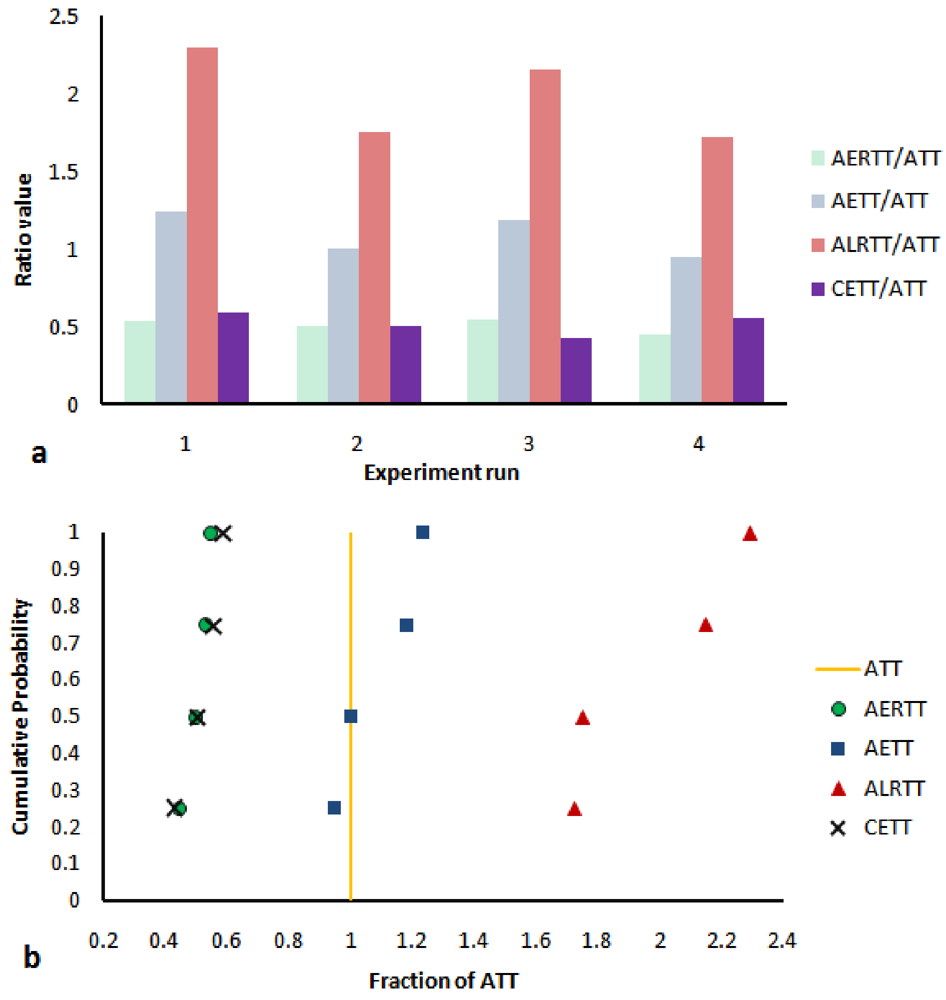


Figure 6.4-1: (a) Pre-experiment travel time ratios and (b) cumulative distributions

6.4.2 Single-vehicle experiment

49 runs of the single-vehicle experiment are carried out on the selected test network, according to the procedure described in Section 6.3.1. These are complemented by 23 runs of the double-vehicle experiment, which also supplies measurements to the single-vehicle test, making up a total of 72 runs, from which conclusions can be drawn on the accuracy of ARIAdNE.

The complete table of measured arrival times and the resulting table of travel times for each individual run have the same form as the ones created for the pre-experiment (Tables 6.4-1, 6.4-2 and 6.4-3); however they are much longer due to the higher number of observations and are therefore not shown here but are included in Appendix C. Only the mean, standard deviation and extreme values of the CETT/ATT, the AERTT/ATT, the AETT/ATT and the ALRTT/ATT distributions, as well as their cumulative distribution plots obtained by ranking them and dividing by the total number of observations, are given here. The former are displayed in Table 6.4-4, while the latter are shown in Figure 6.4-2.

Table 6.4-4: Mean, standard deviation and extreme values of travel time ratio distributions

	Conv.	**ARIAdNE**		
	CETT/ATT	**AERTT/ATT**	**AETT/ATT**	**ALRTT/ATT**
Mean	0.4922	0.5325	0.9492	1.5254
Std. Dev.	0.1228	0.1164	0.1815	0.2764
Minimum	0.2778	0.2813	0.5625	1.0000
Maximum	0.7895	0.8182	1.3636	2.0455

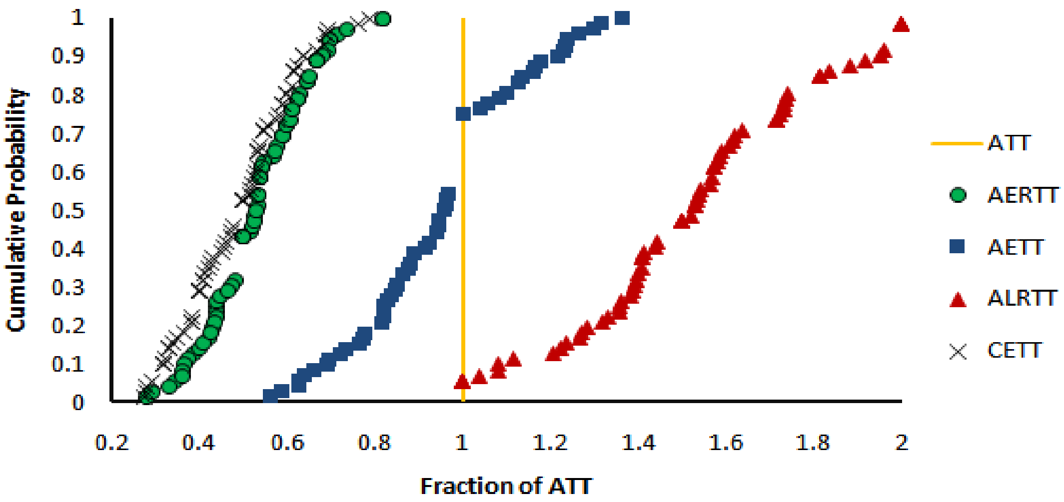


Figure 6.4-2: Cumulative distributions

From the distributions derived it can be seen that the mean AETT/ATT value is approximately 0.95, which shows that, on average, the AETT is only 5% shorter than the ATT. As expected, the AERTT always underestimates the ATT by being on average approximately 47% shorter than it (average AERTT/ATT value of 0.53), and in unexpectedly short trips without any unpredictable delays resulting in early arrivals, the ATT becomes close to the AERTT, with a maximum observed AERTT/ATT value of approximately 0.82 (i.e. the AERTT is 18% shorter than the ATT) in the experiment. On the other hand, as expected, the ALRTT always overestimates the ATT by being on average approximately 53% longer, and in trips where a combination of unpredictable delays arise, the ALRTT nears the ATT; in the experiment, the lowest ALRTT/ATT value is exactly 1, which shows that in one observation, the ATT is equal to the ALRTT.

The above results suggest that ARIAdNE yields fairly accurate results, as not only the AERTT and the ALRTT are good lower and upper bound estimates to the ATT, but also the AETT is a fairly good estimate to it. It is important to note that the bounds are not exceeded in any run of the experiment. Another feature to note is the fact that the curve of the AETT/ATT cumulative distribution is S-shaped, indicating a high concentration of values around the mean rather than around the extremes (also indicated by the low standard deviation value). More specifically, the curve shows that around 75% of the values are +/-20% of the ATT. Finally, the travel time margins computed by ARIAdNE are smaller than the ones computed in the pre-experiment, implying that the reduction in confidence (90% as opposed to 95% in the pre-experiment) yields less conservative results without affecting accuracy. Statistical significance tests for the above results are given in Appendix D.

As opposed to these observations the results obtained by the conventional system do not seem to exhibit the same level of accuracy. Namely, the mean CETT/ATT value is approximately 0.49, while the maximum and minimum values are 0.28 and 0.79. This implies that the CETT constantly underestimates the ATT, on average being 51% shorter than it. It is interesting to note that in none of the runs of the experiment did the CETT estimate the ATT correctly, with the closest observed value being around 79% of the ATT. The CETT/ATT distribution is thus significantly biased.

It could be argued, that the bias of the CETT/ATT distribution is a result of the conventional system using less accurate data than ARIAdNE (ARIAdNE makes use of ITIS FVD^{TM} records, reflecting actual measured speeds, the conventional system uses default estimates of travel times based on the speed limits provided in the map database), or of the driver being allowed to choose among multiple alternative routes in ARIAdNE as opposed to the conventional system. An approximate method to eliminate the bias would be to multiply the distribution by a specific factor, so as to shift the cumulative distribution curve to the right and bring it around the ATT line. Thus, by multiplying the CETT/ATT distribution by 2, the unbiased CETT/ATT

(UCETT/ATT) distribution is obtained and is added to the plot (Figure 6.4-3).

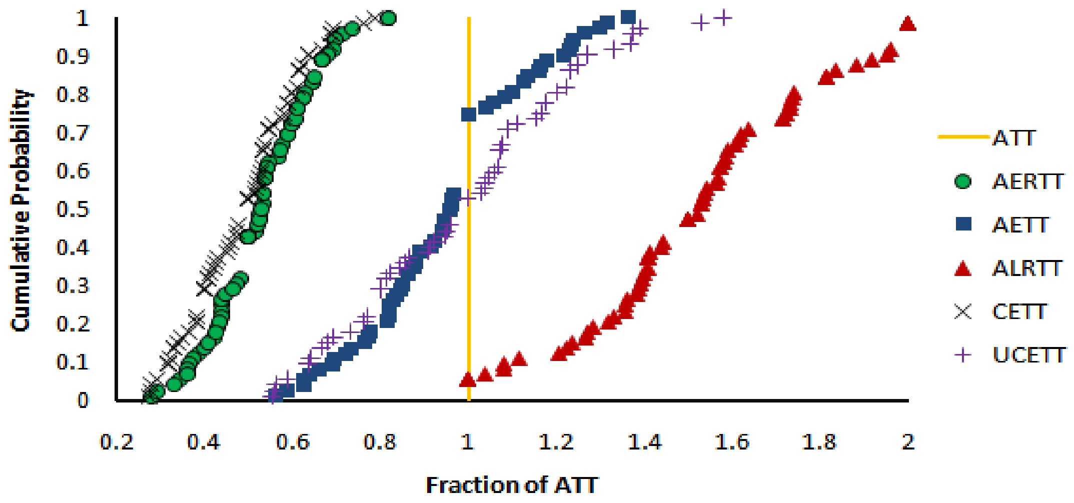


Figure 6.4-3: Cumulative distributions, including UCETT/ATT

With the bias eliminated, the UCETT/ATT distribution can be directly compared to the AETT/ATT distribution. The most important observation that can be made by examining the graph is that the AETT/ATT curve is S-shaped, as mentioned above, while the UCETT/ATT curve is more straight-shaped, with the exception of the upper extreme. This means that while the AETT/ATT distribution has a high concentration of values around the mean, this is not the case for the UCETT/ATT distribution, which again suggests that the estimates calculated by ARIAdNE are more accurate than the ones calculated by the conventional system.

On a more detailed level, it is interesting to note the ability of ARIAdNE to avoid certain links, which have been characterised as unreliable, mainly in terms of lateness. An example of this is shown in Figure 6.4-4, which is a section from the route selected in the 12th run of the experiment, from Marble Arch in West London to Tower Bridge in East London. In this section, while the fastest route (depicted by the dashed line) would use the main road (Regent's Street approaching Piccadilly Circus in London's West End), this is characterised as unreliable and thus the route suggested by ARIAdNE avoids it and guides the driver along an alternative minor road, by-passing the unreliable section. Similar observations are encountered in many other occasions throughout the conduct of the experiment.

Finally, an interesting characteristic is that, in order to ensure a greater level of reliability, ARIAdNE often suggests routes that contain a high proportion of minor roads. The reason for this is that minor roads usually exhibit more stable traffic levels throughout the day and it is thus less likely to experience unpredictable delays on them. Of course the speed that can be achieved on such roads is much lower than on main roads and therefore the AETT is generally higher than the respective value using main roads; nevertheless, routes containing many minor roads have higher reliability indices, and as ARIAdNE uses the RDIN algorithm, which is a risk-minimising method, such routes are preferable.

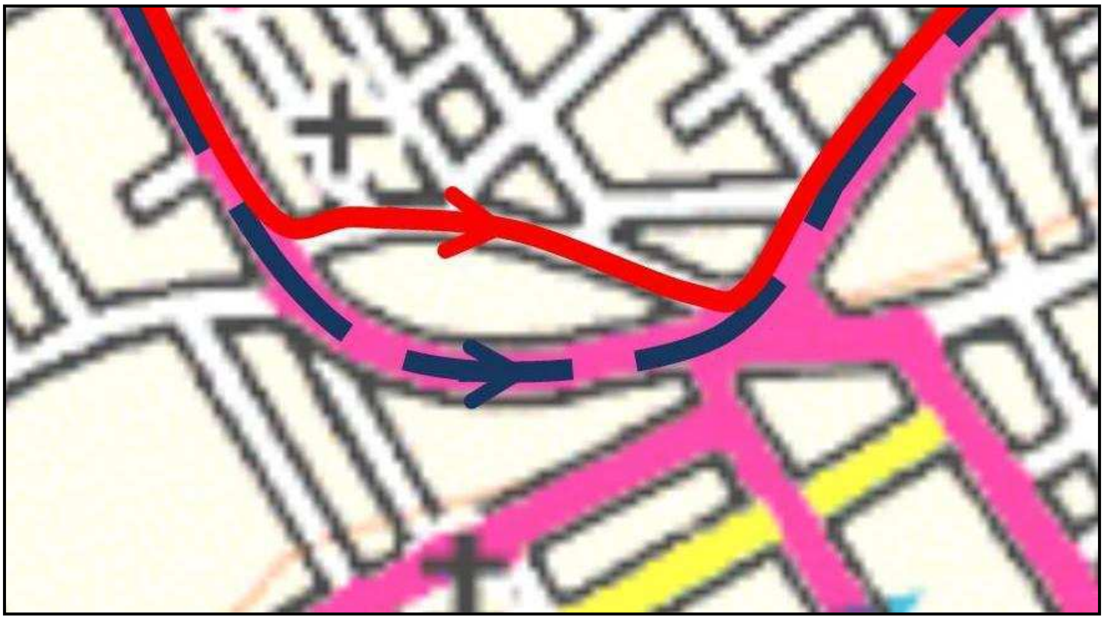


Figure 6.4-4: Avoidance of specific unreliable links by ARIAdNE

6.4.3 Double-vehicle experiment

23 runs of the double-vehicle experiment are carried out according to the procedure described in Section 6.3.2 and conclusions are drawn as regards the performance of ARIAdNE compared to the conventional navigation system. The complete table of observations is included in Appendix C; only a graph showing the trends of the CATT and AATT values throughout the experiment, superimposed for comparison purposes, is shown here (Figure 6.4-5).

As can be seen from the graph, the ARIAdNE vehicle experiences in general similar or lower travel times than the conventional vehicle (i.e. the dotted line lies below the continuous line), with only two exceptions in the 7th and the 12th run of the experiment, while there are many occasions, where the AATT is significantly lower than the CATT (i.e. a clear advantage of

ARIAdNE), such as in the 2nd, 3rd, 10th, 14th, 15th, 16th, 19th and 22nd run. In total, ARIAdNE outperforms the conventional system in 12 runs of the experiment, in 9 runs it is outperformed by it, while the remaining 2 runs are ties. These numbers mean that for less than half of the runs, namely around 39%, the conventional system outperforms ARIAdNE, which thus in turn, for around 61% of the runs, delivers similar or better results than it. This result is further demonstrated by computing the DATT from the CATT and the AATT values, and by then deriving and plotting the DATT/CATT cumulative distribution (Figure 6.4-6).

