# Calibrating route set generation by map matching GPS data


*Master thesis, Mike Fafieanie*
*Deventer/Enschede*
*September 2009*

Goudappel Coffeng
Consultants on traffic and transport

University of Twente
Enschede - The Netherlands

# Calibrating route set generation by map matching GPS data

Master thesis


Author Mike Fafieanie


Date 23td of September 2009

Reference

Version Final version

# Documentation page

Title report
Calibrating route set generation by map matching GPS data
Master thesis

Keywords
Dynamic traffic assignment, StreamLine, OmniTRANS, route choice, route set generation, map matching and GPS data

Author
Ing. M.E. Fafieanie
University of Twente
Centre of Transport studies
mike@ex-plore.nl

Committee members
Prof. Dr. Ir. E.C. van Berkum
University of Twente
Centre of Transport studies
e.c.vanberkum@utwente.nl

Dr. T. Thomas
University of Twente
Centre of Transport studies
t.thomas@utwente.nl

Dr. M.C.J. Bliemer
Goudappel Coffeng
mbliemer@goudappel.nl

J. Zantema MSc.
Goudappel Coffeng
kzantema@goudappel.nl

Date of publication
23td of September 2009

Goudappel Coffeng
Consultants on traffic and transport

University of Twente
Enschede - The Netherlands

# Summary


### *Motivation*

Every person on earth is faced with the daily need of transportation. The enormously increasing travel demand results in traffic problems, like the daily congestion on the highways. Traffic models have been developed to support decision making, which is trying to solve these problems with transportation policy, planning, and engineering.

One of the traffic models is the widely used four-step model. This model generates trips, distributes these trips, chooses a modal split and finally makes an assignment of the traffic throughout the model network. The route choice is modeled by generating a route choice set and then an application of a discrete choice model. The route choice set contains a set of "relevant" routes. For each OD-pair a choice set is constructed.

The route choice set has to include all relevant routes, as routes that have not been created, cannot be chosen in the route choice. Also, it is not advisable to include all available routes, because this results in an enormous computation time and there is no route choice model that can deal correctly with large route choice sets.

Therefore, we will calibrate the generation of route choice sets by using observed routes abstracted from GPS data.

### *Problem definition*

The current problem is that we do not have insights in the performance of the route set generation. It is interesting to know whether the choice set includes all relevant routes between an OD-pair. The route set generation is relative complex and uses many different parameters. We want to find an "optimal" parameter set that includes as many as possible observed routes, but also takes care of the route set size and the inclusion of non-motorway routes. Literature shows that non-motorway routes are often not included in a route set, even though these routes are often used to avoid congestion. The observed routes could also be used to determine why routes are not included and how the route set generation can be improved. All this will be investigated in this research.

### *Methodology*

The research consists of two parts, first we have to obtain observed routes and thereafter the actual calibration will be performed.

In order to obtain observed routes, we have to connect the GPS data with a model network. For this, a so-called map matching algorithm has to be implemented and calibrated. A literature study will be performed to investigate several map matching algorithms. Then a good algorithm will be selected based on the quality and calculation speed of the algorithm. The selected algorithm will be calibrated with a small portion of the GPS data to obtain a high matching quality and finally performed on all GPS data. The obtained map matched routes do not have to be relevant, therefore several filters are applied on this set of matched routes to create a set of relevant observed routes.

The observed routes will be used in the second part of this research. We assume that all observed routes are relevant and have to be included in the generated route sets. The observed routes also represent other relevant routes, which are not part of the observed routes. The purpose is to find a parameter set that maximizes the number of observed routes in the generated route set. An observed route does not have to be exactly the same as a generated route, because small deviations on local roads are not considered important. Because of this, the generated routes are filtered after the route set generation and not all the relevant routes are included, as another relevant route may be almost the same.

Besides the main parameters, two other criteria are used. At first, an average maximum of five routes per route set is allowed to prevent large route sets with the belonging disadvantages. In case that two parameter settings results in the same distance measure, the average number of routes is decisive. Another criterion is that ten important observed routes have been selected, which must be included in the final route set.

### *Results*

The results will discussed in two parts, the map matching results and the calibration of the route set generation.

The investigation of several map matching algorithms founds that Marchal (2004) is the most efficient and fastest algorithm (450 GPS points/s) to map match the GPS data. The purpose of the algorithm is to have a set of paths and choose the path that minimizes the distance between the GPS points and the matched route. The algorithm matches 89% of the routes correctly, which results in 2505 observed routes. These routes are investigated and finally 2136 routes are determined to be relevant for the calibration process.

The calibration of the route set generations also consists of two parts. First, the route set generation filters are determined by using the observed routes. For this, we investigated the observed routes and set the filter parameter values such that they will not remove the observed routes.

Second, the calibration of the route set resulted in two parameter combinations that have an equal value for the distance measure and for this the average number of routes is used to select the optimal parameter set. This parameter set results in a route set generation which includes 89% of the observed routes with an average of 5.03 routes per OD-pair.

An investigation of the observed routes that are not included shows that most routes are filtered because of the maximum number of routes criterion. As it is, too many irrelevant routes are generated, resulting in large route sets, as the current filters still accept many irrelevant routes.

### *Conclusions and recommendations*

This research presents a proper method to use observed routes to calibrate the route set generation.

The implemented map match algorithm of Marchal satisfies the expectations and is considered as a proper method to map match GPS data efficient. One important improvement is performed and several improvements are suggested to achieve better map match results. A suggested improvement to reduce the calculation time is to use less GPS

points, because our investigations show no quality drawback when less GPS points are used.

The performed calibration of the route set generation shows that even the "optimal" parameters cannot include all relevant routes. Several improvements are suggested, which could increase the match percentage. Most important is the use of distance instead of travel time for the filtering of routes. The travel time could not deal with different irrelevant routes on local roads and accepts these routes incorrectly.

It is recommended to investigate the possibilities of using GPS data for the calibration of traffic models or parts of this (e.g. junction modeling) and maybe to be input for traffic models (e.g. replace the traffic movement questionnaires or trip generation). In theory, the steps of the four-step model could be replaced by observed routes if there are enough routes to represent the entire route set. For this case, a method to rescale these routes to coming traffic situations has to be developed (e.g. prediction for the year 2020). At last, GPS data could support traffic research by supplying information about travel times, speeds, departure times and bottlenecks in the network.

# Samenvatting


### *Aanleiding*

Bijna iedere dag worden we geconfronteerd met het feit dat de huidige verkeersvraag de wegcapaciteit overschrijdt. Verkeersmodellen zijn ontwikkelt om ondersteuning te bieden door het inzichtelijk maken van de problemen en hoe maatregelen de situatie kunnen verbeteren.

Een veel gebruikt verkeersmodel is het vierstapsmodel. Dit model genereert ritten, distribueert deze ritten, kiest een vervoerswijze en deelt uiteindelijk het verkeer toe. Deze toedeling bestaat uit een route set generatie en route keuze. De gebruikte route set bestaat uit relevante routes. Voor elk HB-paar wordt een dergelijke route set gegenereerd.

De route set moet alle relevante route bevatten, omdat ze anders niet gekozen kunnen worden bij de route keuze. Hierbij moet er echter wel rekening meegehouden worden dat de set niet zeer groot is. Een grote set leidt namelijk tot zeer lange rekentijden en daarnaast kunnen de huidige route keuze modellen niet omgaan met grote route sets.

### *Probleem definitie*

Er is op dit moment geen goed inzicht in de correctheid van een route set. Het zou interessant zijn als we weten of de route sets nu daadwerkelijk alle relevante routes bevatten. De generatie van de route set is tamelijk ingewikkeld en gebruikt vele parameters. Dit onderzoek heeft als doel om een optimale parameter set te vinden die zoveel mogelijk geobserveerde routes bevat, maar tevens rekening houdt met de grootte van een route set. Daarnaast moet er gekeken worden of provinciale wegen wel mee worden genomen, aangezien onderzoek uitwijst dat routes over deze wegen vaak niet in een route set zitten. De geobserveerde routes kunnen ook gebruikt worden om te onderzoeken waarom ze juist niet worden meegenomen in de route set generatie.

### *Methodiek*

Het onderzoek bestaat uit twee onderdelen, als eerste het verkrijgen van de geobserveerde routes en ten tweede de calibratie van route set generatie met deze routes.

Om geobserveerde routes te verkrijgen moeten we GPS data koppelen aan een netwerk. Een zogenaamd map matching algoritme wordt gebruikt om dit te doen. Hiervoor wordt eerst een literatuurstudie uitgevoerd om te onderzoeken welk algoritme geschikt is. Op basis van de kwaliteit en de berekeningssnelheid wordt een keuze gemaakt. Het algoritme zal gecalibreerd worden om zoveel mogelijk routes correct te map matchen. Nadat dit gebeurd is kan het algoritme toegepast worden op alle GPS data. De routes die hieruit komen zijn wellicht niet allemaal relevant voor de calibratie, daarom worden er nog enkele filters toegepast die leiden tot de relevante geobserveerde routes.

Deze geobserveerde routes worden gebruikt in het tweede deel van dit onderzoek. We nemen hierbij aan dat alle geobserveerde routes relevant zijn en daarom onderdeel moeten zijn van de genereerde route sets. Daarnaast vertegenwoordigen de geobserveerde routes ook de relevante routes die niet tussen de geobserveerde routes zitten. Het doel nu is om de parameters te bepalen die het aantal geobserveerde routes in

de genereerde route set maximaliseert. Hierbij moet wel aangemerkt worden dat deze routes niet helemaal hetzelfde hoeven te zijn, omdat de kleine afwijkingen op lokale wegen niet belangrijk zijn. Daarnaast worden gegeneerde routes ook gefilterd en kan het dus goed voorkomen dat de gegeneerde route set de geobserveerde route niet accepteert, omdat een bijna gelijke route al in de genereerde set zit.

Ter ondersteuning van dit doel zijn er nog twee extra criteria. Als eerste mogen er gemiddeld gezien maximaal vijf routes per route set zijn. In het geval dat twee parameter sets tot dezelfde maximalisatie waarden leiden dan heeft de set die resulteert in de gemiddeld laagste hoeveelheid routes per route set de voorkeur. Verder zijn er tien geobserveerde routes gekozen die onderdeel moeten zijn van de genereerde route sets.

### *Resultaten*

De resultaten zullen in twee delen besproken worden, ten eerste de resultaten van het map match en daarna de calibratie van de route set generatie.

Het literatuuronderzoek naar een geschikt map match algoritme heeft geleidt tot de implementatie van Marchal's algoritme (2004). Dit algoritme heeft efficiënt (450 GPS punten/s) en nauwkeurig de GPS data gekoppeld aan het netwerk. De basis van het algoritme is om meerdere paden te onthouden en hieruit diegene te kiezen die de afstand tussen de GPS punten en de route minimaliseert. Het algoritme heeft 89% van de routes gematched wat resulteert in 2505 routes. De hierna toegepaste filters hebben de niet relevante routes weg gefilterd waardoor 2136 routes bruikbaar zijn voor het calibratie proces.

De calibratie van de route set generatie bestaat tevens uit twee delen. Als eerste zijn de filter waarden bepaald die de route set generatie gebruikt. Hiervoor zijn de geobserveerde route geanalyseerd en zijn de filter waarden zo ingesteld dat de geobserveerde routes niet worden verwijderd bij de filtering.

Ten tweede is de calibratie uitgevoerd die heeft geleid tot twee parameter sets die beiden route sets genereren die eventueel geobserveerde routes bevatten. Daarom heeft het gemiddeld aantal routes per route set doorslag gegeven voor de "optimale" parameter set. Deze set genereert route sets die 89% van de geobserveerde routes bevatten met een gemiddelde van 5.03 routes per OD-paar.

Een onderzoek naar de routes die geen onderdeel uitmaken van de genereerde route sets laat zien dat de meeste routes weg worden gefilterd door de maximale hoeveelheid routes. Dit komt omdat er teveel niet relevante routes worden gegeneerd waardoor het maximale aantal snel bereikt wordt zonder dat alle relevante routes al in de genereerde set zitten.

### *Conclusies en aanbevelingen*

Dit onderzoek beschrijft een methode die in staat is om met geobserveerde routes een route set generatie te calibreren.

Het toegepaste algoritme van Marchal voldoet aan de verwachtingen en is een goede methode om de GPS data efficiënt te map matchen. Een belangrijke verbetering is doorgevoerd en verder zijn enkele verbeteringen voorgesteld om een nog beter resultaat te bereiken. Zo kan de snelheid van het algoritme nog verhoogd worden door minder GPS

punten te gebruiken, aangezien ons onderzoek laat zien dat de kwaliteit hiermee niet achteruit gaat.

De calibratie van de route set generatie laat zien dat zelfs de "optimale" parameters niet zorgen voor route sets die alle geobserveerde routes bevatten. Er zijn wederom enkele verbeteringen mogelijk die dit percentage kunnen verhogen. Het meest belangrijke hierbij is het gebruik van de afstand voor de route filtering in tegenstelling tot de reistijd. Reistijd kan niet goed omgaan met niet relevante routes op locale wegen en accepteert deze routes.