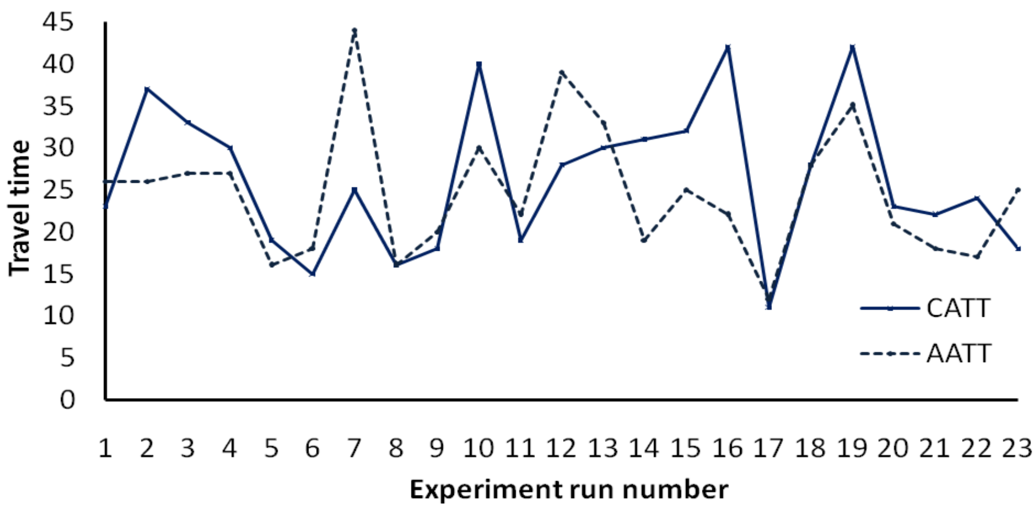


Figure 6.4-5: CATT and AATT values for each run of the experiment

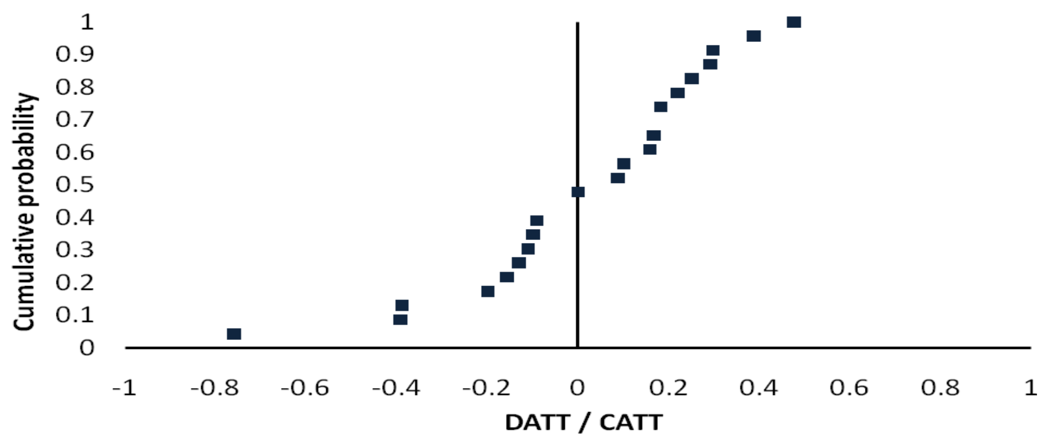


Figure 6.4-6: Cumulative distribution plot of DATT/CATT

From the plot of Figure 6.4-6 it can be seen that the resulting curve intersects the vertical axis, representing zero difference, at a cumulative probability of less than 0.5, which again suggests that ARIAdNE yields better results than the conventional system. A statistical significance test of this result is described in Appendix D.

As was mentioned in Section 6.3.2, minor incidents on a route may cause short delays, thus giving a potential marginal advantage to the one system or the other. Therefore, a filter is applied to the DATT values, such that values lying between -3 and 3 minutes are set equal to zero and the result of the corresponding observation is set to “tie”. The new filtered DATT is called FDATT and is also applied to CATT and AATT, such that the values of FCATT and FAATT are obtained. Similarly to the plot of Figure 6.4-5, a plot of FCATT and FAATT against the number of runs is shown in Figure 6.4-7.

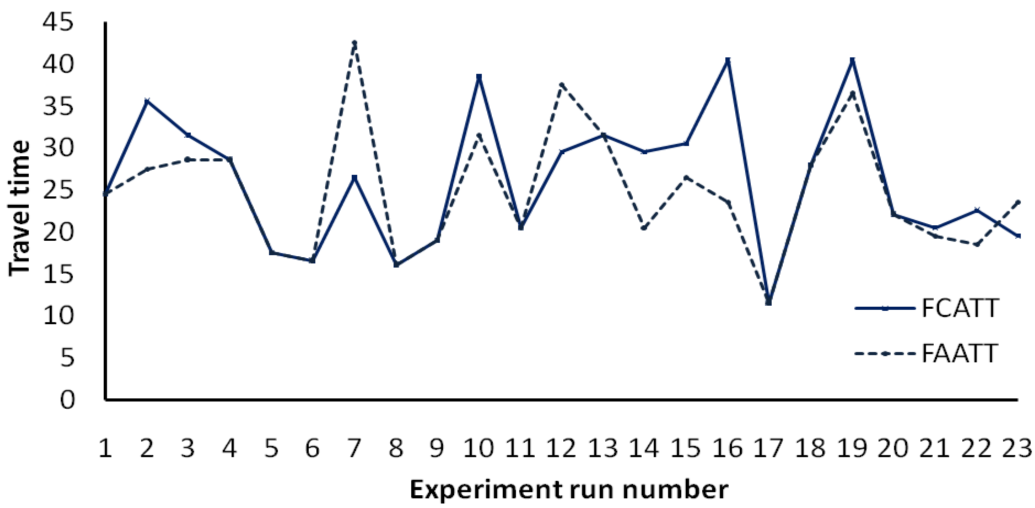


Figure 6.4-7: FCATT and FAATT values for each run of the experiment

The filtered result of the 23 runs of the double-vehicle experiment is hence the following: ARIAdNE outperforms the conventional system 9 times, there are 11 ties between the two systems, while the conventional vehicle arrives before the ARIAdNE vehicle only 3 times. This means that ARIAdNE arrives either earlier or at the same time with the conventional vehicle in 87% of the runs, thus suggesting again that it has a clear advantage. The resulting FDATT/FCATT distribution now has a higher frequency of zero values and the corresponding plot intersects the vertical axis at a higher cumulative probability value, namely 0.6. The plot is shown in Figure 6.4-8 and depicts the findings reported above. The statistical significance of this is assessed in Appendix D.

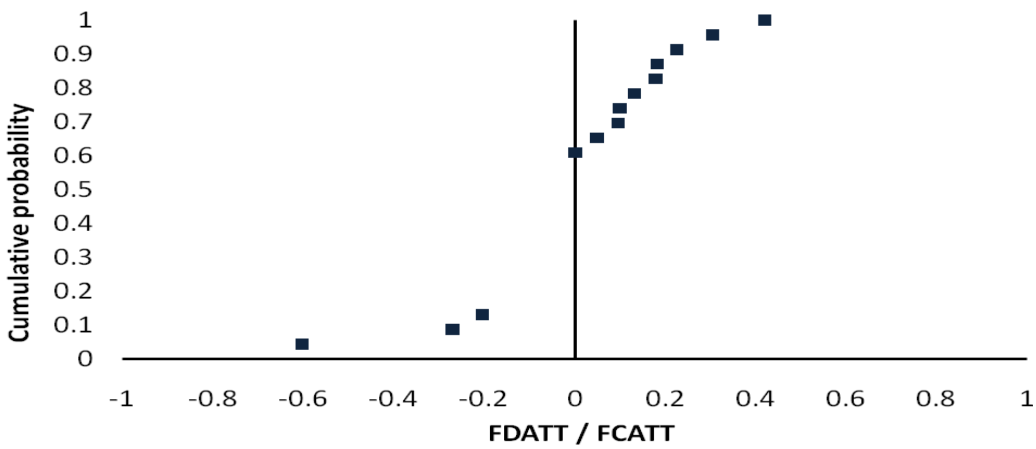


Figure 6.4-8: Cumulative distribution plot of FDATT/FCATT

Despite the overall pattern of ARIAdNE providing better results than the conventional system, it is interesting to note that it also suffers a fairly significant "defeat" during the experiment, and to be more specific, by 19 minutes. In order to explain this, one must not forget that ARIAdNE's underlying methodology, the RDIN algorithm, is a risk-minimising technique. This means that, while unpredictable delays are avoided as much as possible, it cannot be guaranteed that they will not arise at all on a route computed by ARIAdNE, nor can it be guaranteed that the conventional system will compute a route, on which all possible delays will arise. Hence, the result in question is in fact the "exception confirming the rule", as is on the other side the highest DATT value obtained, which is a "win" for ARIAdNE by 20 minutes.

On a more detailed level, it can this time be observed that, not only links characterised as unreliable are avoided by ARIAdNE, but also that they are included in the route computed by the conventional system, which does not have access to reliability information, if they have a high speed limit. An example of this is shown in Figure 6.4-9, which corresponds to the 14th run of the experiment, in which the second highest DATT value (12 minutes) is recorded, and in which route guidance is sought between Bedford Square in Bloomsbury and Tyers Street in Vauxhall (South London).

While the conventional system guides the driver along the theoretically fastest route (Shaftesbury Avenue – Trafalgar Square – Whitehall – Millbank – Lambeth Bridge – Albert Embankment), shown in red, ARIAdNE suggests a different, in theory longer, route (Kingsway – Aldwych – Waterloo Bridge – Kennington Road – Kennington Lane), completely avoiding the circled area containing links characterised as unreliable. The outcome is that congestion is encountered on these particular links, resulting in the conventional vehicle to experience great delay and arrive at the destination much later. As no incidents or other unpredictable events (e.g. roadworks) causing the congestion are neither reported nor observed in the area, it can be conjectured that the delay is due to the low lateness reliability, not considered by the conventional system but included in the route suggested by ARIAdNE.

The underlying conclusion to the double-vehicle experiment is that ARIAdNE seems to be able to compute routes that avoid areas where delays are more likely to occur, whereas this is not the case of the conventional system. The result of this is that, for the same origin and destination, a vehicle equipped with ARIAdNE arrives in general earlier than a vehicle equipped with a conventional system.

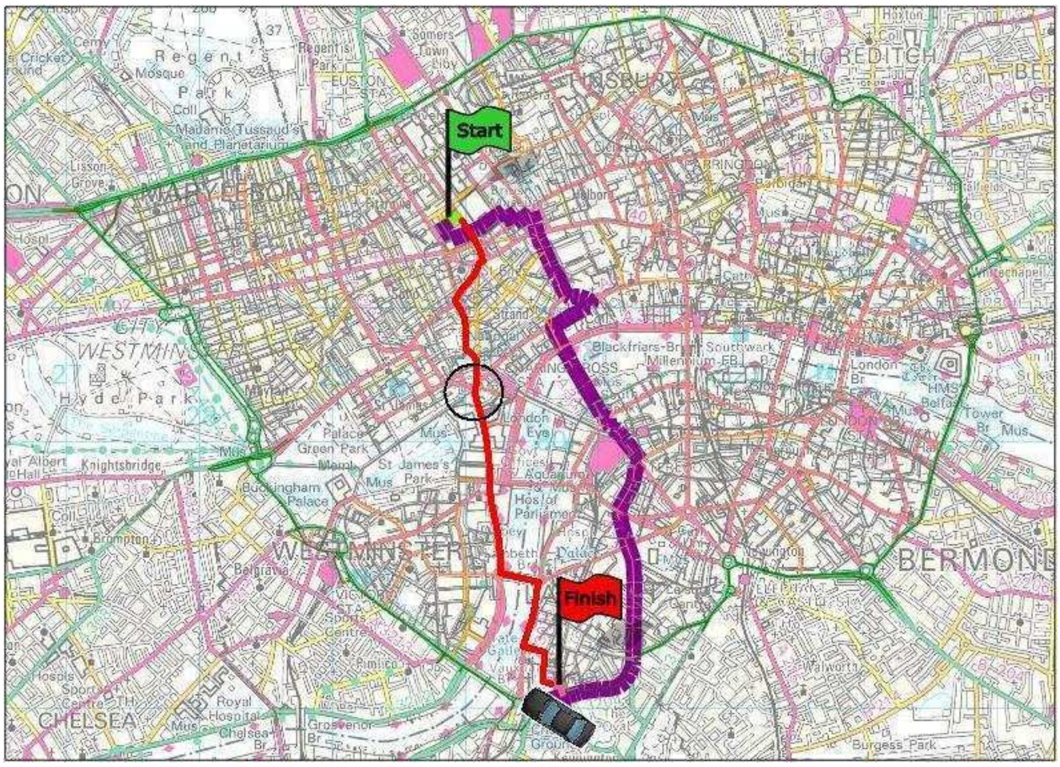


Figure 6.4-9: The advantage of the ARIAdNE route compared with the conventional route

6.5 Concluding remarks

In this chapter, the field experiments carried out in this study aiming to validate the RDIN approach presented in the previous chapters were described. Initially, a description of the methods used in data acquisition and processing was given, followed by a presentation of the procedures employed in the conduct of the experiments. Two experiments were carried out: the single-vehicle experiment and the double-vehicle experiment, the former aiming to assess the precision of the RDIN algorithm and the latter involving a direct comparison with a conventional in-vehicle navigation system, so as to show the superiority of the RDIN approach.

The results of the experiments suggest that the RDIN algorithm not only is a precise approach, which can be implemented in in-vehicle navigation systems, but also yields better results than the conventional navigation system used. It is worth noting that in 87% of the runs of the double-vehicle experiment the route computed and suggested to the driver by the RDIN algorithm, implemented in ARIAdNE, turns out to be equal to or faster than the route suggested by the conventional system. Also, the RDIN algorithm also seems to be able to compute better estimates of the expected time of arrival at the destination than the conventional system.

Hence, it can be conjectured that using the same amount of traffic information (i.e. in the absence of real-time data) the RDIN algorithm is superior to the algorithm used by the conventional system.

CHAPTER 7

Conclusions and recommendations for further research

7.1 Conclusions

The conclusions drawn from the present study are presented in this section, whereas recommendations for further research are given in the next section. Then, a section reporting on the issues arising from the practical implementation of the outcomes in an actual car navigation system is given.

The main aim of this work has been to develop, formulate and test a new reliable dynamic in-vehicle navigation algorithm. Considering the objectives listed in the introduction chapter of this study, the following have been achieved:

- A comprehensive literature review of static and time-dependent path finding algorithms has been carried out, from which it has been concluded that the A* shortest path algorithm is the most suitable one for in-vehicle navigation due to its significant advantages with respect to efficiency and tractability. Also, a method for modelling real road network features and incorporating them into route finding algorithms has been developed. Combining these, two versions of A* have been formulated: forward

A* (FA*), corresponding to the conventional A* search technique on road networks, i.e. searching from the origin to the destination, and reverse A* (RA*), which carries out the search in the opposite direction, i.e. from the destination to the origin with all the links in the network reversed. Both FA* and RA* were mathematically proven to be optimal. Additionally, a technique for incorporating time-dependence in FA* and RA* without violating the consistency condition has been derived, based on the speed of traffic on individual links.

- The topic of travel time uncertainty, variability and reliability has been thoroughly reviewed and it has been found that no existing measure of reliability is suitable for in-vehicle navigation and that a new measure meeting a number of specific requirements such as comprehensibility by the driver, ability to derive travel time values and "dimensionlessness" (so that results are independent of the time units chosen), should be defined. Based on the literature review, it has been assumed that travel time follows a log-normal distribution and two reliability indices have been defined, namely earliness and lateness, resulting from the mean and extreme values of a travel time distribution. The new reliability indices have also been expressed in terms of characteristic values of the distribution of travel speeds, and an expression for calculating the reliability indices of a route from the indices of individual links has been deduced. The inverse procedure has also been derived, such that statistical values can be obtained from reliability index values. Finally, a technique for calculating time-dependent reliability indices in accordance with the method for computing time-dependent travel times has been developed.

- A review of literature relevant to in-vehicle navigation algorithms has been conducted and it has been concluded that Chen's link penalty method is the most relevant approach for developing a reliable dynamic in-vehicle navigation algorithm, and that the present work should build on it. As a result, the reliable dynamic in-vehicle navigation (RDIN) algorithm has been developed, involving the identification of links and junction movements in the road network as unreliable using the newly defined reliability indices, and the penalisation of such unreliable elements to exclude them from the route search. Furthermore, in order to ensure that the route computed satisfies certain driver acceptability criteria, constraints have been imposed on the resulting optimisation problem. Also, a multiple-routing strategy has been adopted, as this has been proven superior for in-vehicle navigation to single-routing strategies; the selected strategy aims at computing a set of routes that are partially disjoint, i.e. do not share too many links. The RDIN algorithm thus consists of a series of consecutive time-dependent FA* and RA* runs (to make use of previously computed information from the FA* run in the RA* runs so as to reduce the search space and accelerate the search), each one with different penalties, as unreliable links are initially excluded from the search, but progressively re-included if a route satisfying the acceptability criteria cannot be found.

- An extension to the RDIN algorithm for re-routing has been developed and mathematically formulated. The new RDIN-R algorithm uses the same concept as RDIN, but is aimed at computing a route from the vehicle's current position to the destination, whenever a traffic incident is reported on the route chosen by the driver or the driver accidentally deviates from it. The FA* algorithm is used, making use of previously computed information by the RA* runs of RDIN, and is run multiple times, until a route satisfying the driver acceptability constraints is found.

- The RDIN and RDIN-R algorithms have been coded in the Visual C#.NET programming environment with the use of a special object hierarchy which allows their execution on real road networks; a user interface closely resembling commercial systems has also been developed to prepare for the testing of the new approach. The end result was the ARIAdNE software tool, which is a complete system for the provision of reliable dynamic route guidance to the driver according to the new strategy, and for the design of networks and simulation of traffic data. Using simulated data, a preliminary laboratory-based experiment has been carried out on a network in Munich, Germany. The experiment has also included a comparison of the routes proposed by the system with the ones known by local network experts. The outcome has been very promising, as the experts' route choices coincided with ARIAdNE's output, thus resulting in a first validation of the new methodology.

- Two field experiments have been conducted using ARIAdNE in a test vehicle and following its route suggestions. Employing historical floating vehicle data, travel time and reliability profiles were derived for the Central London network, so as to carry out the experiments. The first experiment involved following ARIAdNE's route suggestions and evaluating the accuracy of its arrival times estimates compared with a conventional system, while the second experiment involved a more direct comparison of ARIAdNE with the conventional system, making use of two test vehicles and observing which of the two arrived earlier at the destination more often. The results have suggested that not only the RDIN algorithm is an efficient and precise approach that can be implemented in modern in-vehicle navigation systems, but also that it outperforms conventional existing systems, available on the market.

Having summarised the conclusions drawn from this study, the next section gives recommendations for further research, both in terms of improving the new methodology, and in terms of identifying areas to which the new method may be extended.

7.2 Recommendations for further research

While the new strategy has many important advantages making a significant contribution to the field of in-vehicle navigation, there are a few points entailing possible improvements, which may be carried out in further research work on the topic.

Starting from the route finding algorithms, FA* and RA*, described in Chapter 2, a heuristic function based on the Euclidian distance has been used to estimate the travel time from any point in the network to the destination or origin respectively; this is guaranteed to underestimate the actual travel time thus ensuring optimality of the algorithm, nevertheless it would be interesting to test whether better running times can be obtained. Employing another heuristic, providing closer estimates, may result in the actual travel time to being slightly overestimated and optimality to be lost, nevertheless it may yield better running times of the algorithm. The issue of determining the best heuristic function, however, is a research topic in its own right.

Then, considering the application of link penalties for the RDIN and RDIN-R algorithms, described in Chapter 4, a possible improvement may be to increase the probability of the best routes appearing early in the search and to eliminate completely the possibility of acceptable reliable routes being ignored. An adaptive link penalty function, as opposed to the scheme employed in Chapter 4, would ensure that the amount by which penalties are reduced is proportional to the number of the previously excluded links that are to be re-included.

Furthermore, an issue that could be further looked into is the setting of the algorithm parameters. More specifically, the current parameter configuration is network-specific and the parameter values input into the RDIN and RDIN-R algorithms vary with differing network sizes and characteristics. A solution would be to either introduce an automated procedure to set and fine-tune the parameters for the road network prior to the execution of RDIN, or to conduct further research to develop a set of generic parameters, which would, however, introduce modifications to the current algorithm. Also, some parameters are driver-specific (the ones relating to the acceptability constraints), and an appropriate procedure for their setting should be introduced. This would include either allowing the user to set these parameters by himself/herself, or running an automated process that would set the parameters based on previous learning of his/her preferences.

In addition to these, a topic needing further investigation is the existence of link congestion dependence relationships. Identifying patterns, in which congestion spreads in the network so that it can be derived which links would be affected if a traffic incident occurred on particular link, would contribute to the accuracy of the RDIN and RDIN-R algorithms. This could be achieved using microscopic simulation. Also, particularly with respect to RDIN-R, the current method assumes that re-routing is needed whenever an incident on a specific link causes any delay, and that the link becomes unusable under this condition, which is a rather conservative approach. A modification of the algorithm to account for the duration of the expected delay (if this is known) could be useful.

An important point that deserves further attention is the field testing of the RDIN-R algorithm, which was not carried out in this study due to non-availability of data. The algorithm has only been tested through the simulation experiment of Chapter 5 so far, and the result obtained was encouraging, as the route suggestions given by RDIN-R coincided with the route choice of local network experts. Nevertheless, field testing of the re-routing capabilities of the new in-vehicle navigation strategy would provide definite validation evidence for the precisions and applicability of RDIN-R.

Besides improving the current approach, however, there is also the possibility of extending it to account for other related topics. Such is, for example, the use of the RDIN and RDIN-R algorithms in tour optimisation, since cars, like commercial vehicles, often perform multi-stop tours rather than simple trips, thus adding a scheduling dimension to planning. Another topic is the integration of the algorithms with a procedure for learning user preferences, so as to develop an in-vehicle navigation methodology based on personalised route generation. The integration of the algorithms with urban traffic control systems could also be of interest, with the prospect of evolving the current method to a centralised navigation strategy. Finally, an important and timely issue that could be further addressed is the consideration of the environment in in-vehicle navigation, such that RDIN and RDIN-R are extended so as to account for the environmental impact of the navigation system by guiding the driver along routes, on which this impact is low (e.g. preferring commercial to residential areas).

7.3 Practical implementation

The primary aim of this study has been to develop and test a new algorithm for in-vehicle navigation. The resulting RDIN and RDIN-R algorithms have been tested and have given some promising results, suggesting that, subject to the execution of a number of tasks of further research work, they could be implemented in an actual navigation system and used in a car. This concluding section identifies the practical issues that will arise from the implementation of RDIN and RDIN-R in on-board devices.

An issue that has to be dealt with is the positioning accuracy. In the approach developed in this study it has been assumed that the acceptable accuracy in positioning is to identify the link on which the vehicle is located. When it comes to implementing the method in an on-board navigation system, however, much more accurate positioning is required, so as to identify the exact location of the vehicle. Additionally, it is possible that errors in positioning occur, such that the co-ordinates output by the positioning system do not correspond to the actual position of the vehicle. Resolving the positioning issues of an in-vehicle navigation system is a separate research project, involving the integration of the routing algorithms with a map-matching algorithm; it is nevertheless, a very important task that needs to be carried out, should the approach be taken further and be used in an actual navigation system.

Apart from errors in positioning, however, errors may occur in a car navigation system from the map database. Namely, as road networks are subject to continuous changes due to road works and traffic engineering modifications, it is very often the case that navigation systems compute infeasible routes and provide erroneous route guidance due to errors in the map database. In order to overcome this, current technology requires the user to regularly purchase map updates, but in the future, these should automatically be downloaded and installed on the system as soon as they are available, so that it is ensured that the guidance given to the user is not affected by outdated maps.

A further issue that will arise in the practical implementation of RDIN and RDIN-R is the development of appropriate software. In this study, the algorithms have been implemented in ARIAdNE, a platform purposely developed for their laboratory and field testing, which has been designed as closely as possible to actual navigation systems. Nevertheless, ARIAdNE is still prototype software intended for conventional computers, and cannot be run on on-board units and personal handheld devices, as it has only been optimised for its intended environment. New software should be developed for the practical implementation of the algorithms, paying particular attention to efficiency and visual interface. Regarding the former, lower-level languages (e.g. assembly or binary) should be used in order to appropriately decrease computation speed; regarding the latter, a more user-friendly and commercially attractive interface should be developed (e.g. 3D view, touch-screen buttons etc.). These tasks require the input of professional programmers and software designers.

Finally, the last issue that needs to be dealt with is the interfacing of the new navigation system with formats of map and traffic data. While ARIAdNE has already been interfaced with PTV VISUM™ for map data and with ITIS FVD™ for traffic data for the purposes of experimentation, there exist numerous other suppliers of data in different countries, and to ensure transferability and usability, the new system should be interfaced with more suppliers, such as NAVTEQ™ or TeleAtlas™ for map data. Additionally, regarding traffic data, interfacing with suppliers of real-time traffic information will be required, which will also include further hardware and software work, such that the appropriate receiving equipment can be installed and used.

Publications relating to this work

Peer-reviewed journal publications

Kaparias, I., Bell, M.G.H. and Belzner, H. (2008), "Quantifying and measuring travel time reliability for in-vehicle navigation systems", Journal of Intelligent Transport Systems (In press).

Kaparias, I., Bell, M. G. H., Bogenberger, K. and Chen, Y. (2007), "An approach to time-dependence and reliability in dynamic route guidance", Transportation Research Record, vol. 2039, pp. 32-41.

Kaparias, I., Bell, M. G. H., Chen, Y. and Bogenberger, K. (2007), "ICNavS: A tool of reliable dynamic route guidance", IET Intelligent Transport Systems, vol. 1, no. 4, pp. 225-233.

Conference proceedings

Kaparias, I. and Bell, M. G. H. (2008), "In-vehicle navigation using a time-dependent reliable A* search", Presented at the the 18th Triennial Conference of the International Federation of Operational Research Societies (IFORS), Sandton, South Africa.

Kaparias, I., Bell, M. G. H. and Belzner, H. (2008), "Field testing of a reliable in-vehicle navigation algorithm", Presented at the 10th International Conference on Applications of Advanced Technologies in Transportation, Athens, Greece.

Kaparias, I., Bell, M. G. H. and Belzner, H. (2008), "Quantifying and measuring travel time reliability for in-vehicle navigation systems", Presented at the 87th Annual Meeting of the Transportation Research Board, Washington, DC, USA.

Kaparias, I. (2007), "A time-dependent risk-averse algorithm for reliable in-vehicle navigation", Presented at the 39th Annual Conference of the Universities' Transport Study Group, Harrogate, UK.

Kaparias, I., Bell, M. G. H., Bogenberger, K. and Chen, Y. (2007), “An approach to time-dependence and reliability in dynamic route guidance”, Presented at the 86th Annual Meeting of the Transportation Research Board, Washington, DC, USA.

Kaparias, I., Bell, M. G. H., Chen, Y. and Bogenberger, K. (2006), “ICNavS: A tool for reliable route guidance”, Presented at the 13th World Congress on Intelligent Transportation Systems, London, UK.

Kaparias, I. (2006), “Risk-averse dynamic route guidance: Observations from the application on a real network”, Presented at the 38th Annual Conference of the Universities’ Transport Study Group, Dublin, Ireland.

Research reports

Bell, M. G. H., Kaparias, I. and Park, K. (2008), “OFFENSIVE final project report”, BMW Group, Munich, Germany.

Bogenberger, K., Bell, M. G. H., Kaparias, I., Park, K., Bertini, R. L., Boice, S., Busch, F., Leonhardt, A. and Spangler, M. (2006), “Offensive Project Summary”, BMW Group, Munich, Germany.

References

Adbel-Aty, A., Kitamura, R. and Jovanis, P. P. (1995), "Investigating the effect of travel time variability on route choice using repeated measurement stated preference data", Transportation Research Record, vol. 1493, pp. 39-45.

Ahmad, S. H. (1982), "A simple technique for computing network reliability", IEEE Transactions on Reliability, vol. R31, no. 1, pp. 41-44.

Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1993), "Network flows: Theory, Algorithms and Applications", Prentice Hall.

Ahuja, R. K., Orlin, J. B., Pallotino, S. and Scutella, M. G. (2002), "Minimum time and minimum cost-path problems in street networks with periodic traffic lights", Transportation Science, vol. 36, no. 3, pp. 236-336.

Arnott, R., De Palma, A. and Lindsey, R. (1991), "Does providing information to drivers reduce traffic congestion?", Transportation Research A, vol. 25A, no. 5, pp. 309-318.

Asakura, Y., Hato, E. and Kashiwadani, M. (2001), "Stochastic network design problem: an optimal link investment model for reliable network", Proceedings 1st International Symposium on Transportation Network Reliability, Kyoto, Japan, pp. 245-259.

Averbakh, I. and Lebedev, V. (2004), "Interval data minmax regret network optimization problems", Discrete Applied Mathematics, vol. 138, pp. 289-301.

Awasthi, A., Lechevallier, Y., Parent, M. and Proth, J.-M. (2003), "Using data analysis to approximate fastest paths on urban networks", Institut National de Recherche en Informatique et Automatique, 4961.

Bander, J. L. and White, C. C. I. (1991), "A new route optimization algorithm for rapid decision support", Vehicle Navigation and Information Systems Conference, pp. 709-728.

Bander, J. L. and White, C. C. I. (1998), "A heuristic search algorithm for path determination with learning", IEEE Transactions on Systems, Man and Cybernetics, vol. 28, no. 1, pp. 131-134.

Banerjee, G. and Sidhu, D. (2002), "Comparative analysis of path computation techniques for MPLS traffic engineering", Computer Networks, vol. 40, pp. 149-165.

Bates, J., Black, I., Fearon, J., Gilliam, C. and Porter, S. (2002), "Supply models for use in modelling the variability of journey times on the highway network", AET - Proceedings of the European Transport Conference 2002, Cambridge, UK.

Bates, J., Polak, J. W., Jones, P. and Cook, A. (2001), "The valuation of reliability for personal travel", Transportation Research E, vol. 37, pp. 191-229.

Bell, M. G. H., Cassir, C., Iida, Y. and Lam, W. H. K. (1999), "A sensitivity based approach to network reliability assessment", Proceedings 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, pp. 283-300.

Bell, M. G. H., Chen, Y. and Bogenberger, K. (2005), "Robust route guidance in road networks", International Workshop of Urban Road Network Reliability.

Bell, M. G. H. and Iida, Y. (1997), "Network reliability," in "Transportation network analysis", Wiley and Sons, Chichester, pp. 179-192.

Bellman, R. (1958), "On a routing problem", Quarterly of Applied Mathematics, vol. 16, no. 1, pp. 87-90.

Ben-Akiva, M., De Palma, A. and Kaysi, I. (1991), "Dynamic network models and driver information systems", Transportation Research A, vol. 25A, no. 5, pp. 251-266.

Berg Insight (2005), "Mobile personal navigation services".

Berg Insight (2007a), "Connected navigation services".

Berg Insight (2007b), "Mobile maps and navigation".

Black, I., Bates, J., Fearon, J., Gilliam, C., Kerr, A. and Porter, S. (2004), "Research programme measures the variability of urban journey times", Traffic Engineering and Control, vol. 45, no. 11, pp. 409-415.

Blue, V. J., Adler, J. L. and List, G. F. (1997), "Real-time multiple-objective path search for in-vehicle route guidance systems", Transportation Research Record, vol. 1588, pp. 10-17.

Boffey, T. B. (1984), "Graph theory in operations research".

Bonsall, P. (1992), "Drivers' acceptance of route guidance advice: an international comparison", 3rd International Conference on Vehicle Navigation and Information Systems (VNIS 1992).

Boyce, D., Kirson, A. and Schofer, J. L. (1991), "Design and implementation of ADVANCE: The Illinois dynamic navigation and route guidance demonstration program", 1991 SAE Technical Paper Series, 912786, pp. 415-426.

Boyce, D., Rouphail, N. and Kirson, A. (1993), "Estimation and measurement of link travel times in the ADVANCE project", Proceedings of the IEEE-IEE Vehicle Navigation and Information Conference 1993, Ottawa, Canada.

Catling, I. and Belcher, P. (1989), "Autoguide - route guidance in the United Kingdom", Vehicle Navigation and Information Systems Conference 1989, Toronto, ON, Canada, pp. 467-473.

Cerulli, R., Festa, P. and Raiconi, G. (2001), "Shortest paths in randomly time varying networks", IEEE Intelligent Transportation Systems Conference Proceedings, Oakland, USA, pp. 854-859.

Chabini, I. and Lan, S. (2002), "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks", IEEE Transactions on Intelligent Transportation Systems, vol. 3, pp. 60-74.

Chen, K., Yu, L., Guo, J. and Wen, H. (2007), "Characteristics analysis of road network reliability in Beijing based on the data logs from taxis", TRB 2007 Annual Meeting CD-ROM, Washington DC, USA.

Chen, T.-Y., Chang, H.-L. and Tzeng, G.-H. (2001), "Using a weight-assesing model to identify route choice criteria and information effects", Transportation Research A, vol. 35, pp. 197-224.

Chen, Y.-L. and Yang, H.-H. (2000), "Shortest paths in traffic light networks", Transportation Research B, vol. 34, pp. 241-253.

Chen, Y.-L. and Yang, H.-H. (2003), "Minimization of travel time and weighted number of stops in a traffic-light network", European Journal of Operational Research, vol. 144, pp. 565-580.

Chen, Y. and Bell, M. G. H. (2005), "Risk-averse autonomous route guidance by a constrained A* search", Unpublished paper, Imperial College London.

Chen, Y., Bell, M. G. H. and Bogenberger, K. (2005a), "Reliable pre-trip multi-path planning and dynamic adaptation for a centralized road navigation system", ITSC 2005 - 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria.

Chen, Y., Bell, M. G. H., Wang, D. and Bogenberger, K. (2006), "Risk-averse time-dependent route guidance by constrained dynamic A* search in decentralized system architecture", Transportation Research Record, vol. 1944, pp. 51-57.

Chen, Y., Kaparias, I., Bell, M. G. H. and Bogenberger, K. (2005b), "Reliable autonomous route guidance by a constrained A* search considering intersection delays", The Reliability of Traveling and the Robustness of Transport Systems, TRAIL Research School, Delft, The Netherlands, pp. 69-92.