Als laatste wordt er aangeraden om onderzoek te doen naar de mogelijkheden van GPS data om verkeersmodellen of delen daarvan (kruispuntmodellering) te calibreren. GPS data ook invoer zijn voor modellen zoals het vervangen van vervoersonderzoeken of de ritgeneratie. Het zou in theorie mogelijk moeten zijn om het vierstapsmodel te vervangen door geobserveerde routes als deze alle mogelijke routes kunnen vertegenwoordigen. Dan wordt de vraag hoe een opschaling gedaan moeten worden om toekomstige verkeerssituaties te voorspellen. Als laatste kan GPS data verkeersonderzoeken ondersteunen door informatie te geven over reistijden, snelheden, vertrektijden en vertragingen in een netwerk.

# Preface

This thesis is a result of the study during my graduation at the University of Twente, conducted at Goudappel Coffeng in Deventer. Although, my actual workplace was at the company OmniTRANS International.

Ever since I was a little boy, I have been interested in civil aspects. This started by drawing the most impracticable buildings and got more serious in the years afterwards. This resulted in the bachelor Civil Engineering at the Hogeschool of Amsterdam, which I finished about three years ago. I decided to do the master study, because I had the feeling that something was missing. In the years to come, I found the missing thing. I received the challenge to investigate and learn more in depth than only superficial. The belonging student life was wonderful to experience, especially the year where I was in the board of the climbing club. This all ends with this final thesis that is conducted in the last eight months.

In the beginning of this research, I found it hard to get track on the situation. The research range was large and it was difficult to focus completely on the research. When things became clearer, I started to get enthusiastic and time really flew past. Actually, I found it a pity that this research is already to its end, because many aspects are still very interesting to investigate and develop further.

Many people were important for me during the writing of this thesis. I would like to thank my daily supervisor Kobus for his help and the job to constantly improve my English writing. I am grateful to my professor Eric to introduce this master subject and for his assistance during the research. Further I want to thank Michiel for his support and Thomas for his reverse look on the research. My colleagues at OmniTRANS made each day enjoyable and taught me the many aspects belonging to a transport modeling company. A special graduate goes to Jacob who introduced me in the object-oriented programming world. Then I would like to thank my parents being always interested in my work and my friend Palau for giving me support and the necessarily distraction. At last, three years studying would have been really boring without my friends with whom I experienced a great time in Enschede and in all the other countries that we have visited together.

Deventer/Enschede, September 2009

Mike Fafieanie

# Contents

# 1 Introduction

*This chapter introduces the subject of my master thesis. First, section 1.1 gives a short introduction on the route choice in the transport modeling. This section is followed by the research background in section 1.2. In section 1.3 the research objective and questions are presented. The methodology to answer the research questions is discussed in section 1.4 and finally section 1.5 describes the report outline.*

## 1.1 Route choice in transport modeling

Every person on earth is faced with the daily need of transportation. The enormously increasing travel demand of all these people results in traffic problems, like the daily congestion on the highways. Traffic models have been developed to support solving the problems with transportation policy, planning, and engineering.

The well-known four-stage model, presented in Figure 1, is an often used transport model for the last decades. The model consists of four steps. First, trips are generated by using land-use data. After this, the trips are distributed, followed by the mode choice (e.g. public transport or car). Finally, the assignment phase generates sets of routes and assigns vehicle fractions to these routes by performing a route choice.

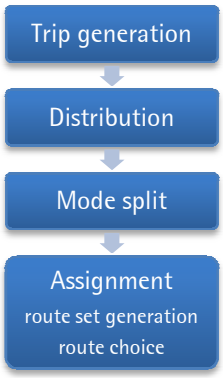



*Figure 1: four-stage model*

Being more specific, the assignment phase deals with route choices of travelers. A rational traveler is assumed to choose the route that has the lowest costs (e.g. travel time) for him. All these rationally chosen routes form the route choice set, briefly called the route set. Each route set contains the routes between a specified origin and destination. These route sets are important, because routes that are not included cannot be chosen during the route choice. On the other hand, it is also not desirable that a route set contains all available routes. There are two reasons for this. First, there is no route choice model that correctly deals with route choices for large route sets. Second, the computation time increases enormously by applying a route choice model on large route sets.

## 1.2 Research background

As reported in the previous section, the route set generation (RSG) has much influence on the results of a traffic model. Fiorenzo-Catalano (2007) found that the basic steps in most RSG algorithms are:

- Step 1: Search a route according to certain conditions;

- Step 2: Evaluate the route to a set of route criteria;
- Step 3: Select or reject the generated route;
- Step 4: Evaluate the resulting route set according to a set of criteria.

According to these basic steps, three parameter sets are available:

1. Certain conditions have to change to generate several best routes (e.g. by increasing the costs of the already generated route).
2. The best route has to be evaluated by a set of route criteria (e.g. route overlap or detour) resulting in a second set of parameters.
3. The entire route set is evaluated by a third set of criteria. Depending on the method of RSG, different parameters are variable, but they are all classified in these parameter set.

The parameter values described above influence the accepted routes in the route sets. These parameters are difficult to calibrate since there isn't much data is available about traveler's route choices and the routes they choose from. Therefore, the route choice is normally calibrated, in contrast to the RSG. This calibration is performed by receiving flows of measurement instruments like detection loops. These flows have to be equal to the predicted flows in the transport model. This is done by adapting the flows in the OD matrixes.

As described before, routes that are not generated cannot be chosen during the route choice. Also, there is no route choice model with can deal correctly with large route sets. These two problems clearly indicate the need to calibrate the RSG. These days, GPS data is becoming available more often through the increasing use of mobile phones and navigation systems. This GPS data consists of observed routes that may offer an opportunity to calibrate the RSG.

There is almost no literature available about the use of GPS data to calibrate the RSG, only Zantema et al. (2007) describe a method to compare route sets with observed routes abstracted of GPS data. The researchers compared 40 generated route sets with observed route sets. Furthermore, they compared the observed route sets of four selected OD-pairs with several generated route sets. The best generated route set is selected by comparing the match quality of the observed routes with the generated routes.

A drawback of the described research is the restricted use of observed routes to calibrate the RSG. The use of GPS data will be improved when all observed routes, abstracted of the GPS data, are compared with the generated route sets. To perform this job, the GPS data has to be connected with a network to receive routes that are comparable with the routes of the generated route sets. This job can be performed by a map matching algorithm (MMA), which finds the path that is the best estimation of the route that was taken by the user.

## 1.3 Research objective and questions

The previous section describes the background of calibrating the RSG, but actually the current knowledge is constrained about this subject. Therefore, it is interesting to perform a study about fine tuning parameters in a RSG by using observed data obtained by map matching GPS data. This results in an "optimal" route set.

Two main questions are formulated to support the accomplishment of the research objective:

1. What is the best MMA to obtain routes from GPS data and how can it be implemented?
2. What is the performance of the RSG with the "optimal" parameter values and how to gain these values?

## 1.4 Research methodology

The purpose of this research is to obtain an optimal route set by fine-tuning parameters. This route set is "optimal" when it satisfies certain criteria. To determine whether these criteria are met, the generated routes are compared with observed routes. These observed routes are obtained from map matched GPS data. Therefore, the research consists of two parts, the map matching and the calibration of the RSG by using the observed routes.

A literature review is performed to investigate several MMA's. With this information, an optimal algorithm is chosen by using an assessment framework (e.g. efficiency and quality). The chosen MMA uses several parameters that have to be calibrated. The best set of parameters is determined by applying several criteria (e.g. deviation, computation time). The GPS data will be map matched by using the optimal parameter settings. For each OD-pair $(i,j)$ in a predefined set of OD-pairs $OD$ there are actually chosen routes $RR_{obs}^{ij}$. However, not all the routes in $RR_{obs}^{ij}$ may be relevant. Therefore a filter $F$ is applied on $RR_{obs}^{ij}$ to obtain the relevant observed routes $R_{obs}^{ij}$. These routes will be used to calibrate the RSG.

The second part of this research, the route set calibration, starts with the determination of the route filters in the RSG. The RSG generates $RR_{gen}^{ij}, \forall (i,j) \in OD$. However not all the routes in $RR_{gen}^{ij}$ are relevant and therefore four filters are applied to obtain $R_{gen}^{ij}$. The parameter values of these filters are obtained by investigating $R_{obs}^{ij}$.

The RSG algorithm requires a number of parameters, defined as $p = (p_1 \dots p_n)$ so $R_{gen}^{ij} = R_{gen}^{ij}(p)$. The purpose of the calibration is to find the best $p$. Best means that $R_{gen}^{ij}$ and $R_{obs}^{ij}$ are as similar as possible. This similarity is indicated with a distance measure $\Delta$ as a function of a changing $p_i$. The purpose is to find the $p$ that maximizes the number of $R_{obs}^{ij}$ included in the $R_{gen}^{ij}$ so we want to find the p that maximizes $\Delta(R_{obs}^{ij}, R_{gen}^{ij}(p))$. The definition of $\Delta$ will be explained further.

Let $r$ be an element of an observed route $R_{obs}^{ij}$ and let $s$ be an element of a generated route $R_{gen}^{ij}$. We define $\delta(r,s) = 1$ when $r$ and $s$ are considered to be equal and $\delta(r,s) = 0$ otherwise. This equality is confirmed the overlap between $r$ and $s$ exceeds a threshold value, where overlap means the percentage of common links, weighted according to the distance.

The distance measure $\Delta$ is the percentage of routes $r$ in $R_{obs}^{ij}$ for which it holds that there is no route $s$ in $R_{gen}^{ij}$ where $\delta(r,s) = 1$. Let $E(s,A) = 1$ when $\exists r \in A: \delta(r,s) = 1$ and $E(s,A) = 0$ otherwise. So $E(s,A)$ indicates whether there exists a route in set $A$ that is considered equal to route $s$. This results in:

$$\Delta\left(R_{obs}^{ij}, R_{gen}^{ij}(p)\right) = \left(1 - \frac{\sum_{s \in R_{gen}^{ij}(p)} E\left(s, R_{obs}^{ij}\right)}{\left|R_{obs}^{ij}\right|}\right) * 100\%$$

Although, the purpose is to include as many observed routes in the generated route sets as possible, several restrictions are applied. At first, an average maximum number of routes in each generated route set will be allowed to prevent large route sets. Besides this, several selected observed routes must be included in the generated route set. These routes represent motorway and non-motorway routes and make sure that both sort of routes are included in the final generated route set. As it is quite unlikely that $\Delta = 100\%$ (a full match), this criteria takes care that at least these important routes are included ($\delta(r, s) = 1$). At last, in case that several parameter sets result in the same value of $\Delta$, the set with the average lowest number of routes in the generated route set is preferred.

## 1.5 Report outline

This report is structured as follows. Chapter 2 supplies a literature review about map matching algorithms followed by an assessment for the optimal algorithm for this research. A second review discusses route set generation generally and several RSG algorithms.

In chapter 3 the used GPS data is discussed and describes how trips are distributed of these data. After this, the study area and the transport network are discussed and compared with each other.

Chapter 4 focuses on the implementation and calibration of a map matching algorithm. A case study is performed and the belonging results are presented in this chapter. The theory of the route set generation is discussed in chapter 5.

The knowledge and results of the previous two chapters is used to set up chapter 6, which analysis the observed routes. Thereby, these routes are used to calibrate the filter parameters of the RSG. After this, chapter 7 describes the actual calibration of the RSG and results in an "optimal" parameter set.

Finally, in chapter 8 the findings of the previous chapters are summarized and several recommendations are presented for further research.

# 2 Literature review

*This chapter discusses a literature review on map matching algorithms and route set generation algorithms. Section 2.1 discusses some map matching algorithms and section 2.2 describes several route set generation algorithms.*

## 2.1 Map matching

The map-matching problem consists in finding the path that is the best estimation of the route that was taken by the user. Many researchers developed algorithms to map match GPS data. This section starts with information about GPS and maps and supports the use of map matching, general information about map matching and an overview of developed algorithms.

### *2.1.1 GPS*

GPS is a global navigation satellite system (GNSS) and is developed by United States Department of Defense. It is the only GNSS in the world and can be used freely. Between the 24 and 32 satellites transmit microwave signals that allow GPS receivers to determine their current location.

The accuracy of the GPS signal depends on the number of satellites that are found by the GPS receiver. It is determined with the deviation, which is the difference between the exact physical position and the position determined by the GPS device. In a city with high buildings, the receiver cannot find many satellites, because the buildings disturb the GPS signal, which results in a high deviation (e.g. 25 meters). Vice-versa, the deviation in the middle of a dessert will be really low (e.g. 4 meters).

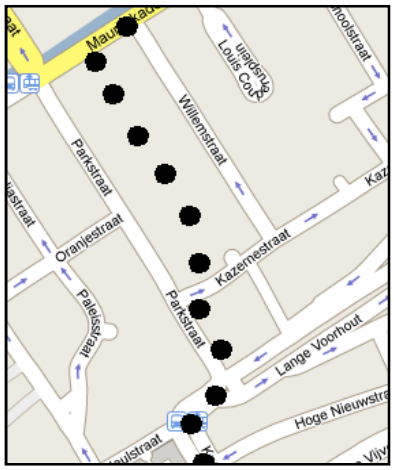



*Figure 2: deviation of GPS points (Google Maps)*

A low accuracy of the GPS signal is one of reasons map matching is needed, but also makes it more difficult. Figure 2 shows a network with two parallel roads with a distance of 40 meters between them. The GPS points have an accuracy of 20 meters and are positioned between the two roads. This results in the question how the vehicle exactly has driven.

### 2.1.2 Maps

Another difficulty of map matching is the map self. A map is a simplified representation of the real traffic network, which could result in for example missing roads. In this case, vehicles are map matched to irrelevant roads.

On the other hand, a "perfect" network won't provide perfect map matching, because it is much more difficult to determine the correct link in a high scale network, as shown in results of Quddus (2006). This is also visible in Figure 2, the network is very detailed, which makes it difficult to map match the GPS points to the correct roads.

### 2.1.3 Terminology

Map matching methods use a few terms that are important to understand correctly:

- Heading: the direction in which the vehicle drives (degrees);
- GPS point: a GPS point is a single point that is positioned by coordinates (*x* and *y* value);
- Trajectory: this is the path (not the path as below) of a moving object that it follows through space;
- Node: links are connected to each other at a node (intersections);
- Link: a link connects nodes with each other (representation of a road section);
- Formpoint: formpoints are positioned between two nodes to give shape to the link;
- Path: a path exists of sequence links;
- Route: a route exists of paths that don't have to be connected with each other (e.g. no signal in tunnel);
- Odometer: a device to measure the covered distance.

### 2.1.4 Methods

**Offline and online**

More than 35 map matching algorithms are produced and published in the literature during the period 1989-2006. Yin and Ouri (2004) roughly classified online map-matching and off-line map matching. Online map matching determines during a trip, in real time, the road segment on which the vehicle currently is located. Quddus et al. (2007) provides a good overview of this classification. A characteristic of online map matching is the slowness of the algorithm, because they don't have to perform faster than real-time. Even so, most of the recent map matching research is about online map matching due to the growing need for ITS devices (e.g. navigation systems).

Offline map matching is appropriate for analyzing historic data and is aimed to be fast. Different algorithms have been developed such as an efficient post-processing map-matching method for large GPS data (Marchal, 2004), a non-generic odometer map matching (Taylor et al., 2006), a weight-based map matching method (Yin and Wolfson, 2004), incremental algorithm with consecutive portions (Brakatsoulas, 2006), a global algorithm comparing the entire trajectory (Brakatsoulas, 2006) and a high integrity algorithm based on the topological method (Quddus, 2006).

Marchal (2004) developed an offline algorithm that is 1.000 times faster than the collection time of the data in comparison to online map matching algorithms. This indicates an online map matching model is not applicable in this research, as the number of GPS data is quite large. In the future, GPS data files will be much larger, so looking to the future; the offline map matching method is the most appreciated. In order to make a

good decision between the three offline map matching methods, they are described in the next paragraph.

**Offline map matching methods**

Within the offline map matching algorithms, three methods are distinguished. There are map-matching methods that use only geometric information, those using topological information as well and the more advanced map matching algorithms. When using only geometric information, one makes use only of the "shape" of the arcs and not of the way in which they are "connected". When using topological information one makes use of the geometry of the arcs as well as of the connectivity, proximity and contiguity of the arcs. Thus, the match is made in context and in relationship to the previous matched GPS point (Greenfield, 2002). The advanced algorithms use more refined concepts such as a Kalmam Filter, Dempster-Shafer's theory, a fuzzy logic model or the application of Bayesian interference.

### *2.1.5 Algorithms*

The simplest algorithm is described by Bernstein and Kornhauser (1996), which uses point-to-point matching whereby each position is matched to the closest road segment. This approach is easy to implement and very fast, but will in practice lead to topological connection problems. Once an incorrect link is selected because of an outstanding GPS point, this cannot be undone and the route will be incorrect.

Marchal (2004) developed a map matching algorithm that only used coordinates collected by GPS. The focus was to develop a fast algorithm for large volumes of data with reasonable matching errors. This is done by selecting the nearest links for each GPS point by an algorithm of White (2000). A path is created for each candidate link. For a new GPS point, the path will be extended by new links and taking care of the network topology. Finally, the most likely path is the one with the lowest deviation between the GPS coordinates and the coordinates of the path.

The odometer map matching algorithm (Taylor et al., 2006) is adapted to incorporate positioning based on odometer derived distances (OMMGPS), when GPS positions are not available. The odometer measures the driven distance until the GPS is back online. A map match technique finds the possibilities in the network of how the vehicle probably drove according to the measured distance. The most likely path is chosen and included in the route.

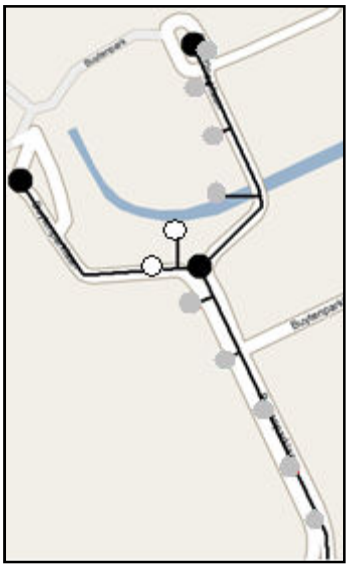


*Figure 3: local look-ahead method (Google Maps)*

Yin and Wolfson (2004) developed a weight-based method. This method computes the distance between the path of GPS points and all links. The weight of the link is a combination of the distance to the path and the heading of the same path. The chosen links are the one with the smallest total weight in relation to the path between the start link and end link.

The incremental algorithm of Brakatsoulas (2006) needs speed, heading, and the network topology to map match the GPS data. The algorithm uses two similarity measures to evaluate the candidate links for a GPS point (Greenfield, 2002). The speed and heading have a scale

factor, which determines the influence of the variables in relation to each other. The link with the highest score is chosen and linked to the GPS point. From there on, a local look-ahead method is introduced. This method takes the last links into account to be sure the correct link is chosen after an intersection. Figure 3 shows the usefulness of this method, the GPS points with a white background are linked to the closest left link, but actually the driver takes the right road. The four grey GPS points behind the two white points overrule the incorrect chosen links, because the total distances between the GPS points and the right link are much lower than to the left link. This prevents the choice of an incorrect link.

The global algorithm of Brakatsoulas (2006) tries to find a path in the road network that is close to the vehicle trajectory (also a curve). The comparison between the paths routes is employed with the Fréchet distance (Fréchet, 1906). All possible paths between the origin and destination are compared with the vehicle trajectory and the path with the lowest difference in distance is chosen.

Quddus (2006) developed a high integrity map matching algorithm based on the topological method. First, the topological algorithm determines the closest node from the GPS point and then selects all links connected to this node as candidate link. A weighting formula selects the correct link for the GPS point by weighting the heading, perpendicular distance and the relative position between the links and the GPS point. Finally, the algorithm determines if the vehicle made a turning movement by checking the heading difference for the next GPS point. If the vehicle made a turning movement, the process starts again, otherwise the second GPS point is also linked to the same link.

### *2.1.6 Assessment*

Four algorithms (Bernstein and Kornhauser, 1996, Marchal, 2004, Brakatsoulas, 2006 and Quddus, 2006) are relevant for the map matching of the GPS data in OmniTRANS. The other algorithms cannot be used because of unknown variables (e.g. odometer) or the lack of a clear description. The four chosen algorithms are compared in the table below.

| | Bernstein and Kornhauser (2006) | Marchal (2004) | Brakatsoulas (2006) | Quddus (2006) |
|---|---|---|---|---|
| Method | Geographic | Topological | Topological | Topological |
| Variables | GPS coordinates | GPS coordinates | GPS coordinates, speed, heading | GPS coordinates, speed, heading |
| Determine distance method | X | White (2000) | Greenfield (2002) | Greenfield (2002) |
| Calculation speed for high resolution* | Very fast | Fast (2,000 GPS point/s) | Middle | Middle (408 GPS point/s) |
| Correct link identification (%)* | Low | High (95.5%) | High | High (88.6%) |
| Detail of description | Very high | High | Middle | Very high |

** Comparison of the algorithms can only be performed on the same data sets (Marchal, 2004). The values above provide only a global impression and cannot always contains a value, because the lack of data.*

*Table 1: overview of offline map match algorithms*

It is difficult to make a correct comparison between the algorithms, because no case studies are performed on the same data set. Despite the lack of comparable data, Marchal's algorithm seems to perform better than the other two algorithms, despite the two extra variables used by Quddus and Brakatsoulas. This choice is based on the fast computation time and high correct link identification of the algorithm. Therefore Marchal's algorithm will be used for this master thesis. If any unexpected problems occur with this algorithm, the algorithm of Quddus is a good second alternative.

## 2.2 Route set generation

A route set is defined as the collection of travel options that satisfy the travel demand of travelers. In case of a multi-modal network, we talk about a choice set, but for this research only vehicle trips are taken into account.

Several procedures exist for the generation of route sets. The constrained enumeration approaches uses a set of constraints that reflects the observed travel behavior, the so-called branch-and-bound algorithm to add routes to a route set (e.g. Hoogendoorn-Lanser, 2005). Another method is the use of repeated (stochastic) shortest path methods, which randomly add routes in a route set (Fiorenzo-Catalano et all., 2004). This last method is investigated during this research and will be discussed in more detail.

### *2.2.1 Repeated shortest path method*

Fiorenzo-Catalano (2007) found that the basic steps of the most route set generation (RSG) algorithms, based on the repeated shortest path method, are according to the next steps:

Step 1: Search a best route according to certain conditions;
Step 2: Evaluate the route to a set of route criteria;
Step 3: Select or reject the generated route;
Step 4: Evaluate the resulting route set according to a set of criteria.

Before the evaluation of alternative routes can be performed a first route must be selected. In almost all approaches, this route is defined as the shortest route, whereby the shortest route is the route with the lowest costs (e.g. travel time or distance). The shortest route is assumed to be correct and is not checked on a set of criteria, because it cannot be compared with other routes. The next section discusses the methods to generate the shortest route.

### *2.2.2 Shortest route generation*

The shortest path searching problem is the process of finding a path between two nodes such that the sum of the weights of its constituent links is minimized. Besides the shortest path between two nodes, there are three other generalizations that can be solved by shortest path algorithms. The four generalizations of the shortest path problem are:

1. Single-pair (one to one): path between two nodes
2. Single-source (one to all): path from a source node to all other nodes
3. Single-destination (all to one): path from all nodes to one destination node
4. All-pairs (all to all): paths between every pair of nodes

The most shortest path algorithms, except Dijkstra's algorithm, are directed on just one generalization. There are three general used algorithms for finding the shortest paths in road networks; these are Dijkstra (1959), Floyd (1962) and Hart et al. (1968). The three algorithms are presented in the table below with the belonging generalization for which they are designed.

| Algorithm | Single-pair | Single-source | Single-destination | All-pairs |
|---|---|---|---|---|
| **Dijkstra** | x | x | x | |
| **Floyd-Warshall** | | | | x |
| **Hart (A*)** | x | | | |

*Table 2: several shortest path algorithms with the belonging generalization*

### *2.2.3 Route set generation*

The actual route set generation is performed with the use of the determined shortest route. The problem is to determine the probability that a particular route X is part of the choice set of individual Y, dependent on the characteristics of both the network and the traveler. Different approaches have been developed for this problem; Fiorenzo-Catalano (2007) provides a good overview with four components that can be used or combined to determine alternative routes.

1. Change network attributes
   A simple example of this component is called link penalty, which increases the impedance on links used by the previously-identified shortest paths when searching for new paths. De la Barra, Perez and Anez (1993) describe a technique by which the shortest path is identified, impedance on those links is increased by a fixed percentage, and the shortest path calculation repeats.

2. Change route criteria
   The change of route criteria can be performed by labeling. Ben-Akiva et al. (1984) have proposed a labeling method using a large number of optimality criteria based on surveyed choice motivations. An optimal path is found for each of the criteria: travel time, distance, scenery, congestion, etc.

3. Change restriction criteria
   This component forces the shortest path to include some links; such links are included in the criteria.

4. Check constraints
   The last component consists of constraints (e.g. overlap, detour-max and detour-min constraints). The generated alternative paths are checked with the constraints and are only accepted when they satisfy the constraints.

Several route set generation approaches perform some criteria on the alternative routes and the entire route set. Fiorenzo-Catalano (2007) presents a framework containing requirements for an adequate choice set and on appropriate choice set generation process.

Requirements for each individual route

- Acyclic criterion: a reasonable route does not contain loops;

- Detour criterion: a reasonable route does not exhibit a detour from the shortest possible connection in terms of one or more measures such as distance or time between origin and destination larger than a maximum threshold ∝ (e.g. 50%);
- Hierarchic deviation: a reasonable route is constituted of a systematic sequence of functional link levels in the network, avoiding route parts going from higher to lower level links and back (e.g. driving at the A1, the N344 and then again the A1).