Chen, Y. and Wang, D. (2007), "Optimum path algorithm for in-vehicle navigation system in distributed structure with un-completed dynamic information", Journal of Highway and Transportation Research and Development, vol. 2, no. 1, pp. 70-75.

Cherkassky, B. V., Goldberg, A. V. and Radzik, T. (1994), "Shortest paths algorithms: theory and experimental evaluation", Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms, pp. 516-525.

Clark, S. and Watling, D. (2005), "Modelling network travel time reliability under stochastic demand", Transportation Research B, vol. 39, pp. 119-140.

Climaco, J. C. N. and Martins, E. Q. V. (1982), "A bicreterion shortest path algorithm", European Journal of Operational Research, vol. 11, pp. 399-404.

Cooke, K. L. and Halsey, E. (1966), "The shortest route through a network with time-dependent internodal transit times", Journal of Mathematical Analysis and Applications, vol. 14, no. 3, pp. 493-498.

Cowan, K. and Gates, G. (2003), "Floating vehicle data system - A smart move", Smart Moving, Birmingham.

Crittin, F. and Bierlaire, M. (2001), "New algorithmic approaches for the anticipatory route guidance generation problem", 1st Swiss Transport Research Conference, Monte Verita / Ascona.

Crittin, F. and Bierlaire, M. (2003), "Solving the anticipatory route guidance generation problem using a generalization of secant methods", 3rd Swiss Transport Research Conference, Monte Verita / Ascona.

Daganzo, C. F. (2002), "Reversibility of the time-dependent shortest path problem", Transportation Research B, vol. 36, pp. 665-668.

Dandy, G. C. and McBean, E. A. (1984), "Variability of individual travel time components", ASCE Journal of Transportation Engineering, vol. 110, no. 3, pp. 340-357.

Dantzig, G. B. (1960), "On the shortest route through a network", Management Science, vol. 6, pp. 187-190.

De Champeaux, D. (1983), "Bidirectional heuristic search again", Journal of the Association for Computing Machinery, vol. 30, no. 1, pp. 22-32.

De Champeaux, D. and Sint, L. (1977), "An improved bidirectional heuristic search algorithm", Journal of the Association for Computing Machinery, vol. 24, no. 2, pp. 177-191.

Deflorio, F. P. (2003), "Evaluation of a reactive dynamic route guidance strategy", Transportation Research C, vol. 11, no. 5, pp. 375-388.

Dijkstra, E. W. (1959), "A note on two problems in connexion with graphs", Numerische Mathematik, vol. 1, pp. 269-271.

Dong, J., Mahmassani, H. S. and Lu, C.-C. (2006), "How reliable is this route? - Predictive travel time and reliability for anticipatory traveler information systems", Transportation Research Record, vol. 1980, pp. 117-125.

Dreyfus, S. E. (1969), "An appraisal of some shortest-path algorithms", Operations Research, vol. 17, pp. 395-412.

Du, Z.-P. and Nicholson, A. (1997), "Degradable transportation systems: sensitivity and reliability analysis", Transportation Research B, vol. 32, no. 3, pp. 225-237.

Easa, S. M. (1985), "Shortest route algorithm with movement prohibitions", Transportation Research B, vol. 19B, no. 3, pp. 197-208.

Eleftheriadou, L. and Cui, X. (2007), "A framework for defining and estimating travel time reliability", TRB 2007 Annual Meeting CD-ROM, Washington DC, USA.

Eppstein, D. (1997), "Finding the k shortest paths", SIAM Journal of Computing, vol. 28, no. 2, pp. 652-673.

Ericsson, E., Larsson, H. and Brundell-Freij, K. (2006), "Optimizing route choice for lowest fuel consumption - Potential effects of a new driver support tool", Transportation Research C, vol. 14, pp. 369-383.

Erkut, E. and Verter, V. (1998), "Modeling of transport risk for hazardous materials", Operations Research, vol. 46, no. 5, pp. 625-642.

Floyd, R. W. (1962), "Algorithm 97: Shortest path", Communications of the Association for Computing Machinery, vol. 5, no. 6, p. 345.

Ford, L. and Fulkerson, D. R. (1962), "Flows in networks", Princeton University Press.

French, R. L. and Lang, G. M. (1973), "Automatic route control system", IEEE Transactions on Vehicular Technology, vol. 22, no. 2, pp. 36-41.

Frith, B. (2000), "Analysis of journey time variability based on Trafficmaster data for the A12", DETR Project Report, PR/T/160/2000.

Fu, L. (2001), "An adaptive routing algorithm for in-vehicle route guidance systems with real-time information", Transportation Research B, vol. 35, pp. 749-765.

Fu, L. and Rilett, L. R. (1998), "Expected shortest paths in dynamic and stochastic traffic networks", Transportation Research B, vol. 32, no. 7, pp. 499-516.

Fu, L., Sun, D. and Rilett, L. R. (2006), "Heuristic shortest path algorithms for transportation applications: State of the art", Computers and Operations Research, vol. 33, pp. 3324-3343.

Garey, M. R. and Johnson, D. S. (1979), "Computers and intractability: a guide to the theory of NP-completeness", W.H. Freeman and Company, New York.

Goldberg, A. V. and Harrelson, C. (2005), "Computing the shortest path: A* search meets graph theory", 16th Annual ACM-SIAM Symposium on Discrete Algorithms.

Habbal, M. B., Koutsopoulos, H. N. and Lerman, S. R. (1994), "A decomposition algorithm for the all-pairs shortest path problem on massively parallel computer architectures", Transportation Science, vol. 28, no. 4, pp. 292-308.

Hall, R. (1986), "The fastest path through a network with random time-dependent travel times", Transportation Science, vol. 20, no. 3, pp. 182-188.

Hart, P. E., Nilsson, N. J. and Raphael, B. (1968), "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems, Science and Cybernetics, vol. 4, no. 2, pp. 100-107.

Hendrickson, C. and Planck, E. (1984), "The flexibility of departure times for work trips", Transportation Research A, vol. 18A, no. 1, pp. 25-36.

Herman, R. and Lam, T. (1974), "Trip time characteristics of journeys to and from work", Proceedings 6th International Symposium on Transportation and Traffic Theory, Sydney, New South Wales, Australia, pp. 57-85.

Heti, G. (1993), "Travelguide: Ontario's route guidance system demonstration", Proceedings of the IEEE-IEE Vehicle Navigation and Information Conference 1993, Ottawa, Canada.

Hock, Q. K. and Srikanthan, T. (2001), "Improving hierarchical route computations for roadway networks", IEEE Intelligent Transportation Systems Conference, Oakland, USA, pp. 112-117.

Hoffmann, G. and Janko, J. (1990), "Travel times as a basic part of the LISB guidance strategy", 3rd IEEE International Conference on Road Traffic Control, London, UK, pp. 6-10.

Holzer, M. (2003), "Hierarchical speed-up techniques for shortest-path algorithms", Diplom , University of Konstanz.

Horn, M. E. T. (2000), "Efficient modeling of travel in networks with time-varying link speeds", Networks, vol. 36, no. 2, pp. 80-90.

Hu, J. and Bell, M. G. H. (2007), "Spatial econometric modelling of congestion prediction in in-vehicle route guidance", Spatial Econometrics Association Conference 2007, Cambridge, UK.

Hu, J., Kaparias, I. and Bell, M. G. H. (2007), "Identification of link congestion dependence patterns for dynamic route guidance", 14th World Congress on Intelligent Transportation Systems, Beijing, China.

Huang, Y.-W., Jing, N. and Rundensteiner, E. A. (1995), "Route guidance support in intelligent transportation systems: an encoded path view approach", University of Michigan Technical Report 1995.

ITIS Holdings PLC Website (2007), http://www.itisholdings.com.

IEEE (1990), "IEEE Standard Glossary of Software Engineering Terminology", IEEE Std 610.12-1990.

Ilgaz, R. D., Gates, G. and James, L. (2002), "Floating vehicle data for network management: A pilot study", 9th World Congress on Intelligent Transportation Systems, Chicago.

J.D.Power and Associates (2006), "2006 Navigation usage and satisfaction study".

Jackson, W. B. and Jucker, J. V. (1981), "An empirical study of travel time variability and travel choice behaviour", Transportation Science, vol. 16, no. 4, pp. 460-475.

Jacob, R., Marathe, M. V. and Nagel, K. (1998), "A computational study of routing algorithms for realistic transportation networks".

Jagadeesh, G. R., Srikanthan, T. and Quek, K. H. (2002), "Heuristic techniques for accelerating hierarchical routing on road networks", IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 4, pp. 301-309.

Jahn, O., Möhring, R. H. and Schulz, A. S. (1999), "Optimal routing of traffic flows with length restrictions in networks with congestion", 658/1999.

Jeffery, D. J. (1981), "The potential benefits of route guidance", TRL Report, LR997.

Jeong, Y. J., Hong, Y.-C. and Kim, T. J. (2007), "A flexible multi-path search algorithm for multi-purpose location-based activities", TRB 2007 Annual Meeting CD-ROM, Washington D.C., USA.

Jimenez, V. M. and Marzal, A. (1999), "Computing the K shortest paths: a new algorithm and an experimental comparison", Lecture Notes in Computer Science, vol. 1688, pp. 15-29.

Kang, S., Oh, S. and Kim, T. J. (2006), "Heuristic algorithm for solving a multimodal location-based concierge service problem", Transportation Research Record, vol. 1972, pp. 123-132.

Karasan, O. E., Pinar, M. C. and Yaman, H. (2002), "The robust shortest path problem with interval data", Computers and Operations Research.

Karimi, H. A. (1996), "Real-time optimal-route computation: a heuristic approach", Journal of Intelligent Transportation Systems, vol. 3, no. 2, pp. 111-127.

Kasperski, A. and Zielinski, P. (2006), "The robust shortest path problem in series-parallel multidigraphs with interval data", Operations Research Letters, vol. 34, pp. 69-76.

Kaufmann, D. E. and Smith, R. L. (1993), "Fastest paths in time-dependent networks for IVHS application", IVHS Journal, vol. 1, pp. 1-12.

Kelton, W. D. and Law, A. M. (1978), "A mean-time comparison of algorithms for the all-pairs shortest path problem with arbitrary arc lengths", Networks, vol. 8, pp. 97-106.

Kerner, B. S., Rehborn, H., Aleksic, M. and Haug, A. (2005), "Traffic prediction systems in vehicles", ITSC 2005 - 8th International IEEE Conference on Intelligent Transportation Systems, pp. 251-256.

Kim, J. and Jung, S. (2002), "A dynamic window-based approximate shortest path re-computation method for digital road map databases in mobile environments", Lecture Notes in Computer Science, vol. 2510, pp. 711-720.

Kirby, R. F. and Potts, R. B. (1969), "The minimum route problem for networks with turn penalties and prohibitions", Transportation Research B, vol. 3, pp. 397-408.

Koenig, S., Likhachev, M. and Furcy, D. (2004), "Lifelong planning A*", Artificial Intelligence, vol. 155, pp. 93-146.

Korf, R. E. (1985), "Depth-first iterative-deepening: an optimal admissible tree search", Artificial Intelligence, vol. 27, pp. 97-109.

Kouvelis, P. and Yu, G. (1997), "Robust discrete optimization and its applications", Kluwer Academic Publishers, Boston.

Lam, T. C. and Small, K. A. (2001), "The value of time and reliability: measurement from a value pricing experiment", Transportation Research E, vol. 37, pp. 231-251.

Lark, J. W. I., White, C. C. I. and Syverson, K. (1995), "A best-first seach algorithm guided by a set-valued heuristic", IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 7, pp. 1097-1101.

Lee, C.-K. (1994), "A multiple-path routing strategy for vehicle route guidance systems", Transportation Research C, vol. 2, no. 3, pp. 185-195.

Lefebvre, N. and Balmer, M. (2007), "Fast shortest path computation in time-dependent traffic networks", Swiss Transport Research Conference 2007.

Li, R., Rose, G. and Sarvi, M. (2007), "Modelling and estimation of travel time variability", TRB 2007 Annual Meeting CD-ROM, Washington DC, USA.

Liu, B. (1997), "Route finding by using knowledge about the road network", IEEE Transactions on Systems, Man and Cybernetics A, vol. 27, no. 4, pp. 436-448.

Liu, G. and Ramakrishnan, K. G. (2001), "A* prune: an algorithm for finding K shortest paths subject to multiple constraints", Proceedings of the 20th Joint Conference of the IEEE Computer & Communications Societies (INFOCOM 2001): Twenty years into the communications odyssey, pp. 743-749.

Liu, H. X., Recker, W. and Chen, A. (2004), "Uncovering the contribution of travel time reliability to dynamic route choice using real-time loop data", Transportation Research A, vol. 38, pp. 435-453.

Lomax, T., Schrank, D., Turner, S. and Margiotta, R. (2003), "Selecting travel reliability measures", Texas Transportation Institute, 474360-1.

Lomax, T., Turner, S. and Margiotta, R. (2001), "Monitoring urban roadways in 2000: Using archived operations data for reliability and mobility measurement", Federal Highway Administration - Operations Core Business Unit, FHWA-OP-02-029.

Lu, J., Ban, X., Qiu, Z., Yang, F. and Ran, B. (2005), "Robust route guidance model based on advanced traveler information systems", Transportation Research Record, vol. 1935, pp. 1-7.

Martins, E. Q. V. (1984a), "An algorithm for ranking paths that may contain cycles", European Journal of Operational Research, vol. 18, no. 1, pp. 123-130.

Martins, E. Q. V. (1984b), "On a multicriteria shortest path problem", European Journal of Operational Research, vol. 16, no. 2, pp. 236-245.

Martins, E. Q. V., Pascoal, M. M. B. and Dos Santos, J. L. E. (1999), "Deviation algorithms for ranking shortest paths", International Journal of Foundations of Computer Science, vol. 10, no. 3, pp. 247-261.

Meyer, U. (2003), "Average-case complexity of single-source shortest-paths algorithms: lower and upped bounds", Journal of Algorithms, vol. 48, pp. 91-134.

Miller-Hooks, E. (2001), "Adaptive least-expected time paths in stochastic, time-varying transportation and data networks", Networks, vol. 37, no. 1, pp. 35-52.

Miller-Hooks, E. and Mahmassani, H. S. (1998), "Least possible time paths in stochastic time-varying networks", Computers and Operations Research, vol. 25, no. 12, pp. 1107-1125.

Miller-Hooks, E. and Mahmassani, H. S. (2000), "Least expected time paths in stochastic, time-varying transportation networks", Transportation Science, vol. 34, no. 2, pp. 198-215.

Miller-Hooks, E. and Mahmassani, H. S. (2003), "Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks", European Journal of Operational Research, vol. 146, pp. 67-82.

Mogridge, M. and Fry, S. (1984), "The variability of car journeys on a particular route in central London", Traffic Engineering and Control, vol. 25, pp. 510-511.

Montemanni, R. and Gambardella, L. M. (2004), "An exact algorithm for the robust shortest path problem with interval data", Computers and Operations Research, vol. 31, pp. 1667-1680.

Montemanni, R., Gambardella, L. M. and Donati, A. V. (2004), "A branch and bound algorithm for the robust shortest path problem with interval data", Operations Research Letters, vol. 32, pp. 225-232.

Montgomery, F. O. and May, A. D. (1987), "Factors affecting travel times on urban radial routes", Traffic Engineering and Control, vol. 28, no. 9, pp. 452-458.

Ni, J. and Deakin, E. (2002), "On-board advanced traveler information systems", University of California Transportation Center Report 631.

Nicholson, T. A. J. (1966), "Finding the shortest route between two points in a network", The Computer Journal, vol. 9, no. 3, pp. 275-280.

Noland, R. B. and Polak, J. W. (2002), "Travel time variability: a review of theoretical and empirical issues", Transport Reviews, vol. 22, no. 1, pp. 39-54.

Noland, R. B. and Small, K. A. (1995), "Travel-time uncertainty, departure time choice, and the cost of morning commutes.", Transportation Research Record, vol. 1493, pp. 150-158.

Noland, R. B., Small, K. A., Koskenoja, P. M. and Chu, X. (1998), "Simulating travel reliability", Regional Science and Urban Economics, vol. 28, pp. 535-564.

Ochieng, W. Y., Quddus, M. A. and Noland, R. B. (2004), "Map-matching in complex urban road networks", Revista Brasileira de Cartografia (Brazilian Journal of Cartography), vol. 55, no. 2, pp. 1-18.

Oh, J.-S. and Chung, Y. (2006), "Calculation of travel time variability from loop detector data", Transportation Research Record, vol. 1945, pp. 12-23.

Orda, A. and Rom, R. (1990), "Shortest-path and minimum-delay algorithms in networks with time-dependent edge length", Journal of the Association for Computing Machinery, vol. 37, no. 3, pp. 607-625.

Personal communication with Ian Forshaw, ITIS Holdings PLC (2007).

Park, C.-K., Sung, K., Doh, S. and Park, S. (2001), "Finding a path in the hierarchical road networks", IEEE Intelligent Transportation Systems Conference, Oakland, USA, pp. 936-942.

Park, D., Kim, H., Hong, S., Lee, C. and Choi, C. (2007a), "Location-based dynamic route guidance system of Korea: system design, algorithms and initial results", Transportation Research Record, vol. 2024 (In press).

Park, D. and Rilett, L. R. (1998), "Forecasting multiple-period freeway link travel times using modular neural networks", Transportation Research Record, vol. 1617, pp. 163-170.

Park, D., Rilett, L. R. and Han, G. (1999), "Spectral basis neural networks for real-time travel time forecasting", ASCE Journal of Transportation Engineering, vol. 125, no. 6, pp. 515-523.

Park, D., Sharma, S. L., Rilett, L. R. and Chang, M. (2002), "Identifying multiple reasonable alternative routes", Transportation Research Record, vol. 1783, pp. 111-118.

Park, K., Bell, M. G. H., Kaparias, I. and Bogenberger, K. (2007b), "Adaptive route choice model for intelligent route guidance", Transportation Research Record, vol. 2000, pp. 88-97.

Park, K., Bell, M. G. H., Kaparias, I. and Bogenberger, K. (2007c), "Learning user preferences of route choice behaviour for adaptive route guidance", IET Intelligent Transport Systems, vol. 1, no. 2, pp. 159-166.

Pattanamekar, P., Park, D., Rilett, L. R., Lee, J. and Lee, C. (2003), "Dynamic and stochastic shortest path in transportation networks with two components of travel time uncertainty", Transportation Research C, vol. 11, pp. 331-354.

Pavlis, Y. and Papageorgiou, M. (1999), "Simple decentralized feedback strategies for route guidance in traffic networks", Transportation Science, vol. 33, no. 3, pp. 264-278.

Pearl, J. (1984), "Heuristics: Intelligent search strategies for computer problem solving", Addison-Wesley.

Pohl, I. (1971), "Bi-directional search", Machine Intelligence, vol. 6, pp. 127-140.

Polenta, T. (2005), "Incorporating uncertainty in real-time route guidance systems", PhD Thesis, Nottingham Trent University.

Polenta, T. and Hartley, J. K. (2003), "A comparative study of stochastic 'least-time' path algorithms in the context of the Nottingham urban network", Proceedings of the UKSim 2003, Cambridge, pp. 194-200.

Polus, A. (1979), "A study of travel time and reliability on arterial routes", Transportation, vol. 8, pp. 141-151.

Porter, S. (2001), "Motorway variability models and their extension to all purpose roads", Seminar on Travel Time Variability, 2 July 2001.

Pu, J., Manning, E., Shoja, G. C. and Srinivasan, A. (2001), "A new algorithm to compute alternate paths in reliable OSPF (ROSPF)", Proceedings International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2001), pp. 299-304.

Quddus, M. A., Ochieng, W. Y. and Noland, R. B. (2007), "Current map-matching algorithms for transport applications: State-of-the-art and future research directions", Transportation Research C, vol. 15, pp. 312-328.

Quek, K. H. and Srikanthan, T. (2002), "Towards efficient roadway network topology with preprocessing", pp. 511-516.

Rakha, H., El-Shawarby, I., Arafeh, M. and Dion, F. (2006), "Estimating path travel-time reliability", ITSC 2006 - 9th International IEEE Conference on Intelligent Transportation Systems, Toronto, Canada, pp. 236-241.

Rakha, H. and Wang, Z. (2005), "Estimating traffic stream space mean speed and reliability from dual- and single-loop detectors", Transportation Research Record, vol. 1925, pp. 38-47.

Rillings, J. H. and Lewis, J. W. (1991), "TravTek", 1991 SAE Technical Paper Series, 912819, pp. 729-734.

Robinson, S. (2005), "The development and application of an urban link travel time model using data derived from inductive loop detectors", PhD Thesis, Imperial College London.

Robinson, S. and Polak, J. W. (2007), "Characterising the components of urban link travel time variability using the k-NN method", TRB 2007 Annual Meeting CD-ROM, Washington DC, USA.

Rosen, D. A., Mammano, F. J. and Favout, R. (1970), "An electronic route-guidance system for highway vehicles", IEEE Transactions on Vehicular Technology, vol. 19, no. 1, pp. 143-152.

Rouphail, N. M., Ranjithan, S. R., El Desouki, W., Smith, T. and Downey Brill, E. (1995), "A decision support system for dynamic pre-trip route planning", Proceedings of the 4th ASCE

(American Society of Civil Engineers) International Conference on Application of Advanced Technologies in Transportation (AATT), New York, pp. 325-329.

Russell, S. and Norvig, P. (2003), "Artificial intelligence: a modern approach", 2nd edn, Prentice Hall.

Salim, V. (1997), "A comparison of shortest path algorithms in urban traffic assignment", Mathematics and Computers in Simulation, vol. 43, pp. 545-551.

Saricks, C. L., Schofer, J. L., Sööt, S. and Belella, P. A. (1997), "Evaluating effectiveness of real-time advanced traveler information systems using a small test vehicle fleet", Transportation Research Record, vol. 1588, pp. 41-48.

Schofer, J. L., Koppelman, F. S. and Charlton, W. A. (1997), "Perspectives on driver preferences for dynamic route guidance systems", Transportation Research Record, vol. 1588, pp. 26-31.

Sen, A., Sööt, S., Thakuriah, P. and Condie, H. (1998), "Estimation of static travel times in a dynamic route guidance system - II", Mathematical Computer Modelling, vol. 27, no. 9-11, pp. 67-85.

Sen, A. and Thakuriah, P. (1995), "Estimation of static travel times in a dynamic route guidance system", Mathematical Computer Modelling, vol. 22, no. 4-7, pp. 83-101.

Sen, A., Thakuriah, P., Zhu, X.-Q. and Karr, A. (1999), "Variances of link travel time estimates: implications for optimal routes", International Transactions in Operational Research, vol. 6, pp. 75-87.

Sen, S., Pillai, R., Joshi, S. and Rathi, A. K. (2001), "A mean-variance model for route guidance in advanced traveler information systems", Transportation Science, vol. 35, no. 1, pp. 37-49.

Sherali, H. D., Desai, J. and Rakha, H. (2006), "A discrete optimization approach for locating Automatic Vehicle Identification readers for the provision of roadway travel times", Transportation Research B, vol. 40, no. 10, pp. 857-871.

Simmons, N., Gates, G. and Burr, J. (2002), "Commercial applications arising from a floating vehicle data system in Europe", 9th World Congress on Intelligent Transportation Systems, Chicago.

Sivakumar, R. A. and Batta, R. (1994), "The variance-constrained shortest path problem", Transportation Science, vol. 28, no. 4, pp. 309-316.

Skriver, A. J. V. and Andersen, K. A. (2000), "A label correcting approach for solving bicriterion shortest-path problems", Computers and Operations Research, vol. 27, pp. 507-524.

Smeed, R. J. and Jeffcoate, G. O. (1971), "The variability of car journey times on a particular route", Traffic Engineering and Control, vol. 13, pp. 238-243.

Stentz, A. (1994), "Optimal and efficient planning for partially known environments", Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, California, pp. 3310-3317.

Stentz, A. (1995), "The focussed D* algorithm for real-time replanning", Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 1652-1659.

Storey, B. and Holtom, R. (2003), "The use of historic GPS data in transport and traffic monitoring", Traffic Engineering and Control, vol. 44, no. 10, pp. 376-379.

Sung, K., Bell, M. G. H., Seong, M. and Park, S. (2000), "Shortest paths in a network with time-dependent flow speeds", European Journal of Operational Research, vol. 121, pp. 32-39.

Sutch, T. (2001), "A note on journey time variability for M6 J4 to J10a in the morning peak from June to November 2000 ", TRL Seminar, 14 June 2001.

Svahn, F. (2004), "In-car navigation usage: an end-user survey on existing systems", Proceedings of IRIS 27.

Traffic Engineering and Control Product Review (2003), "FVD data ripe for transport plans", April 2003, pp. 4-6.

Tanenbaum, A. S. (1988), "Computer Networks", Prentice Hall.

Thomson, J. M. (1968), "The value of traffic management", Journal of Transport Economics and Policy, vol. 2, no. 1, pp. 3-32.

Torrieri, D. (1992), "Algorithms for finding an optimal set of short disjoint paths in a communication network", IEEE Transactions on Communications, vol. 40, no. 11, pp. 1698-1702.

Totani, S. (1980), "Development and current status of CACS (Comprehensive Automobile Traffic Control System)", 30th IEEE Vehicular Technology Conference, pp. 336-341.

Tsavachidis, M. (2000), "Aggregate analysis of driver response to collective route guidance and implications for system control", Road Transport Information and Control, vol. Conference Publication No. 472, pp. 17-21.

Turner, J. D. and Hoffman, F. S. (1990), "Data and route guidance", 3rd IEEE International Conference on Road Traffic Control, London, UK, pp. 16-22.

van Lint, J. W. C. and van Zuylen, H. J. (2005), "Monitoring and predicting freeway travel time reliability - Using width and skew of day-to-day travel time distribution", Transportation Research Record, vol. 1917, pp. 54-62.

von Tomkewitsch, R. (1991), "Dynamic route guidance and interactive transport management with ALI-SCOUT", IEEE Transactions on Vehicular Technology, vol. 40, no. 1, pp. 45-50.

Wahle, J., Annen, O., Schuster, C., Neubert, L. and Schreckenberg, M. (2001), "A dynamic route guidance system based on real traffic data", European Journal of Operational Research, vol. 131, pp. 302-308.

Wang, H. and Zhang, B. (1992), "Route planning and navigation system for an autonomous land vehicle", Vehicle Navigation and Information Systems Conference 1992, pp. 135-140.

Wardrop, J. G. (1952), "Some theoretical aspects of road traffic research", Proceedings of the Institution of Civil Engineers, vol. 1, no. 3, pp. 325-378.

Wattleworth, J. A. and Shuldiner, P. W. (1963), "Analytical methods in transportation; left turn penalties in traffic assignment models", Journal of Engineering Mechanics of the American Society of Civil Engineers, vol. 89, pp. 97-126.

Whitsitt, A. J. and Travis, L. E. (1996), "Traffic route generation and adaptation using case-based reasoning", Journal of Intelligent Transportation Systems, vol. 3, no. 3, pp. 181-204.

Wichmann, D. R. (2004), "Automated route finding on digital terrains", University of Auckland.

Wong, H.-K. and Sussman, J. M. (1973), "Dynamic travel time estimation on highway networks", Transportation Research, vol. 7, no. 4, pp. 355-370.

Wood, K., Maxwell, A., Stevens, A. and Thompson, S. (2006), "Routing assessment of dynamic route guidance systems", TRL Report No PPR093.

Yang, B. and Miller-Hooks, E. (2004), "Adaptive routing considering delays due to signal operations", Transportation Research B, vol. 38, pp. 385-413.

Yang, H.-H. and Chen, Y.-L. (2005), "Finding K shortest looping paths in a traffic-light network", Computers and Operations Research, vol. 32, pp. 571-581.

Yang, H. (1998), "Multiple equilibrium behaviors and advanced traveler information systems with endogenous market penetration", Transportation Research B, vol. 32, no. 3, pp. 205-218.

Yang, H. (1999), "Evaluating the benefits of a combined route guidance and road pricing system in a traffic network with recurrent congestion", Transportation, vol. 26, pp. 299-322.

Yang, H., Kitamura, R., Jovanis, P. P., Vaughn, K. M. and Abdel-Aty, A. (1993), "Exploration of route choice behavior with advanced traveler information using neural network concepts", Transportation, vol. 20, pp. 199-223.

Yen, J. Y. (1971), "Finding the k shortest loopless paths in a network", Management Science, vol. 17, no. 11, pp. 712-716.

Yu, G. and Yang, J. (1998), "On the Robust Shortest Path Problem", Computers and Operations Research, vol. 25, no. 6, pp. 457-468.

Zhan, F. B. and Noon, C. E. (1998), "Shortest path algorithms: an evaluation using real road networks", Transportation Science, vol. 32 (1), pp. 65-73.

Zhao, Y. (1997), "Vehicle location and navigation systems", Artec House Inc., Norwood, MA, USA.

Zielinski, P. (2004), "The computational complexity of the relative robust shortest path problem with interval data", European Journal of Operational Research, vol. 158, pp. 570-576.

Ziliaskopoulos, A. K., Kotzinos, D. and Mahmassani, H. S. (1997), "Design and implementation of parallel time-dependent least time path algorithms for intelligent transportation systems applications", Transportation Research C, vol. 5, no. 2, pp. 95-107.

Ziliaskopoulos, A. K. and Mahmassani, H. S. (1993), "Time-dependent shortest path algorithm for real-time intelligent vehicle highway system applications", Transportation Research Record, vol. 1408, pp. 94-100.

Ziliaskopoulos, A. K. and Mahmassani, H. S. (1996), "A note on least time path computation considering delays and prohibitions for intersection movements", Transportation Research B, vol. 30, no. 5, pp. 359-367.

Appendix A: Distribution of speeds

An empirical validation of the fact that speeds are normally distributed is carried out here. For this, the speed measurements collected by the ITIS FVD™ system during the period between 1 September 2006 and 1 December 2006 in Central London are used. The details of the data collection and processing are set out in Section 6.2.

As the data were aggregated in weekdays and weekends, as well as in 15-minute intervals, the links and time intervals chosen here are the ones for which the a large number of speed measurements (>200) are available. Namely, five link-interval cases are selected and shown in Table A-1, along with the corresponding availability of data. The descriptive statistics for the five link-interval cases are shown in Table A-2.

Table A-1: Selected links and time intervals

NAVTEQ™ link ID	Street name	Time interval	No of measurements (N)
25968315 F	Victoria Embankment	Weekday 07.45-08.00	267
25968315 F	Victoria Embankment	Weekday 08.15-08.30	287
25325371 F	Victoria Embankment	Weekday 07.45-08.00	244
25307675 T	Park Lane	Weekday 08.30-08.45	285
25307576 T	Park Lane	Weekday 08.45-09.00	258

Table A-2: Speed distributions descriptive statistics (in km/h)

NAVTEQ™ link ID	Mean speed (time)	Std. deviation	Max. speed	Min. speed
25968315 F	29.52	13.62	79.2	10.56
25968315 F	25.41	11	74.4	7.32
25325371 F	20.95	9.42	52.69	7.07
25307675 T	24.71	10.46	51.98	8.02
25307576 T	26.53	12.09	78	7.67

Using Microsoft Excel, the speed distributions for the five link-interval cases are tabulated. For each distribution, the following procedure is followed: taking each value v_i of the speed time distribution, its rank $\mathrm{RANK}(v_i)$ is calculated, and from this, the corresponding cumulative probability $y(v_i) = 1 - (\mathrm{RANK}(v_i) - 1) / (N - 1)$ is derived. By plotting $y(v_i)$ against v_i for each i, where $1 < i < N$, the cumulative distribution plot $y(v)$ of v is drawn.

The mean and standard deviation values of the speed distribution are $\bar{\upsilon}$ and $\sigma(\upsilon)$. Taking each speed value υ_i again and considering the cumulative probability distribution function of a normal distribution,

$$y_N(\upsilon_i) = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\upsilon_i - \bar{\upsilon}}{\sigma(\upsilon)\sqrt{2}}\right),$$

the cumulative distribution plot $y_N(\upsilon)$ of υ based on the normal distribution can be drawn.