Requirements for choice set on individual level (OD-pair)

- Overlap criterion: the mutual overlap between two routes should be less than a determined percentage with respect to the shorter one of the two routes.
- Comparability criterion: the travel disutility between two routes should be comparable within a given threshold;
- Detour-max criterion: the non-common parts of two partly overlapping routes should have a maximum detour;
- Detour-min criterion: the two partly overlapping should have a minimum detour between the two routes not smaller than a given percentage;
- Choice set size criterion: the choice set should contain a limited number of alternatives.

Requirements for choice set on group level (OD zone)

- All the criteria on individual level
- Spatial variability criterion: routes of the choice set should be spatially different with respect to the links used
- Preferential variability criterion: routes of the choice set should represent the taste variation of each group of travelers.

### *2.2.4 Route set generation algorithms*

This section presents several route set generation algorithms with their advantages and disadvantages. The in the previous section described components and requirements are several times used. It is important to notice that the choice of a route set generation depends on the kind of network. For example, many alternative routes are relevant in large cities, this in contrast to a global network of the Netherlands with only two or three relevant routes per OD-pair.

**Compute all acyclic routes**

This method finds all routes except the cyclic routes, which are logically irrelevant. The storage of all these routes could result in problems especially when performing the method on large networks. This approach is not useful for transport models because the enormous number of routes results in much computation time, even if restrictions are used.

**Compute the k-shortest routes**

This method determines the shortest acyclic route, followed by the 2$^{sd}$ shortest, 3$^{rd}$ shortest etc., till the *k* shortest routes are found. In comparison with the acyclic method, this method will results in less routes. Nevertheless, many irrelevant routes will be generated because the routes are not assessed with requirements. The inclusion of relevant routes with a high detour will cause large route sets, because the most routes with a high detour are irrelevant. This makes this method not practical.

Van Der Zijpp and Fiorenzo-Catalano (2005) developed a constrained k-shortest route by which the routes have to satisfy predefined constraints. The method performs better than the k-shortest route method, but is slow and has a difficulty to generate particular alternatives.

**Compute the essentially least cost routes**
The method uses the same approach as the k-shortest path, but instead of the *k* shortest routes, all shortest routes are found within a predefined bandwidth. This bandwidth is defined as the maximum increase of the costs. For example, if the shortest route is 10km long and the bandwidth is 1.1, all routes shorter than 11km are found. The advantage in contrast with the k-shortest route method is the fact that all routes within specified cost bandwidth are included and not just a predetermined number of routes. However, just like the two methods before, the overlap and the number of routes are high and not adaptable; therefore this method is not practicable in use (Fiorenzo-Catalano, 2007).

**Compute all efficient routes**
When more criteria are available, this method could be useful. The method finds the paths that minimize the costs function of a weighted sum of path attributes. This method is not often used and therefore no results are available to judge this method.

**Compute a column generation**
This method finds the shortest route after each iteration step of an equilibrium assignment (e.g. performed stochastic of deterministic). This shortest path is calculated with the network conditions of the previous iteration step. For example, 10 iterations could result in a maximum of 10 new routes, but probably some routes are identical. Because congestion influences only a few links, it is likely that this method finds many routes with only small detours. It will take much time before the useful alternative routes with a large detour are found. This method could be useful for networks without many short detours like motorway networks.

**Compute the most probable routes**
The most probable routes are determined with this method and all routes that are likely to be chosen will be accepted. This method uses the Monte Carlo approach (Sheffi & Powell, 1982), which randomizes all links for one or more attributes (e.g. free flow time or distance) for each iteration to cause perhaps new shortest paths. These new paths are checked on detour and overlap in comparison with the earlier accepted routes and if they satisfy these criteria, they are accepted. The approach stops when the maximum number of iterations is obtained. The method is a described as a proper method for generating route sets especially is some additional criteria like are applied like detour and overlap filters (Fiorenzo-Catalano, 2007).

The accelerated Monte Carlo approach increases the variance, which means that the links are randomized over broader range and routes with a higher detour are found faster. In this case, the method stops when the maximum variance or the maximum number of iterations is obtained. Fiorenzo-Catalano (2007) investigated this method without filter criterion and concludes that this method may be applied for generating choice sets. A drawback of the method is that randomizing link attributes may determine more differentiation of routes in terms of costs, but not necessarily in terms of spatial difference. The method may generate much more routes than the Monte Carlo approach, because the increasing variance, which might result a route set with too many routes.

### 2.2.5 StreamLine

The modeling software that was used for this research is OmniTRANS, developed by OmniTRANS International in the Netherlands. This program is a software environment for transport planning and modeling. This software package contains a dynamic traffic assignment framework called StreamLine.

StreamLine is developed to be a framework for the assignment phase in transport models. A part of the framework is the possibility to implement different route generation algorithms like the described methods in the previous section. At the moment only several default methods are available to generate route sets. The shortest path generator is Dijkstra's algorithm and the route set generator is based on the accelerated Monte Carlo method.

### 2.2.6 Conclusions

The route set generation has the purpose to generate sets with relevant routes. A relevant route is defined by an accepted route that is taken by at least one person in the respondent group. The previous sections describe several methods and procedures to generate route sets. This research will use the default route set generator of StreamLine and actual method choices are not possible. Nevertheless, it is important to recognize advantages and disadvantages of the used methods within StreamLine abstracted from the literature.

The StreamLine route set generator uses a repeated shortest path method, which assumes that the shortest path is correct. It might be possible that the requirements for an individual route are not obtained. Besides this, once a route is accepted, the route cannot be removed from the route set. These two drawbacks are important to realize, because they will probably occur during the route set calibration.

Furthermore, Dijkstra's shortest path algorithm is used. This algorithm is usable for one to one, one to all and all to one generalizations of the route set. Nevertheless, the algorithm could also be used for the all to all generalizations by performing an one to all method for each origin zone. This is not very efficient, in comparison with of an algorithm directed on the all to all method, but is much faster than individual one to one generalizations for each OD-pair.

StreamLine uses the accelerated Monte Carlo method to generate route sets. Fiorenzo-Catalano (2007) concludes that this method may be applied for generating route sets. The method cannot guarantee for exhaustiveness of all attractive alternatives. This is important to realize as it will make a full match of all observed routes very unlikely. Besides this, the method is not focused on spatial difference, but on differentiation of the routes in terms of costs. At last, the increasing variance might result in many more routes than desired, it is important to take this into account and take care of large route sets.

# 3 Data

Two data sources are used for this master thesis, the GPS data and the model network. This data has much influence on the research and the results. This chapter describes both data sources.

*The GPS data consists of the actual trips made by drivers, which we want to use for our calibration. This data is described in section 3.1. Section 3.2 discusses the model network which is used for this research.*

## 3.1 GPS data

### *3.1.1 Source*

In the year 2006, a peak avoidance strategy pilot called "Spitsmijden" has been performed between Zoetermeer and Den Haag (Knockaert, 2007). This project was set up to study the feasibility of a reward scheme to encourage commuters not to drive during the morning rush hour. A total of 340 respondents were selected to participate in this research. Each respondent commutes at least three times per week from Zoetermeer towards The Hague.

The respondents had to avoid the rush hours to receive a reward of money or a Yeti Smartphone. This Smartphone was chosen by 108 respondents, they received the Yeti at the beginning of the pilot. The Smartphone is equipped with a GPS receiver to register the position of the car. The obtained GPS data was used to analyze whether the participations avoid the rush hours and adapt their departure time. On the other hand, the data is also used to check the driver's logbook.

The GPS data include all respondents' chosen routes independent of origin, destination or rush hours. The "Spitsmijden" project only used the GPS data of the rush hours with an origin in Zoetermeer and destination in The Hague. For this research, all relevant routes will be used.

### *3.1.2 Selecting relevant GPS data*

The purpose of the GPS device in the Yeti Smartphone is to calculate the location and store this position in a longitude and latitude coordinate. Besides the position of this GPS point, much other information is stored like respondent name, unixtime, signal validity factor and many more. This results in a file size of 1.2 GB with 9.5$M$ GPS points (144$B$ per GPS point. The GPS points are recorded once per second. Because a large file size is not appropriate to perform some operations, all unnecessary data is removed.

The file size is reduced eight times by using four criteria. At first, all unnecessary column-data (e.g. signal validity) is removed, which result in only five data columns including ID

number, yeti username, time and position (longitude and latitude). Besides this, all sequent double GPS points (both longitude and latitude position are similar) and all GPS points outside the network are removed. The last adjustment is the elimination of all GPS points that were recorded before the 28th of September 2006, because the respondents did not receive the Smartphone before this date. The GPS points before this date have been recorded during some test trips.

### 3.1.3 GPS points to trips

The GPS points have to be divided in groups of GPS points that constitute a trip. There are three criteria applied to create these groups. At first, in case the username changes, a new trip will start.

Furthermore, the maximum time between two GPS points is set to 60 seconds. A longer time period results in one trip that consists of two trips (e.g. dropping the kids at school when driving to work) and a shorter time period results in dividing trips, which do not have to be split in two trips (e.g. waiting for a signalized intersection). In case of congestion inside a tunnel, the GPS signal will not be available. If the vehicle is in the tunnel for more than 60 seconds, the route will be incorrectly divided in two routes. Fortunately, the network consists of only one long tunnel.

The last parameter is the maximum distance difference between two GPS points. This value is set to 500m, because the longest tunnel in the network is 450 meters (central station of The Hague). The routes that include this tunnel will not be divided in two separated routes in case there is no congestion as discussed in the previous section. The map match algorithm cannot match the trip correctly to the network if the GPS signal is not available for a large distance.

The table below summarizes the three criteria to create these groups. In case that one of the criteria is obtained, a new group will be generated.

| Criteria | Value |
|---|---|
| User difference | change |
| Time difference | 60s |
| Distance difference | 500m |

*Table 3: criteria with belonging values that are used to create groups of GPS points*

Groups of GPS points are also judged on the total number of GPS points. This prevents unnecessary map matching of irrelevant trips with less GPS points. A group has to consist of at least 200 GPS points, because less GPS points will result in routes that are too short (e.g. approximately less than 2.5km) to be a relevant observed route.

### 3.1.4 Statistics

Before the map matching is performed, some statistics can already be determined by analyzing the trips extracted from the GPS data. The first notable case is the fact that the only 85 travelers of the group of 108 contribute to the GPS data. This difference is explicable by the case that several respondents took the public transport and some GPS receivers did not work probably.

During 118 days, the respondent's car trips are recorded. A total of 5444 trips can be traced of the GPS points by applying the parameters of the previous section. The number of routes per day and the departure times are presented in the figures below.

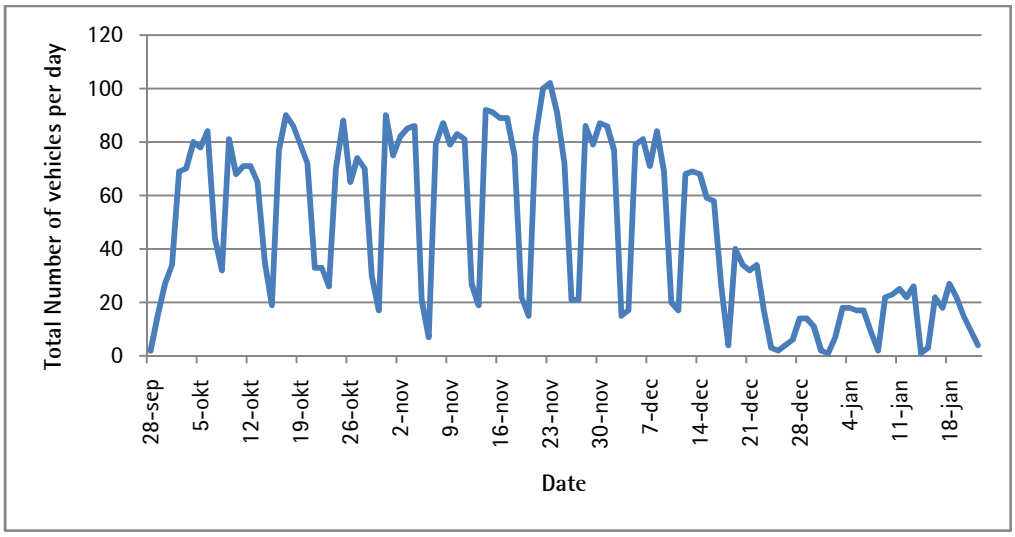



*Figure 4: the number of vehicles per day*

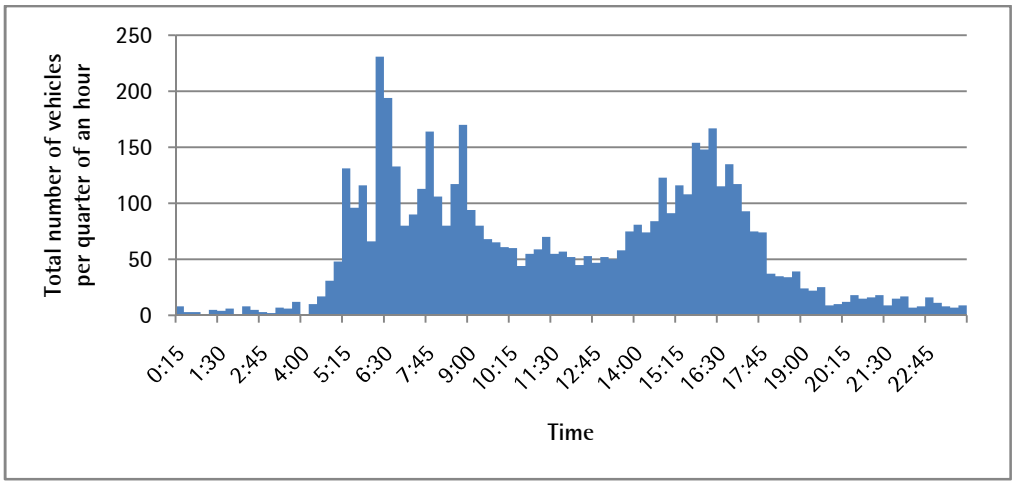



*Figure 5: the total number of vehicles (of all days) per quarter of an hour*

The tables show the large number of vehicles on work days (e.g. approximately 80 vehicles per work day). Besides this, the two rush hours are indicated well and an average of 50 vehicles is departing between both rush hours. At last, both tables show a good distribution of the vehicles over the day and week that represent the total vehicle distribution well.