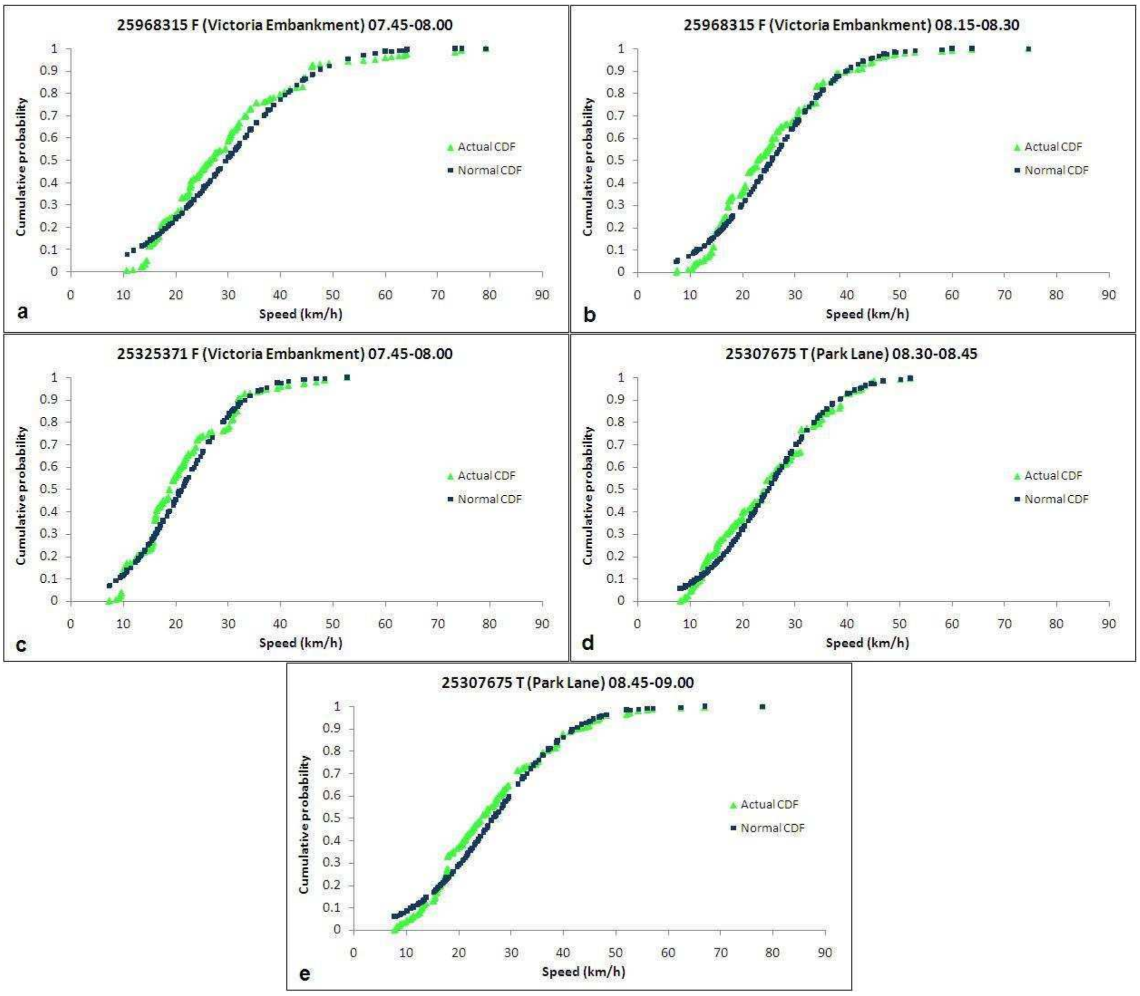


Figure A-1: Comparison of $y(\upsilon)$ and $y_N(\upsilon)$

By drawing $y(\upsilon)$ and $y_N(\upsilon)$ against υ for each link-interval case in Figure A-1 it can be seen that the curves nearly coincide with each other, which suggests that the assumption of speed being normally distributed seems plausible. To further test this, Kolmogorov-Smirnov (K-S) tests are carried out on the five distributions.

For a speed distribution the K-S statistic D is

$$D = \sup_{\upsilon}\left|y(\upsilon) - y_N(\upsilon)\right|.$$

The null hypothesis in the K-S test is that speed is normally distributed, i.e. H_0: $D = 0$. The null hypothesis is rejected at significance level α if $D > D^{\alpha}_{crit}$, where D^{α}_{crit} is the critical value of the K-S statistic for level α and is equal to

$$D^{\alpha}_{crit} = \frac{K_{\alpha}}{\sqrt{N}}$$

K_{α} is obtained from tables of the Kolmogorov distribution, and for samples with $N > 35$, it is $K_{0.05}$ = 1.36 and $K_{0.01}$ = 1.63 for significance levels $\alpha = 0.05$ and $\alpha = 0.01$ respectively.

The results of the K-S tests for the speed distributions of the five link-interval cases examined are given in Table A-3. Out of the five distributions and taking a significance level of 0.01, the null hypothesis is only rejected once, meaning that for four out of the five link-interval cases, the distribution of speeds is similar to the normal distribution. In one of them the null hypothesis is even accepted at level 0.05. This suggests that the assumption that speeds are normally distributed is a fairly accurate one.

Table A-3: K-S test results

NAVTEQ™ link ID	D	$D^{0.05}_{crit}$	$D^{0.01}_{crit}$	Normal at 0.05	Normal at 0.01
25968315 F	0.096562	0.083231	0.099754	NO	YES
25968315 F	0.09582	0.08028	0.09622	NO	YES
25325371 F	0.10778	0.08707	0.10435	NO	NO
25307675 T	0.0793	0.08056	0.09655	YES	YES
25307576 T	0.09606	0.08467	0.10148	NO	YES

Appendix B: Example of travel time distribution

For Example 2 in Section 3.3.7, the log-normal distribution of travel times $t(l)$ on link l, which is $\lambda(l)$ = 100 m long, is extracted from a normal distribution of 200 speed measurements, $\upsilon(l)$. The complete range of values is shown in Table B-1, while the descriptive statistics of the speed and travel time distributions are given in Table B-2.

Table B-1: The complete speed distribution and the resulting travel time distribution

N	$\upsilon(l)$ (m/s)	$t(l)$ (s)	N	$\upsilon(l)$ (m/s)	$t(l)$ (s)	N	$\upsilon(l)$ (m/s)	$t(l)$ (s)	N	$\upsilon(l)$ (m/s)	$t(l)$ (s)
1	14.10	7.09	51	7.27	13.76	101	13.43	7.45	151	16.21	6.17
2	11.17	8.95	52	19.34	5.17	102	17.55	5.70	152	13.06	7.66
3	15.73	6.36	53	11.16	8.96	103	16.54	6.05	153	17.17	5.82
4	18.83	5.31	54	13.04	7.67	104	13.18	7.59	154	16.58	6.03
5	18.60	5.38	55	17.27	5.79	105	18.91	5.29	155	18.23	5.49
6	20.20	4.95	56	16.40	6.10	106	9.72	10.29	156	6.69	14.94
7	8.45	11.84	57	17.62	5.67	107	16.65	6.01	157	16.39	6.10
8	14.30	6.99	58	16.79	5.96	108	14.65	6.83	158	19.40	5.15
9	18.29	5.47	59	10.88	9.19	109	15.13	6.61	159	9.83	10.17
10	11.74	8.52	60	11.65	8.58	110	13.04	7.67	160	15.14	6.61
11	12.93	7.73	61	17.08	5.85	111	13.35	7.49	161	19.06	5.25
12	9.93	10.07	62	15.97	6.26	112	17.55	5.70	162	20.06	4.98
13	9.46	10.57	63	12.18	8.21	113	17.41	5.74	163	15.72	6.36
14	12.07	8.29	64	14.28	7.00	114	16.37	6.11	164	15.44	6.48
15	12.68	7.89	65	15.39	6.50	115	17.07	5.86	165	20.56	4.86
16	8.65	11.57	66	16.67	6.00	116	19.89	5.03	166	14.87	6.72
17	13.30	7.52	67	15.42	6.49	117	15.91	6.28	167	13.03	7.68
18	13.79	7.25	68	12.27	8.15	118	16.77	5.96	168	17.72	5.64
19	15.40	6.49	69	20.65	4.84	119	20.56	4.86	169	14.97	6.68
20	13.90	7.19	70	16.46	6.07	120	13.99	7.15	170	18.12	5.52
21	14.02	7.13	71	15.22	6.57	121	18.11	5.52	171	16.30	6.13
22	13.89	7.20	72	17.49	5.72	122	15.43	6.48	172	19.36	5.17
23	19.03	5.26	73	17.59	5.69	123	18.42	5.43	173	14.64	6.83
24	14.74	6.78	74	13.09	7.64	124	14.56	6.87	174	12.11	8.25
25	14.44	6.92	75	12.23	8.18	125	12.66	7.90	175	10.38	9.63
26	13.46	7.43	76	18.33	5.45	126	18.23	5.49	176	7.30	13.69
27	20.92	4.78	77	11.40	8.77	127	13.25	7.54	177	15.31	6.53
28	17.60	5.68	78	10.32	9.69	128	16.60	6.02	178	15.90	6.29
29	22.13	4.52	79	17.13	5.84	129	16.64	6.01	179	14.95	6.69
30	13.04	7.67	80	16.92	5.91	130	14.05	7.12	180	15.60	6.41
31	19.98	5.00	81	21.62	4.63	131	13.68	7.31	181	15.77	6.34
32	10.16	9.84	82	19.33	5.17	132	10.90	9.17	182	19.43	5.15
33	16.62	6.02	83	18.91	5.29	133	20.98	4.77	183	15.26	6.55
34	17.71	5.65	84	15.34	6.52	134	13.30	7.52	184	11.65	8.58
35	20.76	4.82	85	15.01	6.66	135	15.26	6.55	185	10.83	9.23
36	14.75	6.78	86	16.36	6.11	136	14.30	6.99	186	15.57	6.42
37	13.43	7.45	87	14.92	6.70	137	23.51	4.25	187	17.85	5.60
38	17.03	5.87	88	11.84	8.45	138	18.76	5.33	188	10.47	9.55
39	13.86	7.22	89	9.68	10.34	139	17.64	5.67	189	13.17	7.59
40	17.27	5.79	90	17.48	5.72	140	19.00	5.26	190	16.57	6.03
41	10.67	9.37	91	16.33	6.12	141	15.56	6.43	191	14.35	6.97
42	12.46	8.03	92	16.85	5.93	142	16.63	6.01	192	13.97	7.16
43	10.44	9.58	93	15.64	6.39	143	14.25	7.02	193	14.51	6.89
44	13.91	7.19	94	11.92	8.39	144	11.34	8.82	194	13.68	7.31
45	14.90	6.71	95	18.71	5.34	145	18.80	5.32	195	12.59	7.94
46	15.08	6.63	96	14.07	7.11	146	14.13	7.08	196	16.93	5.91
47	14.03	7.13	97	12.48	8.01	147	11.08	9.02	197	15.18	6.59
48	21.58	4.63	98	12.54	7.98	148	17.29	5.78	198	18.79	5.32
49	9.77	10.23	99	13.71	7.29	149	17.35	5.76	199	15.79	6.33
50	12.79	7.82	100	13.64	7.33	150	16.28	6.14	200	15.17	6.59

Table B-2: Descriptive statistics of the speed and travel time distributions

Speed (m/s)		*Travel time (s)*	
Time-mean	15.21	Mean	6.90
Space-mean	14.49	Standard Error	0.12
Standard Error	0.22	Median	6.55
Median	15.26	Mode	7.45
Mode	13.43	Standard Deviation	1.68
Standard Deviation	3.11	Sample Variance	2.83
Sample Variance	9.65	Kurtosis	4.36
Kurtosis	-0.11	Skewness	1.68
Skewness	-0.13	Range	10.69
Range	16.81	Minimum	4.25
Minimum	6.69	Maximum	14.94
Maximum	23.51	Sum	1380.02
Sum	3041.16	Count	200.00
Count	200.00		

Appendix C: Complete results of field experiments

C.1 Single-vehicle experiment

Table C.1-1: Recorded departure and arrival times in the single-vehicle experiment

Run	Date	Route		DT	Conv.	ARIAdNE			ATA
		Origin	Destination		CETA	AERTA	AETA	ALRTA	
1	16/07/2007	Bayswater Road	Arch Street	15:48	16:01	16:05	16:18	16:36	16:22
2	16/07/2007	Meadow Road	Whitechapel High Street	16:32	16:40	16:39	16:45	16:53	16:45
3	16/07/2007	Commercial Road	Buckingham Street	16:57	17:06	17:07	17:13	17:23	17:14
4	16/07/2007	John Adam Street	Park Crescent	17:24	17:29	17:31	17:38	17:48	17:42
5	16/07/2007	Park Crescent	Bayswater Road	17:47	17:51	17:54	17:57	18:02	17:57
6	17/07/2007	Bayswater Road	Potters Fields	13:43	13:57	14:03	14:16	14:35	14:33
7	17/07/2007	Potters Fields	Dugard Way	14:49	14:57	14:55	14:59	15:05	15:05
8	17/07/2007	Renfrew Road	Loxham Street	15:12	15:23	15:20	15:28	15:41	15:28
9	17/07/2007	Argyle Street	Laud Street	15:44	15:55	16:00	16:13	16:30	16:14
10	17/07/2007	Laud Street	Park Crescent	16:18	16:28	16:29	16:39	16:52	16:43
11	17/07/2007	Park Crescent	Knightsbridge	16:45	16:56	16:56	17:02	17:10	17:03
12	18/07/2007	Bayswater Road	Potters Fields	9:36	9:51	9:55	10:09	10:29	10:09
13	18/07/2007	Thrale Street	Park Crescent	10:53	11:02	11:02	11:11	11:25	11:25
14	18/07/2007	Portland Place	Tyers Street	11:31	11:42	11:45	11:55	12:10	11:58
15	19/07/2007	Bayswater Road	Whitechapel High Street	8:57	9:11	9:15	9:31	9:54	9:38
16	19/07/2007	Commercial Road	Gower Mews	10:03	10:13	10:13	10:23	10:37	10:37
17	19/07/2007	Bedford Square	Tyers Street	10:51	11:01	11:02	11:11	11:23	11:08
18	19/07/2007	Tyers Street	Park Crescent	11:26	11:36	11:41	11:52	12:07	11:47
19	19/07/2007	Portland Place	Culross Street	12:09	12:13	12:15	12:20	12:27	12:21
20	19/07/2007	Culross Street	Potters Fields	12:24	12:39	12:38	12:52	13:12	12:52
21	19/07/2007	Potters Fields	Knightsbridge	14:12	14:25	14:27	14:38	14:54	14:46
22	20/07/2007	Bayswater Road	Whitechapel High Street	14:21	14:37	14:42	14:58	15:19	14:58
23	20/07/2007	Commercial Road	Jubilee Gardens	15:16	15:26	15:26	15:34	15:45	15:32
24	20/07/2007	Jubilee Gardens	Circus Mews	15:44	15:56	16:02	16:13	16:28	16:06
25	20/07/2007	Enford Street	Old Kent Road	16:28	16:46	16:46	17:01	17:23	16:58
26	20/07/2007	Old Kent Road	Knightsbridge	17:04	17:19	17:14	17:22	17:34	17:23
27	24/07/2007	Bayswater Road	Whitechapel High Street	15:01	15:15	15:24	15:39	15:59	15:45
28	24/07/2007	Commercial Road	Tyers Street	16:02	16:14	16:15	16:25	16:40	16:26
29	24/07/2007	Tyers Street	Chequer Street	16:30	16:44	16:48	17:00	17:15	16:52
30	24/07/2007	Chequer Street	Old Kent Road	17:03	17:14	17:15	17:24	17:37	17:24
31	24/07/2007	Old Kent Road	Bayswater Road	17:37	17:53	17:48	17:59	18:13	18:03
32	25/07/2007	Bayswater Road	Warren Mews	14:16	14:22	14:24	14:30	14:39	14:28
33	25/07/2007	Warren Street	Lily Place	14:43	14:50	14:53	15:00	15:10	15:00
34	25/07/2007	St Cross Street	Jubilee Gardens	15:14	15:20	15:20	15:25	15:32	15:25
35	25/07/2007	Belvedere Road	Rockingham Street	15:37	15:41	15:44	15:50	15:57	15:47
36	25/07/2007	Rockingham Street	Knightsbridge	15:56	16:09	16:07	16:17	16:30	16:13
37	26/07/2007	Bayswater Road	Potters Fields	13:12	13:28	13:33	13:47	14:07	13:47
38	26/07/2007	Potters Fields	Mecklenburgh Square	14:03	14:16	14:17	14:27	14:41	14:22
39	26/07/2007	Mecklenburgh Square	Knightsbridge	14:31	14:43	14:45	14:56	15:11	14:57
40	26/07/2007	Bayswater Road	Old Kent Road	15:26	15:42	15:41	15:54	16:12	15:49
41	26/07/2007	Old Kent Road	Three Cups Yard	16:05	16:16	16:14	16:23	16:34	16:31
42	26/07/2007	Sandland Street	Bayswater Road	16:39	16:48	16:53	17:02	17:15	17:07
43	27/07/2007	Bayswater Road	Whitechapel High Street	9:36	9:51	9:51	10:04	10:24	10:04
44	27/07/2007	Commercial Road	Tyers Street	10:15	10:29	10:28	10:38	10:52	10:39
45	27/07/2007	Tyers Street	Cursitor Street	10:51	11:00	11:01	11:08	11:17	11:06
46	27/07/2007	Furnival Street	Cato Street	11:12	11:20	11:26	11:35	11:48	11:35
47	27/07/2007	Brendon Street	Carlton Gardens	11:55	12:02	12:06	12:13	12:23	12:17
48	27/07/2007	Carlton Gardens	Potters Fields	12:32	12:46	12:47	12:59	13:14	13:00
49	27/07/2007	Potters Fields	Knightsbridge	14:16	14:32	14:29	14:40	14:55	14:42
50	30/07/2007	Bayswater Road	Potters Fields	14:08	14:22	14:22	14:35	14:53	14:34
51	30/07/2007	Potters Fields	Bayswater Road	14:39	14:54	14:57	15:11	15:30	15:05
52	30/07/2007	Bayswater Road	Galway Street	15:30	15:41	15:47	15:57	16:17	15:57
53	30/07/2007	Lever Street	Tyers Street	16:17	16:31	16:30	16:41	16:55	16:44
54	30/07/2007	Tyers Street	Bedford Square	17:02	17:13	17:12	17:20	17:31	17:18
55	30/07/2007	Bedford Square	Bayswater Road	17:38	17:44	17:44	17:51	18:00	17:56