The location of a GPS point might have a deviation as concluded in the literature review. An analysis of the trips shows that especially in the city Den Hague deviations occur. The maximum detected deviation is approximately 50 meters. In the network there are a few tunnels. Inside these tunnels, the GPS receiver cannot determine the vehicle's position. After the vehicle passes a tunnel, it takes some seconds before the GPS receiver can determine the correct position. Figure 7 shows an example of a tunnel in the network for which it took some time before the GPS receiver recovers the signal.

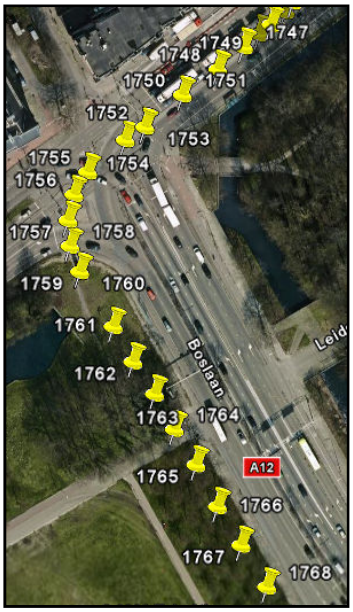



*Figure 6: Deviation in GPS signal (Google Maps)*

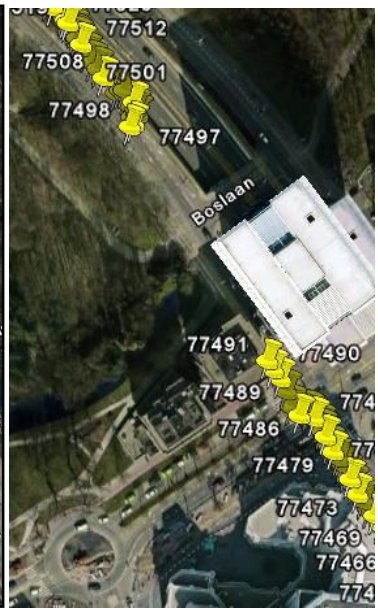



*Figure 7: loss of GPS signal in a tunnel (Google Maps)*

## 3.2 Study area

### *3.2.1 Area*

*Figure 8: Haaglanden network*

This research takes place in the Haaglanden region including the cities The Hague, Zoetermeer and Delft. The Haaglanden network in OmniTRANS is shown in Figure 8. The cities are connected with highways that are often congested in the rush hours. Alternatives are some secondary roads between the cities. There are some highway ramps between Zoetermeer and The Hague where it's possible to switch between the highway and the secondary roads. Inside the cities, especially in The Hague, many roads lead to the

same destination. Drivers could easily change their route in the city at the moment of driving (e.g. a truck is unloading).

### 3.2.2 Network model

The representation of the physical transportation network is the Haaglanden network in OmniTRANS. This network includes the most roads, only local roads (30km/h roads) are not included. Figure 9 shows the differences between the model network and the physical roads. The model network consists of 2084 mostly two-way roads (links) and 168 origins / destinations (centroids). Intersections are represented as nodes, but nodes could also split a curved link. There are 1295 nodes in the network. These network objects have their own attributes like a link which include also the maximum speed, number of lanes, allowed driving directions and maximum capacity.

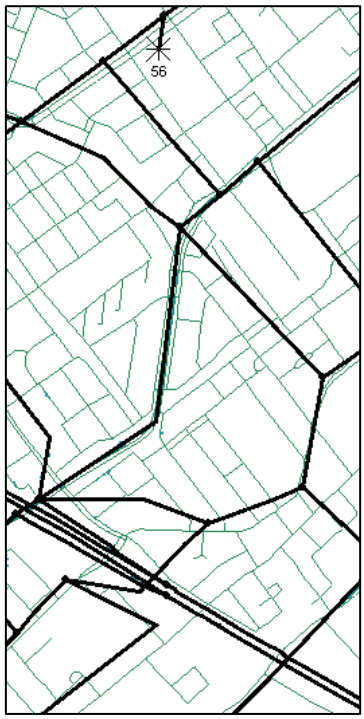



*Figure 9: network model (bold) and the road network (thin)*

During the research, several errors have been found in the network. These errors like e.g. incorrect link attribute (e.g. incorrect speed) or non-existing roads have been adapted. The most important changes are made for links with a large curve. These links are separated into several links by inserting a node to increase the quality of the map matching algorithm. This will be explained further in section 4.2.2 about the deviation of the map matched routes. All adaptations in the network were performed after detecting incorrect results during the research.

# 4 Map matching

The map matching's purpose is to find the path that best represents the route that was taken by the driver. This route is represented by GPS points that are stored with a GPS device like a mobile phone or a navigation system. The GPS points consist of coordinates and will be matched to the links in a network model. In the last decades, many algorithms have been developed to perform this job, but no algorithm results in a 100% correct match. A new map matching algorithm based on Marchal (2004) is developed to map match GPS points to an OmniTRANS network model. The GPS data, described in the previous chapter, will be map matched to test the map matching algorithm and after this, the map matched routes will be used for the route set calibration described in chapter 5.

*Section 4.1 introduces the problem statement of map matching shortly. After this, section 4.2 discusses the theory of the map matching algorithm. The approach is described in section 4.3, followed by the case study description in section 4.4. The results of this case study are presented in section 4.5 and a second case study is shortly described in section 4.6. The fine tuning process is discussed in section 4.7 and the limitations and recommendations in section 4.8. Finally the conclusion is presented in section 4.9.*

## 4.1 Problem statement

The goal of the matching algorithm is to identify the route represented in links taken by a driver equipped with a GPS. Because the GPS receiver is only used in the vehicle, we can restrict ourselves to a network representation of the physical transportation network. There are two sources of errors that make map matching quite difficult. First, the GPS receivers have a deviation as described in section 3.1.4. Second, the used network is a simplified representation of the psychical world that is discussed in section 3.2.2. The combination of these two errors is shown in Figure 10; the network (bold lines) is not on the exact position of the physical network (thin lines) and the GPS device has a deviation (sequent points) in comparison to the physical network.

*Figure 10: GPS points plotted in Haaglanden network (bold lines)*

An important addition to this problem statement is the computation time. GPS data could easily include millions of GPS points. The map matching algorithm has to be fast and thus has to perform a minimum number of computations.

## 4.2 Theory

This section describes the theory behind the used map matching algorithm. In the literature review, chapter 1, several map matching algorithms have been described and finally Marchal's algorithm (2004) is chosen to use for this research. This algorithm was chosen because of the short computation time and good performance. Besides Marchal's algorithm, small adaptations on the algorithm are proposed. Furthermore, some theory about the deviation, centroid connection and parameters is presented.

### *4.2.1 Map matching algorithm*

Marchal (2004) developed a map matching algorithm that only used coordinates collected by GPS. The focus was to develop a fast algorithm for large volumes of data with reasonable matching errors. This is done by selecting the nearest links for each GPS point by an algorithm of White (2000). A path is created for each candidate link. For a new GPS point, the path will be extended by new links and taking care of the network topology. Finally, the most likely path is the one with the lowest deviation between the GPS coordinates and the coordinates of the path. The algorithm will be discussed in detail below.

**Distance between GPS point and link**

The variables below are used to determine the distance between a GPS point and a link.

$G(V, E)$ = Graph describing the network with $V$ is set of nodes and $E$ is the set of links;

$Q_i$= GPS point

$Q_{1...T}$ = Set of points given by the GPS data with $i = 1 \dots T$, the GPS points consist of a pair of coordinates $(x_i, y_i)$

$Q^t$ = projection of $Q$ at the link $AB$;

$AB$ = link between $A$ and $B$;

$d_e$ = Euclidean distance ("ordinary" distance between two points).

The distance is determined with:

$$d(Q, AB) = \begin{cases} d_e(Q, Q^t) \\ \min\left(d_e(Q, A), d_e(Q, B)\right) \end{cases} \begin{matrix} if\ Q^t \epsilon [AB] \\ elsewhere \end{matrix} \quad (1)$$

With the above definition, $d_{Q,AB} = d_{Q,BA}$ and a GPS point is equally distant from two opposite directed links of a given road segment. For this reason, the link is shifted perpendicular to the right, cause the driving side in the Netherlands. This shift is indicated with $\lambda$ and results in $A' = A + \lambda$ and $B' = B + \lambda$.

The figure below illustrated the distance between the GPS point and the link with the shift to the right. Important is the difference in distance measuring between $Q$ and $Q_1, Q_2$ as described in the function $d(Q, AB)$ above.

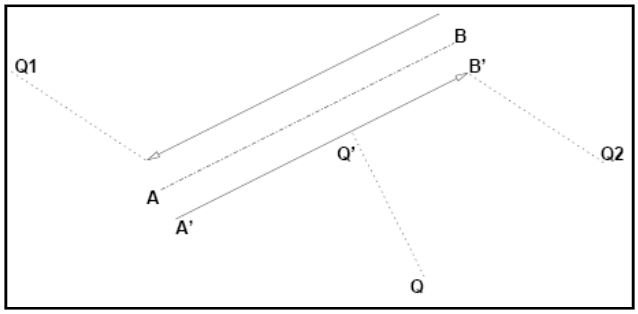



*Figure 11: distance between points $(Q, Q_1, Q_2)$ and shifted link A'B' (Marchal, 2004).*

**Score of path**

$P\{E_1, E_2, \dots, E_p\}$ is the path composed of the $p$ subsequent links $E_1, E_2, \dots, E_p$. Each path $P$ receives a score $F$. The absolute score $F$ is defined as:

$F = \sum_{j=1}^{p} \sum_{i=1}^{r} d(Q_i, E_j) \cdot \delta_{ij}$ (2)

($\delta_{ij} = 1$ if $Q_i$ is matched, $Q_{ij} = 0$ otherwise)

Now, all paths have a score and the problem is to find the path that minimizes $F$.

**Initialization**

The algorithm is initialized to find the set $S$ of the $N$ nearest links from the first GPS point $Q_1$. For more accuracy, a few more GPS points can be used for the initialization.

**Breaking routes into paths**

Through interruptions in the GPS data stream, a whole route could not matched perhaps by a single continuous path. Interruptions are detected with if-loops like the GPS signal is missing, distance between to GPS point is too large (e.g. 300m) or longer then a specific period (e.g. > 30 sec). The algorithm does not provide a solution to link the paths together, but a solution could be, using the Dijkstra's shortest route algorithm to link the paths.

**Following the network topology**

After linking $Q_1$ to the $N$ nearest link, the following procedure is applied with the next GPS point $Q_i$ for each of the path $P$ in the set $S$.

1. Assume that $Q_i$ is matched to the last link $E_p$ that is $\delta_{ip} = 1$;
2. Update the score of path $P$ using absolute score formula for $F$;
3. Insert $P$ in a new sorted set $V$;
4. Check if the route has reached the next intersection (i.e. the destination of $E_p$) by using the following condition.

   $\sum_{k=p_0}^{k=i-1} d_e(Q_k, Q_{k+1}) > \propto \cdot L(E_p)$ (3)

   with:

   $Q_{p_0}$ = the first point matched by $E_p$;

   $\propto$ = preventing problems that arise if a vehicle performs a U-turn in the middle of a long link (e.g. $\propto = 0.5$);

   $L(E_p)$= length of $E_p$.
5. If yes, create the new paths $P_k^{FS}$ that correspond to the forward star (links that are connected to the link) of $E_p$;
6. Update the score of paths $P_k^{FS}$ using absolute score formula for $F$;
7. Insert $P_k^{FS}$ in the new sorted set $V$.

Let $FS(E_p)$ be the set of downstream links from $E_p$ (forward star). If the condition (3) holds, a new path $P_k^{FS}$ is created for each of the link in $FS(E_p)$. These children paths differ from their ancestor since they have one more link: $E_{p+1}^k \epsilon\, FS(E_p)$.
where $E_{p+1}^k$ is the last link of path $P_k^{FS}$. The children paths are initialized with the last point so that $\delta_{ip} = 1$. This method prevents the choice of the wrong path if the GPS points are positioned in the middle of a Y-intersection.