Table C.1-1 (continued)

Run	Date	Route		DT	Conv.	ARIAdNE			ATA
		Origin	Destination		CETA	AERTA	AETA	ALRTA	
56	31/07/2007	Bayswater Road	Whitechapel High Street	13:33	13:47	13:52	14:07	14:26	14:17
57	31/07/2007	Commercial Road	Jubilee Gardens	14:33	14:41	14:40	14:47	14:56	14:49
58	31/07/2007	Jubilee Gardens	Park Crescent	15:18	15:26	15:27	15:35	15:46	15:38
59	31/07/2007	Park Crescent	Old Kent Road	15:48	16:04	15:59	16:10	16:26	16:18
60	31/07/2007	Old Kent Road	Knightsbridge	16:34	16:48	16:45	16:53	17:05	16:56
61	01/08/2007	Bayswater Road	Potters Fields	8:50	9:05	9:07	9:22	9:43	9:29
62	01/08/2007	Potters Fields	Bedford Square	9:43	10:00	9:55	10:06	10:29	10:16
63	01/08/2007	Bedford Square	Tyers Street	10:30	10:39	10:39	10:49	11:03	10:49
64	01/08/2007	Tyers Street	Lever Street	11:33	11:46	11:50	12:02	12:11	11:58
65	01/08/2007	Lever Street	Carlton Gardens	12:23	12:35	12:32	12:41	12:54	12:45
66	01/08/2007	Carlton Gardens	Bayswater Road	13:10	13:18	13:18	13:23	13:23	13:22
67	02/08/2007	Bayswater Road	Old Kent Road	10:46	11:01	10:59	11:09	11:24	11:14
68	02/08/2007	Old Kent Road	Bedford Square	11:23	11:37	11:38	11:45	11:58	11:58
69	02/08/2007	Bedford Square	Tyers Street	12:19	12:28	12:30	12:40	12:52	12:40
70	02/08/2007	Tyers Street	Whitechapel High Street	13:02	13:12	13:13	13:21	13:33	13:20
71	02/08/2007	Commercial Road	Jubilee Gardens	13:32	13:42	13:41	13:48	13:58	13:49
72	02/08/2007	Jubilee Gardens	Bayswater Road	14:04	14:16	14:13	14:20	14:31	14:29

Table C.1-2: Recorded travel times (in minutes) and travel time ratios

Run	Conv.	ARIAdNE			ATT	Conv.		ARIAdNE			ATT
	CETT	AERTT	AETT	ALRTT		CETT/ATT	UCETT/ATT	AERTT/ATT	AETT/ATT	ALRTT/ATT	
1	13	17	30	48	34	0.3824	0.7647	0.5000	0.8824	1.4118	1
2	8	7	13	21	13	0.6154	1.2308	0.5385	1.0000	1.6154	1
3	9	10	16	26	17	0.5294	1.0588	0.5882	0.9412	1.5294	1
4	5	7	14	24	18	0.2778	0.5556	0.3889	0.7778	1.3333	1
5	4	7	10	15	10	0.4000	0.8000	0.7000	1.0000	1.5000	1
6	14	20	33	52	50	0.2800	0.5600	0.4000	0.6600	1.0400	1
7	8	6	10	16	16	0.5000	1.0000	0.3750	0.6250	1.0000	1
8	11	8	16	29	16	0.6875	1.3750	0.5000	1.0000	1.8125	1
9	11	16	29	46	30	0.3667	0.7333	0.5333	0.9667	1.5333	1
10	10	11	21	34	25	0.4000	0.8000	0.4400	0.8400	1.3600	1
11	11	11	17	25	18	0.6111	1.2222	0.6111	0.9444	1.3889	1
12	15	19	33	53	33	0.4545	0.9091	0.5758	1.0000	1.6061	1
13	9	9	18	32	32	0.2813	0.5625	0.2813	0.5625	1.0000	1
14	11	14	24	39	27	0.4074	0.8148	0.5185	0.8889	1.4444	1
15	14	18	34	57	41	0.3415	0.6829	0.4390	0.8293	1.3902	1
16	10	10	20	34	34	0.2941	0.5882	0.2941	0.5882	1.0000	1
17	10	11	20	32	17	0.5882	1.1765	0.6471	1.1765	1.8824	1
18	10	15	26	41	21	0.4762	0.9524	0.7143	1.2381	1.9524	1
19	4	6	11	18	12	0.3333	0.6667	0.5000	0.9167	1.5000	1
20	15	14	28	48	28	0.5357	1.0714	0.5000	1.0000	1.7143	1
21	13	15	26	42	34	0.3824	0.7647	0.4412	0.7647	1.2353	1
22	16	21	37	58	37	0.4324	0.8649	0.5676	1.0000	1.5676	1
23	10	10	18	29	16	0.6250	1.2500	0.6250	1.1250	1.8125	1
24	12	18	29	44	22	0.5455	1.0909	0.8182	1.3182	2.0000	1
25	18	18	33	55	30	0.6000	1.2000	0.6000	1.1000	1.8333	1
26	15	10	18	30	19	0.7895	1.5789	0.5263	0.9474	1.5789	1
27	14	23	38	58	44	0.3182	0.6364	0.5227	0.8636	1.3182	1
28	12	13	23	38	24	0.5000	1.0000	0.5417	0.9583	1.5833	1
29	14	18	30	45	22	0.6364	1.2727	0.8182	1.3636	2.0455	1
30	11	12	21	34	21	0.5238	1.0476	0.5714	1.0000	1.6190	1
31	16	11	22	36	26	0.6154	1.2308	0.4231	0.8462	1.3846	1
32	6	8	14	23	12	0.5000	1.0000	0.6667	1.1667	1.9167	1
33	7	10	17	27	17	0.4118	0.8235	0.5882	1.0000	1.5882	1
34	6	6	11	18	11	0.5455	1.0909	0.5455	1.0000	1.6364	1
35	4	7	13	20	10	0.4000	0.8000	0.7000	1.3000	2.0000	1
36	13	11	21	34	17	0.7647	1.5294	0.6471	1.2353	2.0000	1
37	16	21	35	55	35	0.4571	0.9143	0.6000	1.0000	1.5714	1
38	13	14	24	38	19	0.6842	1.3684	0.7368	1.2632	2.0000	1
39	12	14	25	40	26	0.4615	0.9231	0.5385	0.9615	1.5385	1
40	16	15	28	46	23	0.6957	1.3913	0.6522	1.2174	2.0000	1

Table C.1-2 (continued)

Run	Conv.	ARIAdNE			ATT	Conv.		ARIAdNE			ATT
	CETT	AERTT	AETT	ALRTT		CETT/ATT	UCETT/ATT	AERTT/ATT	AETT/ATT	ALRTT/ATT	
41	11	9	18	29	26	0.4231	0.8462	0.3462	0.6923	1.1154	1
42	9	14	23	36	28	0.3214	0.6429	0.5000	0.8214	1.2857	1
43	15	15	28	48	28	0.5357	1.0714	0.5357	1.0000	1.7143	1
44	14	13	23	37	24	0.5833	1.1667	0.5417	0.9583	1.5417	1
45	9	10	17	26	15	0.6000	1.2000	0.6667	1.1333	1.7333	1
46	8	14	23	36	23	0.3478	0.6957	0.6087	1.0000	1.5652	1
47	7	11	18	28	22	0.3182	0.6364	0.5000	0.8182	1.2727	1
48	14	15	27	42	28	0.5000	1.0000	0.5357	0.9643	1.5000	1
49	16	13	24	39	26	0.6154	1.2308	0.5000	0.9231	1.5000	1
50	14	14	27	45	26	0.5385	1.0769	0.5385	1.0385	1.7308	1
51	15	18	32	51	26	0.5769	1.1538	0.6923	1.2308	1.9615	1
52	11	17	27	47	27	0.4074	0.8148	0.6296	1.0000	1.7407	1
53	14	13	24	38	27	0.5185	1.0370	0.4815	0.8889	1.4074	1
54	11	10	18	29	16	0.6875	1.3750	0.6250	1.1250	1.8125	1
55	6	6	13	22	18	0.3333	0.6667	0.3333	0.7222	1.2222	1
56	14	19	34	53	44	0.3182	0.6364	0.4318	0.7727	1.2045	1
57	8	7	14	23	16	0.5000	1.0000	0.4375	0.8750	1.4375	1
58	8	9	17	28	20	0.4000	0.8000	0.4500	0.8500	1.4000	1
59	16	11	22	38	30	0.5333	1.0667	0.3667	0.7333	1.2667	1
60	14	11	19	31	22	0.6364	1.2727	0.5000	0.8636	1.4091	1
61	15	17	32	53	39	0.3846	0.7692	0.4359	0.8205	1.3590	1
62	17	12	23	46	33	0.5152	1.0303	0.3636	0.6970	1.3939	1
63	9	9	19	33	19	0.4737	0.9474	0.4737	1.0000	1.7368	1
64	13	17	29	38	25	0.5200	1.0400	0.6800	1.1600	1.5200	1
65	12	9	18	31	22	0.5455	1.0909	0.4091	0.8182	1.4091	1
66	8	8	13	13	12	0.6667	1.3333	0.6667	1.0833	1.0833	1
67	15	13	23	38	28	0.5357	1.0714	0.4643	0.8214	1.3571	1
68	14	15	22	35	35	0.4000	0.8000	0.4286	0.6286	1.0000	1
69	9	11	21	33	21	0.4286	0.8571	0.5238	1.0000	1.5714	1
70	10	11	19	31	18	0.5556	1.1111	0.6111	1.0556	1.7222	1
71	10	9	16	26	17	0.5882	1.1765	0.5294	0.9412	1.5294	1
72	12	9	16	27	25	0.4800	0.9600	0.3600	0.6400	1.0800	1

Table C.1-3: Travel time ratios to ATT and corresponding cumulative probabilities

Conv.				ARIAdNE						ATT	ATT CP
CETT	CETT CP	UCETT	UCETT CP	AERTT	AERTT CP	AETT	AETT CP	ALRTT	ALRTT CP		
0.27778	0.01389	0.55556	0.01389	0.28125	0.01389	0.5625	0.01389	1	0.05556	1	0
0.28	0.02778	0.56	0.02778	0.29412	0.02778	0.58824	0.02778	1	0.05556	1	0.01408
0.28125	0.04167	0.5625	0.04167	0.33333	0.04167	0.625	0.04167	1	0.05556	1	0.02817
0.29412	0.05556	0.58824	0.05556	0.34615	0.05556	0.62857	0.05556	1	0.05556	1	0.04225
0.31818	0.09722	0.63636	0.09722	0.36	0.06944	0.64	0.06944	1.04	0.06944	1	0.05634
0.31818	0.09722	0.63636	0.09722	0.36364	0.08333	0.66	0.08333	1.08	0.08333	1	0.07042
0.31818	0.09722	0.63636	0.09722	0.36667	0.09722	0.69231	0.09722	1.08333	0.09722	1	0.08451
0.32143	0.11111	0.64286	0.11111	0.375	0.11111	0.69697	0.11111	1.11538	0.11111	1	0.09859
0.33333	0.13889	0.66667	0.13889	0.38889	0.125	0.72222	0.125	1.20455	0.125	1	0.11268
0.33333	0.13889	0.66667	0.13889	0.4	0.13889	0.73333	0.13889	1.22222	0.13889	1	0.12676
0.34146	0.15278	0.68293	0.15278	0.40909	0.15278	0.76471	0.15278	1.23529	0.15278	1	0.14085
0.34783	0.16667	0.69565	0.16667	0.42308	0.16667	0.77273	0.16667	1.26667	0.16667	1	0.15493
0.36667	0.18056	0.73333	0.18056	0.42857	0.18056	0.77778	0.18056	1.27273	0.18056	1	0.16901
0.38235	0.20833	0.76471	0.20833	0.43182	0.19444	0.81818	0.20833	1.28571	0.19444	1	0.1831
0.38235	0.20833	0.76471	0.20833	0.4359	0.20833	0.81818	0.20833	1.31818	0.20833	1	0.19718
0.38462	0.22222	0.76923	0.22222	0.4375	0.22222	0.82051	0.22222	1.33333	0.22222	1	0.21127
0.4	0.29167	0.8	0.29167	0.43902	0.23611	0.82143	0.25	1.35714	0.23611	1	0.22535
0.4	0.29167	0.8	0.29167	0.44	0.25	0.82143	0.25	1.35897	0.25	1	0.23944
0.4	0.29167	0.8	0.29167	0.44118	0.26389	0.82927	0.26389	1.36	0.26389	1	0.25352
0.4	0.29167	0.8	0.29167	0.45	0.27778	0.84	0.27778	1.38462	0.27778	1	0.26761

Table C.1-3 (continued)

Conv.				ARIAdNE						ATT	ATT CP
CETT	CETT CP	UCETT	UCETT CP	AERTT	AERTT CP	AETT	AETT CP	ALRTT	ALRTT CP		
0.4	0.29167	0.8	0.29167	0.46429	0.29167	0.84615	0.29167	1.38889	0.29167	1	0.28169
0.40741	0.31944	0.81481	0.31944	0.47368	0.30556	0.85	0.30556	1.39024	0.30556	1	0.29577
0.40741	0.31944	0.81481	0.31944	0.48148	0.31944	0.86364	0.33333	1.39394	0.31944	1	0.30986
0.41176	0.33333	0.82353	0.33333	0.5	0.43056	0.86364	0.33333	1.4	0.33333	1	0.32394
0.42308	0.34722	0.84615	0.34722	0.5	0.43056	0.875	0.34722	1.40741	0.34722	1	0.33803
0.42857	0.36111	0.85714	0.36111	0.5	0.43056	0.88235	0.36111	1.40909	0.375	1	0.35211
0.43243	0.375	0.86486	0.375	0.5	0.43056	0.88889	0.38889	1.40909	0.375	1	0.3662
0.45455	0.38889	0.90909	0.38889	0.5	0.43056	0.88889	0.38889	1.41176	0.38889	1	0.38028
0.45714	0.40278	0.91429	0.40278	0.5	0.43056	0.91667	0.40278	1.4375	0.40278	1	0.39437
0.46154	0.41667	0.92308	0.41667	0.5	0.43056	0.92308	0.41667	1.44444	0.41667	1	0.40845
0.47368	0.43056	0.94737	0.43056	0.5	0.43056	0.94118	0.44444	1.5	0.47222	1	0.42254
0.47619	0.44444	0.95238	0.44444	0.51852	0.44444	0.94118	0.44444	1.5	0.47222	1	0.43662
0.48	0.45833	0.96	0.45833	0.52273	0.45833	0.94444	0.45833	1.5	0.47222	1	0.4507
0.5	0.52778	1	0.52778	0.52381	0.47222	0.94737	0.47222	1.5	0.47222	1	0.46479
0.5	0.52778	1	0.52778	0.52632	0.48611	0.95833	0.5	1.52	0.48611	1	0.47887
0.5	0.52778	1	0.52778	0.52941	0.5	0.95833	0.5	1.52941	0.51389	1	0.49296
0.5	0.52778	1	0.52778	0.53333	0.51389	0.96154	0.51389	1.52941	0.51389	1	0.50704
0.5	0.52778	1	0.52778	0.53571	0.54167	0.96429	0.52778	1.53333	0.52778	1	0.52113
0.51515	0.54167	1.0303	0.54167	0.53571	0.54167	0.96667	0.54167	1.53846	0.54167	1	0.53521
0.51852	0.55556	1.03704	0.55556	0.53846	0.58333	1	0.75	1.54167	0.55556	1	0.5493
0.52	0.56944	1.04	0.56944	0.53846	0.58333	1	0.75	1.56522	0.56944	1	0.56338
0.52381	0.58333	1.04762	0.58333	0.53846	0.58333	1	0.75	1.56757	0.58333	1	0.57746
0.52941	0.59722	1.05882	0.59722	0.54167	0.61111	1	0.75	1.57143	0.61111	1	0.59155
0.53333	0.61111	1.06667	0.61111	0.54167	0.61111	1	0.75	1.57143	0.61111	1	0.60563
0.53571	0.65278	1.07143	0.65278	0.54545	0.625	1	0.75	1.57895	0.625	1	0.61972
0.53571	0.65278	1.07143	0.65278	0.56757	0.63889	1	0.75	1.58333	0.63889	1	0.6338
0.53571	0.65278	1.07143	0.65278	0.57143	0.65278	1	0.75	1.58824	0.65278	1	0.64789
0.53846	0.66667	1.07692	0.66667	0.57576	0.66667	1	0.75	1.60606	0.66667	1	0.66197
0.54545	0.70833	1.09091	0.70833	0.58824	0.69444	1	0.75	1.61538	0.68056	1	0.67606
0.54545	0.70833	1.09091	0.70833	0.58824	0.69444	1	0.75	1.61905	0.69444	1	0.69014
0.54545	0.70833	1.09091	0.70833	0.6	0.72222	1	0.75	1.63636	0.70833	1	0.70423
0.55556	0.72222	1.11111	0.72222	0.6	0.72222	1	0.75	1.71429	0.73611	1	0.71831
0.57692	0.73611	1.15385	0.73611	0.6087	0.73611	1	0.75	1.71429	0.73611	1	0.73239
0.58333	0.75	1.16667	0.75	0.61111	0.76389	1	0.75	1.72222	0.75	1	0.74648
0.58824	0.77778	1.17647	0.77778	0.61111	0.76389	1.03846	0.76389	1.73077	0.76389	1	0.76056
0.58824	0.77778	1.17647	0.77778	0.625	0.79167	1.05556	0.77778	1.73333	0.77778	1	0.77465
0.6	0.80556	1.2	0.80556	0.625	0.79167	1.08333	0.79167	1.73684	0.79167	1	0.78873
0.6	0.80556	1.2	0.80556	0.62963	0.80556	1.1	0.80556	1.74074	0.80556	1	0.80282
0.61111	0.81944	1.22222	0.81944	0.64706	0.83333	1.125	0.83333	1.8125	0.84722	1	0.8169
0.61538	0.86111	1.23077	0.86111	0.64706	0.83333	1.125	0.83333	1.8125	0.84722	1	0.83099
0.61538	0.86111	1.23077	0.86111	0.65217	0.84722	1.13333	0.84722	1.8125	0.84722	1	0.84507
0.61538	0.86111	1.23077	0.86111	0.66667	0.88889	1.16	0.86111	1.83333	0.86111	1	0.85915
0.625	0.875	1.25	0.875	0.66667	0.88889	1.16667	0.875	1.88235	0.875	1	0.87324
0.63636	0.90278	1.27273	0.90278	0.66667	0.88889	1.17647	0.88889	1.91667	0.88889	1	0.88732
0.63636	0.90278	1.27273	0.90278	0.68	0.90278	1.21739	0.90278	1.95238	0.90278	1	0.90141
0.66667	0.91667	1.33333	0.91667	0.69231	0.91667	1.23077	0.91667	1.96154	0.91667	1	0.91549
0.68421	0.93056	1.36842	0.93056	0.7	0.94444	1.23529	0.93056	2	0.98611	1	0.92958
0.6875	0.95833	1.375	0.95833	0.7	0.94444	1.2381	0.94444	2	0.98611	1	0.94366
0.6875	0.95833	1.375	0.95833	0.71429	0.95833	1.26316	0.95833	2	0.98611	1	0.95775
0.69565	0.97222	1.3913	0.97222	0.73684	0.97222	1.3	0.97222	2	0.98611	1	0.97183
0.76471	0.98611	1.52941	0.98611	0.81818	1	1.31818	0.98611	2	0.98611	1	0.98592
0.78947	1	1.57895	1	0.81818	1	1.36364	1	2.04545	1	1	1

C.2 Double-vehicle experiment

Table C.2-1: Recorded departure and arrival times in the double-vehicle experiment

Run	Date	Route		DT	Conv.		ARIAdNE			
		Origin	Destination		CETA	CATA	AERTA	AETA	ALRTA	AATA
1	30/07/2007	Bayswater Road	Potters Fields	14:08	14:22	14:31	14:22	14:35	14:53	14:34
2	30/07/2007	Potters Fields	Bayswater Road	14:39	14:54	15:16	14:57	15:11	15:30	15:05
3	30/07/2007	Bayswater Road	Galway Street	15:30	15:41	16:03	15:47	15:57	16:17	15:57
4	30/07/2007	Lever Street	Tyers Street	16:17	16:31	16:47	16:30	16:41	16:55	16:44
5	30/07/2007	Tyers Street	Bedford Square	17:02	17:13	17:21	17:12	17:20	17:31	17:18
6	30/07/2007	Bedford Square	Bayswater Road	17:38	17:44	17:53	17:44	17:51	18:00	17:56
7	31/07/2007	Bayswater Road	Whitechapel High Street	13:33	13:47	13:58	13:52	14:07	14:26	14:17
8	31/07/2007	Commercial Road	Jubilee Gardens	14:33	14:41	14:49	14:40	14:47	14:56	14:49
9	31/07/2007	Jubilee Gardens	Park Crescent	15:18	15:26	15:36	15:27	15:35	15:46	15:38
10	31/07/2007	Park Crescent	Old Kent Road	15:48	16:04	16:28	15:59	16:10	16:26	16:18
11	31/07/2007	Old Kent Road	Knightsbridge	16:34	16:48	16:53	16:45	16:53	17:05	16:56
12	01/08/2007	Bayswater Road	Potters Fields	8:50	9:05	9:18	9:07	9:22	9:43	9:29
13	01/08/2007	Potters Fields	Bedford Square	9:43	10:00	10:13	9:55	10:06	10:29	10:16
14	01/08/2007	Bedford Square	Tyers Street	10:30	10:39	11:01	10:39	10:49	11:03	10:49
15	01/08/2007	Tyers Street	Lever Street	11:33	11:46	12:05	11:50	12:02	12:11	11:58
16	01/08/2007	Lever Street	Carlton Gardens	12:23	12:35	13:05	12:32	12:41	12:54	12:45
17	01/08/2007	Carlton Gardens	Bayswater Road	13:10	13:18	13:21	13:18	13:23	13:23	13:22
18	02/08/2007	Bayswater Road	Old Kent Road	10:46	11:01	11:14	10:59	11:09	11:24	11:14
19	02/08/2007	Old Kent Road	Bedford Square	11:23	11:37	12:05	11:38	11:45	11:58	11:58
20	02/08/2007	Bedford Square	Tyers Street	12:19	12:28	12:42	12:30	12:40	12:52	12:40
21	02/08/2007	Tyers Street	Whitechapel High Street	13:02	13:12	13:24	13:13	13:21	13:33	13:20
22	02/08/2007	Commercial Road	Jubilee Gardens	13:32	13:42	13:56	13:41	13:48	13:58	13:49
23	02/08/2007	Jubilee Gardens	Bayswater Road	14:04	14:16	14:22	14:13	14:20	14:31	14:29

Table C.2-2: Recorded travel times (in minutes) and comparison between the two systems

Conv.		ARIAdNE				DATT	FDATT	FCATT	FAATT
CETT	CATT	AERTT	AETT	ALRTT	AATT				
14	23	14	27	45	26	-3	0	24.5	24.5
15	37	18	32	51	26	11	8	35.5	27.5
11	33	17	27	47	27	6	3	31.5	28.5
14	30	13	24	38	27	3	0	28.5	28.5
11	19	10	18	29	16	3	0	17.5	17.5
6	15	6	13	22	18	-3	0	16.5	16.5
14	25	19	34	53	44	-19	-16	26.5	42.5
8	16	7	14	23	16	0	0	16	16
8	18	9	17	28	20	-2	0	19	19
16	40	11	22	38	30	10	7	38.5	31.5
14	19	11	19	31	22	-3	0	20.5	20.5
15	28	17	32	53	39	-11	-8	29.5	37.5
17	30	12	23	46	33	-3	0	31.5	31.5
9	31	9	19	33	19	12	9	29.5	20.5
13	32	17	29	38	25	7	4	30.5	26.5
12	42	9	18	31	22	20	17	40.5	23.5
8	11	8	13	13	12	-1	0	11.5	11.5
15	28	13	23	38	28	0	0	28	28
14	42	15	22	35	35	7	4	40.5	36.5
9	23	11	21	33	21	2	0	22	22
10	22	11	19	31	18	4	1	20.5	19.5
10	24	9	16	26	17	7	4	22.5	18.5
12	18	9	16	27	25	-7	-4	19.5	23.5

Table C.2-3: Travel time difference ratios and corresponding cumulative probabilities

DATT/AATT	DATT/AATT CP	DATT/CATT	DATT/CATT CP	FDATT / FAATT	FDATT / FAATT CP	FDATT/FCATT	FDATT/FCATT CP
-0.115384615	0.260869565	-0.130434783	0.260869565	0	0.608695652	0	0.608695652
0.423076923	0.913043478	0.297297297	0.913043478	0.290909091	0.913043478	0.225352113	0.913043478
0.222222222	0.739130435	0.181818182	0.739130435	0.105263158	0.695652174	0.095238095	0.695652174
0.111111111	0.565217391	0.1	0.565217391	0	0.608695652	0	0.608695652
0.1875	0.608695652	0.157894737	0.608695652	0	0.608695652	0	0.608695652
-0.166666667	0.173913043	-0.2	0.173913043	0	0.608695652	0	0.608695652
-0.431818182	0.043478261	-0.76	0.043478261	-0.376470588	0.043478261	-0.603773585	0.043478261
0	0.47826087	0	0.47826087	0	0.608695652	0	0.608695652
-0.1	0.304347826	-0.111111111	0.304347826	0	0.608695652	0	0.608695652
0.333333333	0.826086957	0.25	0.826086957	0.222222222	0.869565217	0.181818182	0.869565217
-0.136363636	0.217391304	-0.157894737	0.217391304	0	0.608695652	0	0.608695652
-0.282051282	0.086956522	-0.392857143	0.086956522	-0.213333333	0.086956522	-0.271186441	0.086956522
-0.090909091	0.347826087	-0.1	0.347826087	0	0.608695652	0	0.608695652
0.631578947	0.956521739	0.387096774	0.956521739	0.43902439	0.956521739	0.305084746	0.956521739
0.28	0.782608696	0.21875	0.782608696	0.150943396	0.782608696	0.131147541	0.782608696
0.909090909	1	0.476190476	1	0.723404255	1	0.419753086	1
-0.083333333	0.391304348	-0.090909091	0.391304348	0	0.608695652	0	0.608695652
0	0.47826087	0	0.47826087	0	0.608695652	0	0.608695652
0.2	0.652173913	0.166666667	0.652173913	0.109589041	0.739130435	0.098765432	0.739130435
0.095238095	0.52173913	0.086956522	0.52173913	0	0.608695652	0	0.608695652
0.222222222	0.739130435	0.181818182	0.739130435	0.051282051	0.652173913	0.048780488	0.652173913
0.411764706	0.869565217	0.291666667	0.869565217	0.216216216	0.826086957	0.177777778	0.826086957
-0.28	0.130434783	-0.388888889	0.130434783	-0.170212766	0.130434783	-0.205128205	0.130434783

Appendix D: Statistical significance tests

The statistical significance of the results obtained from the single- and double-vehicle experiments is tested here.

D.1 Single-vehicle experiment

Three statistical significance tests are carried out for the single-vehicle experiment. The first one tests whether the AERTT is a good lower bound to the ATT, i.e. whether AERTT/ATT < 1, the second one assesses whether the ALRTT is a good upper bound to the ATT, i.e. whether ALRTT/ATT > 1, and the third one tests how well the AETT estimates the ATT, i.e. whether AETT/ATT = 1.

For the first test, the sample mean is $\overline{\mathrm{AERTT/ATT}} = $ 0.533, the sample variance $s_{\mathrm{AERTT/ATT}} =$ 0.116 and the sample size is $n = $ 72. The null hypothesis is H_0: $\mu_{\mathrm{AERTT/ATT}} = $ 1, and the alternative hypothesis is H_1: $\mu_{\mathrm{AERTT/ATT}} <$ 1, where $\mu_{\mathrm{AERTT/ATT}}$ is the mean AERTT/ATT. Based on these, the t-test value is:

$$t = \frac{\overline{\mathrm{AERTT/ATT}} - 1}{s_{\mathrm{AERTT/ATT}}\sqrt{n}} = \frac{0.533 - 1}{0.116\sqrt{72}} = -34.064$$

For n-1 degrees of freedom, i.e. 71, the corresponding value of Student's t-distribution for a one-tailed test at a significance level of 0.05 is $t_{71,0.05}$ = 1.666, and since $|t| > t_{71,0.05}$, the null hypothesis is rejected and the alternative hypothesis is accepted. The AERTT is thus a lower bound to the ATT to a 0.05 significance level.

For the second test, the sample mean is $\overline{\mathrm{ALRTT/ATT}} = $ 1.525, the sample variance $s_{\mathrm{ALRTT/ATT}}$ $=$ 0.276 and the sample size is again $n = $ 72. The null hypothesis is H_0: $\mu_{\mathrm{ALRTT/ATT}} = $ 1, and the alternative hypothesis is H_1: $\mu_{\mathrm{ALRTT/ATT}} >$ 1, where $\mu_{\mathrm{ALRTT/ATT}}$ is the mean ALRTT/ATT. Based on these, the t-test value is:

$$t = \frac{\overline{ALRTT/ATT} - 1}{s_{ALRTT/ATT}\sqrt{n}} = \frac{1.525 - 1}{0.276\sqrt{72}} = 16.128$$

For n-1 degrees of freedom, i.e. 71, the corresponding value of Student's t-distribution for a one-tailed test at a significance level of 0.05 is $t_{71,0.05}$ = 1.666, and since $|t| > t_{71,0.05}$, the null hypothesis is rejected and the alternative hypothesis is accepted. The ALRTT is thus an upper bound to the ATT to a 0.05 significance level.

For the third test, the sample mean is $\overline{AETT/ATT}$ = 0.949, the sample variance $s_{AETT/ATT}$ = 0.182 and the sample size is again n = 72. The null hypothesis is H_0: $\mu_{AETT/ATT}$ = 1, and the alternative hypothesis is H_1: $\mu_{AETT/ATT} \neq 1$, where $\mu_{AETT/ATT}$ is the mean AETT/ATT. Based on these, the t-test value is:

$$t = \frac{\overline{AERTT/ATT} - 1}{s_{AETT/ATT}\sqrt{n}} = \frac{0.949 - 1}{0.182\sqrt{72}} = -2.373$$

For n-1 degrees of freedom, i.e. 71, the corresponding value of Student's t-distribution for a two-tailed test at a significance level of 0.05 is $t_{71,0.05/2}$ = 1.994, and since $|t| > t_{71,0.05/2}$, the null hypothesis is rejected at the 0.05 significance level and the alternative hypothesis is accepted. Nevertheless, $t_{71,0.02/2}$ = 2.381, and as $|t| < t_{71,0.02/2}$, the null hypothesis is accepted at the 0.02 significance level. Hence, the AETT is a good estimate of the ATT to a 0.02 significance level, though not to the 0.05 level.

D.2 Double-vehicle experiment

Two tests are carried out for the double-vehicle experiment, so as to assess the statistical significance of the difference in actual travel time of ARIAdNE and the conventional systems, for both the unfiltered and filtered results.

To test the statistical significance of the unfiltered result a one-tailed t-test is used. Namely, the sample size is n = 23, the sample mean is calculated as $\overline{DATT}$ = 1.739 and the sample standard deviation is worked out as s_{DATT} = 8.225. The null and alternative hypotheses are thus formulated as H_0: $\mu_{DATT} = 0$ and H_1: $\mu_{DATT} > 0$, where μ_{DATT} is the mean DATT. Based on these, the t-test value is:

$$t = \frac{\overline{DATT} - 0}{s_{DATT}\sqrt{n}} = \frac{1.739}{8.225\sqrt{23}} = 1.014$$

For n-1 degrees of freedom, i.e. 22, the corresponding value of Student's t-distribution at a significance level of 0.05 is $t_{22,0.05}$ = 1.717, and since $|t| < t_{22,0.05}$, the null hypothesis cannot be rejected. Thus, ARIAdNE's advantage over the conventional system is not statistically significant at the 0.05 level. However, taking a significance level of 0.2, the corresponding value of Student's t-distribution is $t_{22,0.2}$ = 0.859, and since $|t| > t_{22,0.2}$, the null hypothesis is rejected and the alternative hypothesis is accepted. As such, ARIAdNE's advantage over the conventional system is statistically significant at the 0.2 level.

To assess the statistical significance of the filtered result, a t-test is used in the same way as in the unfiltered result assessment. Namely, for the filtered result the sample size is n = 23, the sample mean is $\overline{DATT}$ = 1.261 and the sample standard deviation is s_{DATT} = 6.181. The null and alternative hypotheses are again formulated as H_0: $\mu_{DATT} = 0$ and H_1: $\mu_{DATT} > 0$. Based on these, the t-test value is:

$$t = \frac{\overline{DATT} - 0}{s_{DATT}\sqrt{n}} = \frac{1.261}{6.181\sqrt{23}} = 0.978$$

As in the case of the unfiltered result, for n-1=22 degrees of freedom, the corresponding value of Student's t-distribution at a significance level of 0.05 is $t_{22,0.05}$ = 1.717, and since $|t| < t_{22,0.05}$, the null hypothesis cannot be rejected. Thus, the outcome is that, similarly to the unfiltered result, ARIAdNE's advantage over the conventional system is not statistically significant at the 0.05 level. However, taking a significance level of 0.2, the corresponding value of Student's t-distribution is $t_{22,0.2}$ = 0.859, and since $|t| > t_{22,0.2}$, the null hypothesis is rejected and the alternative hypothesis is accepted. As such, even for the filtered result, ARIAdNE's advantage over the conventional system is statistically significant at the 0.2 level.