**Maintaining a set of candidate links**
Children paths are inserted into the sorted set $V$ and ranked according to their score. When $Q_i$ is processed, $S$ is emptied and only the $N$ best elements of $V$ are inserted into $S$ to avoid that $S$ grows indefinitely.

When the last $Q_i$ is processed, the best path from $S$ is kept as the result of the map matching process. Similarly, when the route is broken and a new path had to be started. Figure 12 shows the flow chart of the algorithm.

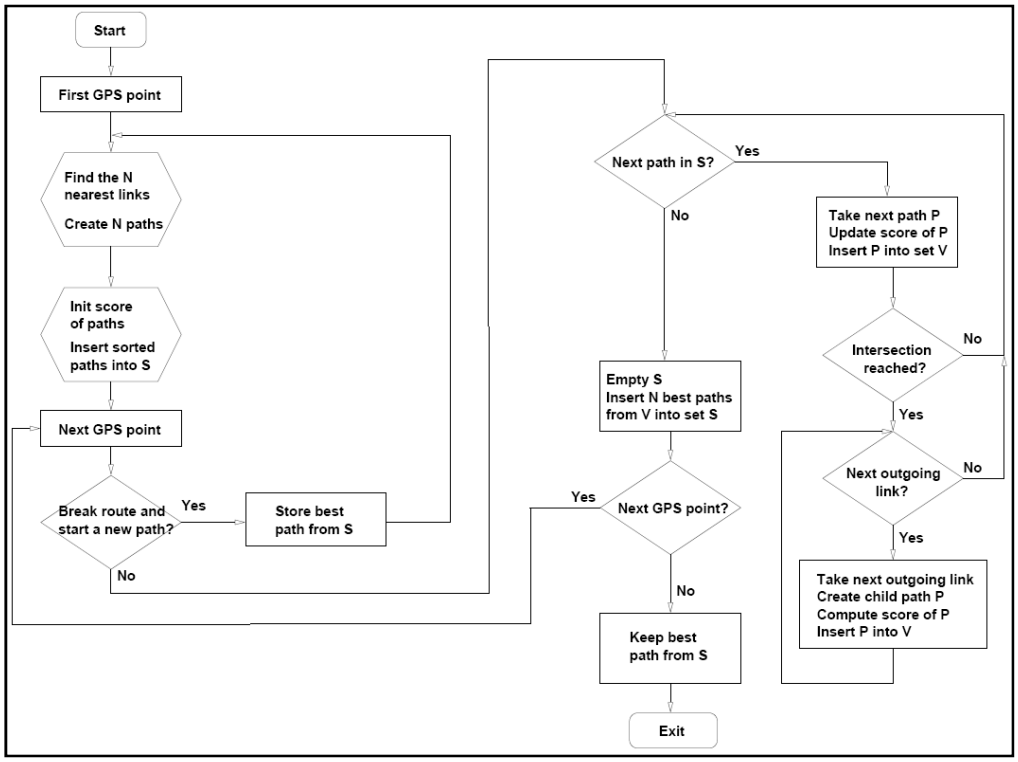



*Figure 12: Flow chart of map matching algorithm (Marchal, 2004).*

Summarizing, the methodology of the algorithm comes down to the following. At first, several unique paths will be generated that all include a link near the first GPS point. After this, all GPS points will be selected sequent and checked with each path. The main idea of the algorithm is to increase the score of the path, which includes the nearest link in comparison of the selected GPS point. The next link is determined if the distance between the GPS point, matched to the last intersection, and the selected GPS point is more than the length of the last link in the path. The accessory path is copied to new unique paths, which all receive a different topological connected link at the arrived intersection. The old path is deleted. The maximum number of paths is fixed and before selecting the next GPS point, the surplus paths with the lowest scores are removed. Finally, the last GPS point is selected and the path with the highest score is chosen. This path is stored as the chosen route.

### 4.2.2 Adaptations

The original algorithm of Marchal shifts a link by using the parameter $\lambda$, because the distance of a GPS point is otherwise equal from the two opposite links. The analysis of the GPS data in section 3.1.4 shows that especially in Den Hague, the GPS receiver had a deviation of approximately 40 meters. Therefore, a GPS point with a deviation in direction of the opposite links (incorrect side of the road) will never be connected to the correct direction, because the opposite shifted link is always closer.

Therefore, each GPS point will be matched to a two-direction link and direction is determined by the topological connection of the links to each other. There is always only direction available, namely the previous link. Therefore the $\lambda$-parameter will not be used.

### 4.2.3 Route connection with centroids

All map matched routes do have to start at an origin centroid and end at a destination centroid, because all travel data (e.g. attraction of a centroid) is stored for centroids. The actual arrival of a vehicle is probably not at a centroid and therefore the route has to be connected to a centroid. A simple method is used to connect a route with the centroids. This method finds the closest centroid and extends the current route with the links of the shortest path to that centroid.

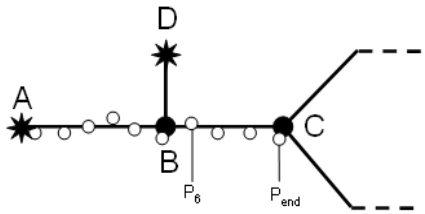



*Figure 14a: GPS points plotted in the network*

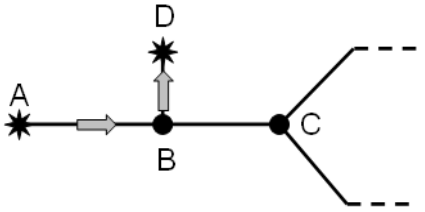



*Figure 14b: map matched route*

At first, the nearest centroids are selected for the last link's end node in the route. Now, the distance is calculated between the end node and each selected centroid and between the penultimate node and each selected centroid. The penultimate node is also taken into account, because the vehicle does not have to drive over the entire last link. The centroid that has the shortest to the end or penultimate node will be selected as final centroid of the route. In case the penultimate node resulted in the closest link, the last link of the route has to be removed to prevent a surplus link.

An example is shown in Figure 4, the route starts in centroid A and ends at node C. Another centroid is also available, which is D. The map matching algorithm selects link A-B and B-C to be the map matched route. This means that the route starts at a centroid, which is correct, but the route does not end at a centroid. Therefore, the nearest centroid is selected compared to penultimate node B and node C. The centroid D is the closest centroid and the map matched route has to be connected to this centroid. Therefore, the route end is connected to centroid D, and the link B-C has to be removed, because otherwise the route would be incorrect (A-B, B-C, B-D).

This described method could also be applied on the beginning of the map matched route is needed. It is a simple method but could result in illogical routes, because the shortest path to a centroid does not have to be the most logical path when looking to the entire route. A better method should be developed as Marchal (2004) also suggested, however this will not be part of this research.

### 4.2.4 Deviation

In order to check the accuracy of the map matching algorithm, we will use a deviation measure that expresses the difference between the GPS points and the map matched route.

This deviation is determined for many selected GPS points equally divided along the driven route. The ratio of selected GPS points in urban areas will be higher in comparison with the rural roads, because the mean speed is lower here. Besides this, the mean link length is much shorter in an urban area, which results in a fair relation of checked links between rural and urban roads.

The distance between each GPS point and the map matched route is calculated in meters. The mean distance is the deviation of the route. Because of incorrectness while connecting the route with the centroids, the GPS points at the start and end of the route are not taken into account while calculating the deviation. This example is shown in Figure 14 with centroids *A* and *D*. The GPS points will be map matched to link *A*-*B* and *B*-*C*. In the traffic model, the route cannot end in node *C*; the route has to end at a centroid. Centroid D is selected and the end of the will be changes, link *B*-*C* is removed and link *B*-*D* is added, which results in an incorrect deviation if GPS points $P_6$ till $P_{end}$ are used, because the nearest link is removed from the route.

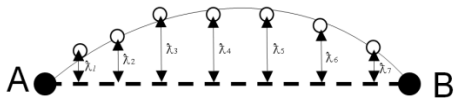


*Figure 15: map matching of a curved link*

It is important to notice that the deviation of some routes is high, but this is practically incorrect. There are two reasons for this fact. The first reason is the shape of links in the network. The distance between the selected GPS point and the nearest link is measured by a straight line. This results in a difficulty, because some links has a curve shape. Because the straight distance is used, the GPS points are further positioned in comparison with the straight link between the links A and B. This causes a rise of the deviation value. Figure 15 shows a link between node *A* and *B*. The GPS points are positioned along this line and despite their correct position, the deviation will be high, because the deviation is calculated concerning the straight line between node *A* and *B*.

Another important reason is the network deviation in the network model. This network does not include all roads and especially in the urban areas, the map matched route will deviate from the driven route that results in a higher deviation. Both problems result in the fact that routes with a higher deviation (e.g. 100 meters) could be correct.

### 4.2.5 Parameters

The algorithm uses several parameters, which have influence on the performance of the map matching algorithm. The two most important values are discussed first, the other parameters after this.

**α-parameter**

The α-parameter is included in the condition that determines if the vehicle has reached the end of the link. The value of α determines the percentage of the link length at which the algorithm starts to find new topological

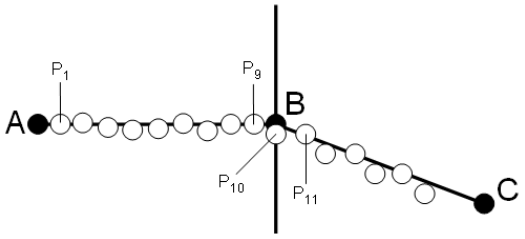


*Figure 16: effect of the α-parameter*

connected links. The old path is copied to new paths and a unique topological connected link is added. Figure 16 shows the influence of the α-parameter. The algorithm selected link *A-B* and is now checking the GPS points till the length of link *A-B* is smaller than the distance between node *A* and the selected GPS point. The GPS points indicated with $P_1$ till $P_{10}$ are equally divided over link *A-B*. This means that $P_1$ is positioned at 10% of the link length and $P_{10}$ at 100%. If α is 100%, the algorithm will find new links at $P_{10}$. In case of α is 90%, the new links will be found at $P_9$.

In Marchal's algorithm, α is set to 0.5, because of u-turns in the middle of a link. A low value for the α-parameter results in the fact that new links will be chosen before the GPS points "arrive" at the end of the link.

**Path size**
The algorithm has a fixed number of paths that can be stored. This amount has influence on the computation time and on the performance. An increase in path size will result in a higher quality, but also a higher computation time. In an exceptional case the deviation might increase for a specific route. This is a result of an extra incorrect path with a high score accidentally, which generates new paths and overwrites the correct path. Normally, the deviation will decrease if more paths are kept in account.

**Other parameters**
The last sections describe several additional parameters that are used by the map matching algorithm. The table below sums these parameters and supplies a small description.

| Parameter name | Description |
|---|---|
| Map matching | |
| Starting number of links | The nearest links are selected concerning the first GPS point to generate the same number of start paths |
| Starting number of centroids | The nearest centroids are selected to connect the route with the origin and destination |
| Remove filters | |
| Route detour | If the map matched route has a large detour in comparison with the shortest route, the route will be removed |
| Minimum route length | Map matched routes with too few links are removed |
| Quality criteria | |
| Number of checkpoints | Several GPS points are selected to determine the deviation between the GPS points and the map matched route |
| Skipped quality points | The first and last groups of GPS points are not used to determine the deviation of the map matched route because of the route connection discussed in section 4.2.4 |
| Maximum deviation for correct routes | All map matched routes with an acceptable deviation are accepted in the final route set and used for further research. |

*Table 4: the other parameters used by the map matching algorithm*

## 4.3 Approach

The purpose of this approach is to map match the GPS data correctly with a low deviation. In order to perform this job, the parameters of the map match algorithm have to be calibrated. The first experiment in the case study is to calibrate these parameters by

using a part of the entire GPS data set. The parameter set will be judged on quality of the map matched route and on the computation time. It is not efficient to use the entire GPS data set, because the map matching will take much time. The second part of the case study is to perform the calibrated algorithm on the entire GPS data set. After this, the map matched routes can be analyzed and assessed on the deviation to use them for the route set generation.

Initially, a small second case study had the purpose to calibrate the map match algorithm for another GPS data set and network. Unfortunately, this was not successful because of memory issues.

Finally, the observed routes are assessed and a large part of route is appropriated for the route set generation. Several map matched routes are fine tuned, because of small errors in the map matching. These adapted routes are also added to the accepted routes.

## 4.4 Case study

The map matching algorithm is used to map match the "Spitsmijden" GPS data to the network model. About 9.5 million GPS points were collected. All information about the GPS data is represented in section 3.1. These data will be map matched to the Haaglanden network. This network has a quite high resolution and includes 2084 mostly two-way links and 1295 nodes.

The algorithm was coded in Ruby, the program language used in OmniTRANS jobs. A heavy computer is used to run the algorithm with a dual quad core processor with 3.2 GHz and 32 GB RAM memory. Although, OmniTRANS runs this algorithm only on one core (with a maximum of 4 GB RAM memory).

### *4.4.1 Used parameters*

**α-parameter and maximum path size**
These two parameters are variable. The maximum path size is set to 5, 10, 15 and 20. Higher path sizes result in too much computation time. The α-parameter is set to 0.7, 0.8, 0.9, 1.0 and 1.1. As discussed in section 4.2.5, low values might result in incorrect results, but because Marchal uses a low value so we are interested in these results, because we want to perform a constricted number of computations, our lowest value is 0.7 and not 0.5.

**Map matching**
The start number of links is set to the nearest five links, because all plausible links are likely to be positioned within this area. The five nearest centroids are selected for the centroid connection, because the most acceptable centroid will likely be within this area.

**Remove filters**
Routes shorter than 4km are not taken into account, these trips are not representative for the route set calibration. The influence of the centroid linking to the start and end GPS position is too large for these routes. Secondly, routes that are two times longer than the shortest route will be removed because it is not likely that drivers will make such a detour. These excluded routes are investigated and include one or more short stops.

**Deviation**
The GPS points used to determine the deviation are without 100 GPS points at the end and start of the route, because of the centroid linking as discussed in section 4.2.2. The deviation self is determined with 100 GPS points equally selected over the remaining GPS points. Finally, the matched routes with a deviation of 150m or lower are accepted in the final route set and used for further research. This deviation seems to be high but because of the network resolution and the incorrectness with curved links, both described in 4.2.4, the routes with this deviation should be correct.

The table below shows the parameter values used for this case study.

| Parameter name | Value |
|---|---|
| α-parameter | 0.7,0.8,0.9,1.0,1.1 |
| maximum path size | 5, 10, 15, 20 |
| Map matching | |
| Starting number of links | 5 |
| Starting number of centroids | 5 |
| Remove criteria | |
| Route detour | 2 |
| Minimum route length | 4km |
| Quality criteria | |
| Number of checkpoints | 100 |
| Skipped GPS points | 2x100 |
| Maximum deviation for correct routes | 150m |

*Table 5: parameters with the used value and description*

## 4.5 Results

### *4.5.1 Computational speed*

A first experiment is performed on 2.5% of the GPS data (approx. 250.000 GPS points). The algorithm generates 71 routes, which is independent of the used parameters. The computational speed is determined with a α-parameter of 1 and maximum path sizes of 5, 10, 15 and 20. The α-parameter does have influence the runtime, as the number of calculations is the same. The difference in runtime is caused by the path size. Results reported in Table 6 show the runtimes.

| Path size | 5 | 10 | 15 | 20 |
|---|---|---|---|---|
| Runtime (s) | 275 | 425 | 575 | 730 |
| GPS points per second | 909 | 588 | 435 | 343 |

*Table 6: runtimes for different path sizes to map match 250.000 GPS points*

Fortunately, the runtimes are linear, which indicates a well structured algorithm without saving surplus data into the RAM memory. The figure shows the linear curve of the runtimes. The runtimes are not comparable with Marchal's algorithm, because of differences like program language, network resolution, parameter settings and used computer.

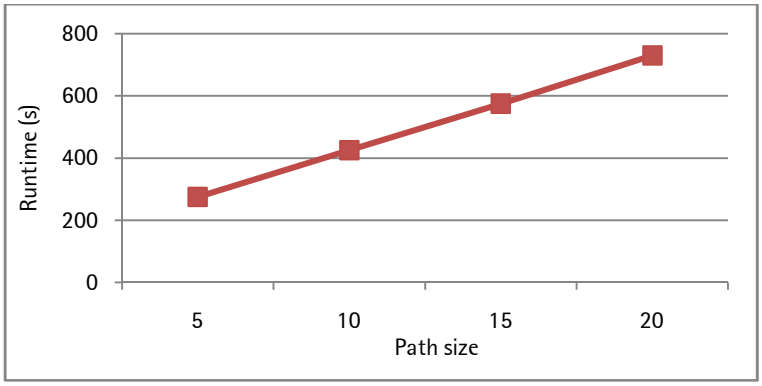



*Figure 17: runtimes for different path sizes to map match 250.000 GPS points*

### 4.5.2 Calibration

The optimal parameter combination is defined as the set that results in the lowest deviation. These parameter settings are investigated with a part of the entire data because of long computation times. A final experiment is performed with the chosen parameters and executed on all data.

**First experiment**

The important adaptable parameters are the α-parameter and the maximum path size. The map matching algorithm is performed with different combinations of these two parameters resulting in a mean deviation. Table 7 shows the results.

| Path size | 5 | 10 | 15 | 20 |
|---|---|---|---|---|
| α = 0.7 | 193 | 101 | 71 | 63 |
| α = 0.8 | 131 | 114 | 46 | 40 |
| α = 0.9 | 97 | 83 | 44* | 44 |
| α = 1.0 | 111 | 67* | 67 | 56 |
| α = 1.1 | 192 | 128 | 107 | 104 |

** optimal combinations of the α-parameter and the maximum path size*

*Table 7: the mean deviation (meters) by map matching 250.000 GPS points without exceptional routes*

The above results do not include routes that cannot be map matched correctly. The origin and destination of these routes were located inside the study area, but the chosen route was outside the network. Especially, the chosen routes between Wassenaar and Zoetermeer were not possible to map match correctly. This can be seen in Figure 18; there is no eastwards road presented from Wassenaar to Zoetermeer while this road does exist. These kinds of routes are deleted of the data set and results in more realistic deviations. Besides these routes, some other routes cannot be map matched because of route specific characteristics (e.g. the route cannot be connected to a centroid).

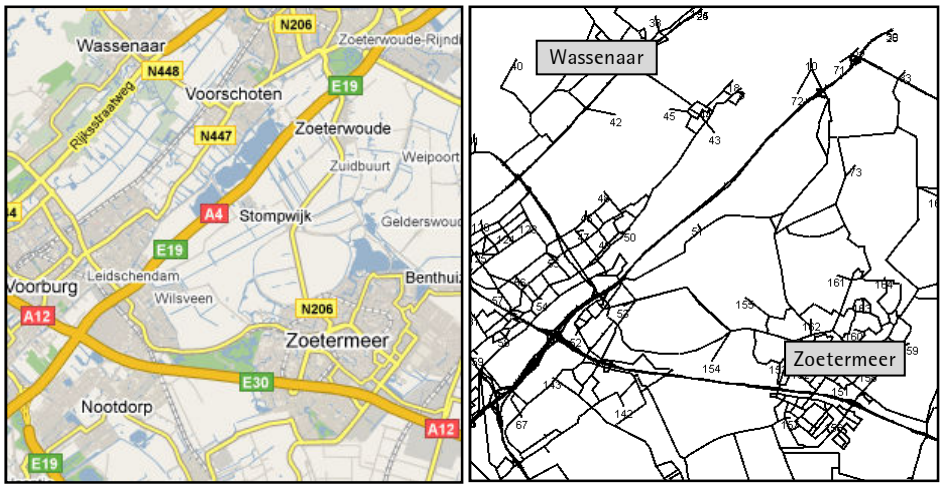



*Figure 18: the eastwards route between Wassenaar and Zoetermeer is missing in the network model*

The lowest deviation is obtained with a α-parameter of 0.9 and a path size of 15 or 20. Because the runtime of the first parameter combination is lower than the second combination, a α-parameter of 0.9 and a path size of 15 are chosen as optimal parameter set. The deviation difference between this combination and a path size of 10 and a α-parameter of 1.0 makes it worth to have a longer runtime.

The map matching algorithm cannot match all routes correctly because of several reasons, which will be discussed later. The optimal parameter set map matches 71% of all available routes (groups of GPS points). This number will increase by combining two parameter sets. A combination of parameter set 1 and 2 (as marked in Table 7) result in a match of 84% of all routes. The deviation will increase to 45 instead of 44, but more routes are matched. The runtime will increase with 174%, but for this research, this extra runtime is of little concern. In case of other map matching processes in future, this choice has to be reconsidered because of the longer runtime.

The α-parameter used by Marchal (2004) results is not the optimal value as shown in the previous table. A low value for the α-parameter results in the fact that vehicle is virtually too far behind the already map matched links. This could result in incorrect map matching results, because the score is still determined in relation to the position of the car. This problem is described extensively in Appendix C.

### *4.5.3 Deviations*

The second experiment consists of executing the algorithm on all GPS data (9.5*M* GPS points) with the two parameter setting combinations found during the performance of the test experiment. These parameter combinations have a α-parameter of 0.9 respectively 1.0 and a path size of 15 respectively 10. The map matched route with the lowest deviation is selected for further research. The table below shows the distribution of the route deviations.

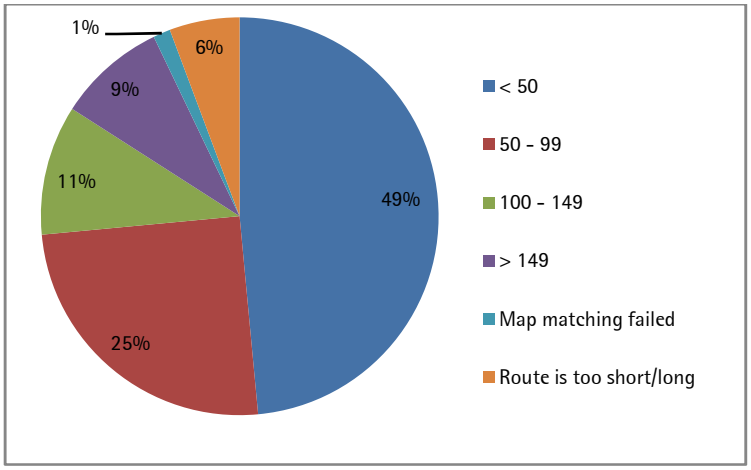



*Figure 19: the distribution of route deviations*

The figure above shows that 85%, 2505 routes, of all routes are map matched with a deviation of less than 150 meters. There routes are useable for further research. Furthermore 6%, 170 routes, are still too short or too long and are not appropriated for the map matching. So in fact, 89% of all routes are map matched correctly, because it is not fair to take the too short and too long routes into account when calculating this percentage. The excluded 10%, 305, routes can be divided into two categories, the deviation was too high (9%, 263 routes) or it was not possible to map match the route correctly (1%, 42 routes) because the route cannot be connected to a centroid. A large part of the routes with a high deviation is caused by the missing road between Wassenaar and Zoetermeer as already noticed in the test experiment.

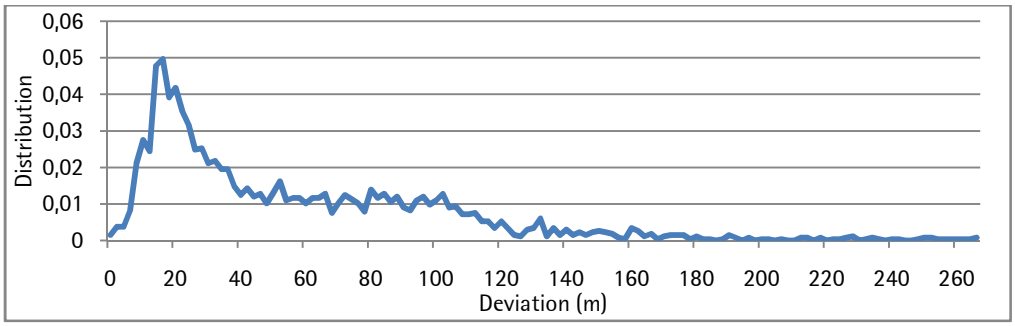



*Figure 20: the distribution of route deviations*

The figure above shows the distribution of the deviations in more detail. The deviation range between 50 and 120 meters is remarkable and is therefore investigated. We found out that the causes of higher deviation are already indicated in section 4.2.4. At first, the deviation is not calculated correctly for links with a curved shape and besides this, the network has a deviation in comparison with the actual road position.

## 4.6 Alternative case study

**Data**

Besides the GPS data of the "Spitsmijden" project, another GPS data set is available. This set consists of GPS points recorded every 10 seconds in contrast to every single second like the first dataset. It is interesting to map match these routes because of this difference. Furthermore, the data set includes routes in the entire country of the

Netherlands, which has to be map matched on a large OmniTRANS network with more than 234.000 links and 4.200 centroids.

**Approach**
It is tried to perform the map matching algorithm on this dataset to have a better view of the algorithm performance. Unfortunately, the map matching algorithm could not store all the network data in the primary storage. Because of this problem, the GPS data is map matched in the Haaglanden network.

**Evaluation**
The most map matched routes in the second data set do not satisfy the location criteria, they are located outside the 'Haaglanden region', but 16 routes do satisfy. The results of the map matching are good; all the routes are map matched with a deviation lower than 100 meters by using a α-parameter of 0.9 and a path size of 15.

**Results**
The performance of the algorithm is at least the same in comparison with the first GPS data set. This is easy accountable, because most GPS points are unnecessary, they are only used to increase the score of the best path. This is interesting, because the map matching could be faster when fewer GPS points are taken into account.

## 4.7 Fine tuning of the map matched routes

As concluded in section 4.5.3 the deviation of the map matched routes is low, but this does not mean that all map matched routes are correct. This section describes the fine tuning of the map matched routes to be sure that these routes are correct for the further research.

A first problem is the fact that the deviation does not indicate an incorrect map matching at an off-ramp and on-ramp of a highway. An example of this is shown in Figure 21, the GPS points are positioned between the off-ramp link and the highway link. With the speed of the GPS points it is possible to determine that the vehicle stayed on the highway, but the algorithm chose the off-ramp link. The incorrect links are selected, because the GPS points are closer to the off-ramp than the highway link.

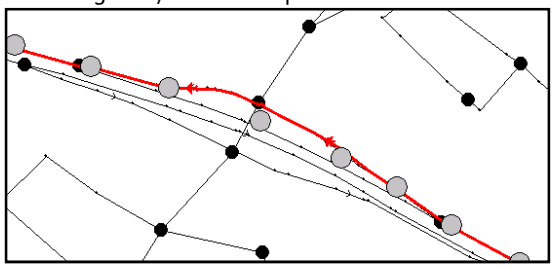


*Figure 21: the situation has a low deviation but results in an incorrect map matched route*

A second problem exists through the users that make a detour for a short stop like dropping the kids at school. Figure 22 shows this problem, the driver's route is marked by the black points. The route actually consists of two trips, because the driver drops his kids at school; he changes his route because of this. At the school, the driver stops for only one minute. Besides this, the driver also stops a few times for one minute at a junction.

It is quite difficult to determine if a route consists of only one trip. For a human eye it's easy to observe that the black route is quite strange and the black/green route should be more logical. Because of this, the 1 minute-criterion will not find short stops like mentioned in the example. Unfortunately, this time cannot be set lower, because

otherwise all routes with long stops at traffic lights will be incorrectly separated in two trips.

*Figure 22: the black route actually consists of two trips because of short stop at the school*

**Theory**

This section describes two checks performed on the map matched routes to be sure that all map matched routes are correct.

At first, a route set generation performed for a few iterations will generate only some logical routes between the origin and destination. We assume that in case a map matched route is nearly equal to one of these generated routs, the map matched route is correct, because it is included in a small route set of routes and outstanding routes will not be included in those generated route sets.

This route generation is performed for only 10 iterations without any detour or overlap parameters. An average of 7 routes will be generated between each selected OD-pair. If the map matched route has an overlap of at least 90% in distance in comparison with one of the routes in the route set, the map matched route will be accepted as correct. The last 10% is such a small deviation that it will also be accepted in the comparison of the observed routes with the generated routes.

The second check is only performed on the map matched routes that are not included in the generated route sets. These routes will be checked visually. A small script is developed to supports this visual check. When the script is started, the next map matched route is shown on the OmniTRANS network and a popup appears with an "accepted", "change" and "unaccepted" button. A route is accepted when no strange detours, because a short stop is made, are visible. Routes with a small error are changed (e.g. the off-ramp and on-ramp problem as discussed before). The routes with a strange detour because of a short stop are removed. After selecting one of the buttons, the choice is stored and the next route will be plotted in the network. All accepted and changed routes are added to the final data set. The script is further discussed in more detail in Appendix B.

**Application**

In total, 2505 routes are map matched as described in section 4.3.3. The first test results in 1082 routes that are included in the generated route sets. All other routes are checked visually, which results in 1054 additional correct routes. The other 369 routes are removed. This results in a total of 2136 routes that will be used in the remaining research.

It is interesting to describe the removed routes, because for several cases the map matching algorithm does not detect or does not generate the routes correctly. Taking the produced errors in account could improve the map matching algorithm in the future. There are four main errors that cause incorrectness.

1. Approximately 80% of the removed routes are not correctly matched to the centroids. The added links from the map matched route's destination till the final centroid result in a detour in contrast with the shortest route. This mismatch makes it more difficult to get a 100% match between the map matched route and the observed routes. The incorrect matched links have to be accepted to

prevent false results (e.g. illogical routes to the centroid). This should result in a similarity of approximately 90% between a map matched route in comparison with a route in the StreamLine route set. The figures below show an example of centroid connection resulting in an incorrect route.


*Figure 23: plotted GPS points - map matched route - connection to wrong centroid – more logical route*

2. Problems with the short stops are already expected as mentioned earlier in this section. The short stops are not always detected, because the deviation of the GPS receiver. If a vehicle parks for five minutes and the GPS remains on, the coordinates could still change. A better solution could be to check the change in speed; this change will be 0 or 1 km/h, because the deviation is only a few meters. A maximum speed of 1km/h during a minute could indicate a stop.

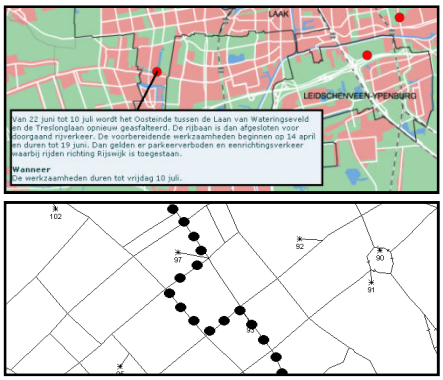



*Figure 24: consequently detours in a route could be caused by road works. Some detour locations within The Hague are indicated as road work locations.*

3. In some cases, the vehicle makes a strange detour because of road works. The closing of a highway off-ramp could lead to large difference between the Map matched and the StreamLine route set. These routes are removed after they are detected in road works messages from the city The Hague. Small road blocks like a garbage truck are not interesting, because this mostly happens in a residential area, which is not taken into account because of the network resolution. Otherwise, the vehicle will make a small detour, which is only a small deviation in comparison with the logical route. Figure 24 shows the website of the city The Hague that supplies information about all road works. The strange detour in the figure is explained by the road works.
4. The map match algorithm matched some routes incorrect at highway ramps. The vehicle is matched to the ramp links, but drives also 120km/h that really indicates that the vehicle was driving at the highway. A speed check could prevent this problem. Although, these small errors will be not result in a deviation of more than 10%.

## 4.8 Conclusions

The algorithm developed by Marchal (2004) is successfully implemented in OmniTRANS and produces routes that can be used for many purposes in accordance with the

OmniTRANS network. The implemented algorithm is able to map match in an acceptable runtime; the computational time is 450 times faster than the collection time (450 GPS points/s). Approximately 89% of the routes have a deviation lower than 150 meters, which make them useful for other research. The efficiency of the algorithm depends on the two most important parameters; this research found out that the α-parameter of 0.9 and the path size of 15 are optimal for this network. The parameters used by Marchal are not optimal, because they could result in an incorrect score determination when the GPS point are positioned to far behind the matched links (for more information see Appendix C). The deviations represented in Table 7 prove the deviation difference between the used parameters. As discussed in the previous section, several improvements can be performed to achieve better results. The available time was limited to prevent these limitations of the algorithm.

The usefulness of the algorithm is high when looking ahead to the future where GPS data is more and more available and much research can be performed with this data. The validation of travel times calculated by traffic models, the calibration of junction modeling, the route and mode choices of drivers are just a few examples of the many purposes.

## 4.9 Limitations and recommendations

The implemented map matching algorithm has a few limitations. As suggestion for improvement, the following items are interesting:

1. Use the formpoints to determine the distance between a GPS point and a link instead of splitting the curved links to separated links;
2. Upgrade the way of determining the deviation by reversing the check; select an number of formpoints along the links of the map matched route and determine the distance to the nearest GPS point. This prevents the problem of high deviations because of curved links and the removed links because of centroid connection;
3. Improve the method to determine the deviation, because incorrect links near the correct links are not determined well. In case of highway off-ramps and on-ramps close to the highway itself; the speed could be used to determine the correct links.
4. Create an object in OmniTRANS to use the map matching function more easily;
5. Change the usage of network objects to use less primary memory as recognized problem in section 4.6.
6. Compare the deviation of map matched routes between one GPS data recorded every second with the same map matched routes, but now aggregated to a record of every 10 seconds to realize a shorter runtime.
7. Improve the centroid selection at the start and end of a route, because incorrect routes could be created. Marchal (2004) also recognized this problem, but did not supply a solution. A simple solution could be to replace the last *N* links with the shortest route. A more comprehensive solution could be to replace the last links with an attribute of 30km/h or 50km/h for the shortest route if the policy maker is not interested in the route choices on these roads. In this case, the main roads in the GPS route are still intact and have to be match exactly to the routes in the route set, but the unnecessary deviation will not influence the matching results. Another solution could be to assign areas to each centroid. The vehicle's destination is located in an area that is connected to a centroid. If the vehicle

arrives at that area, the route is extended with the shortest route to the destination centroid. The reverse approach could be used for the origin centroid.

8. Improve the calculation speed by taking different road types into account. In case of a highway that is matched correctly for many GPS points, it's a pity that nine other paths are still taken into account while it is certain that these paths are incorrect.
9. Investigate the 10% of the routes, which have a deviation of more than 150 meters. In case of a general cause, the algorithm could be improved.

# 5 Route set generation

*In section 5.1 the route set calibration will be introduced. Section 5.2 will give a comprehensive description of the route set generation by discussing randomization, alternative route generation and filtering of routes.*

## 5.1 Introduction

Travelers have a choice between several routes to their destination. These routes may differ in distance, travel time or another specific attribute like the number of signalized intersections. The costs of a route are indicated with one (or more) of these attributes. A rational driver is considered to choose the route that has the lowest costs for him.

The driver's route choice is modeled in the route choice model. Before the actual choice can be determined, the available routes have to be known. Only the routes within the route set have potential of being chosen. Therefore, a good route set needs to be generated.

As discussed in the literature review, several route set generators have been developed. The Monte Carlo method will be used in this research, because the optimal parameter settings of this method, as implemented in StreamLine, are unknown. The optimal settings have to make sure that the route sets include all relevant routes that may be chosen and not too many irrelevant routes.

The calibration of these parameters is done with the observed routes (the map matched routes) described in chapter 4. The goal is to find parameter values that generate route sets that include as many map matched routes as possible, but also take the average number of routes per route set into account and has relatively few excess routes.

*Figure 25: the main roads and highways in the Haaglanden area*

It is important to notice that the observed route parts on main roads (highways and provincial roads) have more priority to be matched correctly than the local roads within an urban area. The main roads have much more influence on the traffic situation and therefore have to be

included correctly in the route set. Contrary, the drivers' route choice in an urban area depends on local circumstances (e.g. unloading truck, small-scale road works) that are difficult to match correctly with the routes in the route sets.

### 5.1.1 Definitions

In order to understand this chapter clearly, the used definitions and point of views are described in this section.

There are two kinds of routes used for the route set calibration. The observed routes are the originally map matched routes from the GPS data and are part of the observed route set. These routes present the rational chosen routes and have been checked on correctness in the previous chapter. For this, the routes are relevant that means that each route is taken by at least one person in the respondent group. The StreamLine route generation model routes are the generated routes and constitute the generated route set. These route sets have to include as many as possible observed routes.

Furthermore, routes can be identified in three ways. The cheapest route is the route with the lowest costs. The shortest route has the shortest length and the fastest route has the lowest free flow travel time.

## 5.2 Theory

This section discusses the theory behind the StreamLine route set generator. The operating is based on the accelerated Monte Carlo method. The method assumes that the first shortest route is relevant for the route set. For each iteration link costs are adapted, which might result in a new shortest route. This new shortest route will be compared with the other already accepted alternative routes. This comparison is performed by using several filters like the detour and overlap of the new route.

Each link has cost defined as $C_a$. This cost will be randomized by using a cumulative distribution function (CDF) of Gamma, whereby for each iteration $i$ the costs of the link will be adapted. The costs are adapted by multiplying the link costs with a factor $X_a^{(i)}$ abstracted of Gamma's CDF. The factor depends for each the iteration and for each link. The method is mathematically described as:

$C_a^{(i)} = C_a \cdot X_a^{(i)}$

with:

$X^{(i)} = \Gamma(\frac{1}{\theta^{(i)}}, \theta^{(i)})$

for:

$i = 1, \dots, N$

$N$ = maximum iterations

The gamma distribution is described as $X \sim \Gamma(k, \theta)$ whereby the $\theta$ is different for each iteration, so $\theta^{(i)}$. The $\theta^{(i)}$ depends on the number of times $\alpha$ that the variance is increased. Each time that a so-called threshold is reached, the value of $\theta^{(i)}$ increased with $\theta^{grow}$. This can be described as:

$\theta^{(i)} = \alpha^{(i)} \cdot \theta^{grow}$

This section discusses the randomization process first. After this, the alternative route generation is explained, followed by route filtering and the route set generation stopping criteria.

### 5.2.1 *Randomization*

In order to randomize the link costs a cumulative distribution function (CDF) is needed. This distribution function describes the probability distribution of a real-valued random variable. The randomizing process selects a random value for the probability value to determine the belonging variable. The mean value must be 1, because these values present the factor $X_a^{(i)}$, which is multiplied with the link costs $C_a$.

The literature is not specified about which distribution function should be chosen for the optimal route set generation with the Monte Carlo method. There are two requirements for the distribution function:

1. The distribution does not have negative values, because this could result in negative travel times, which is not desirable. Kant (2008) found that a gamma distribution has preference instead of the uniform distribution with the simple randomizer, because of this.
2. The variance must be changeable without changing the mean value of the distribution. An exponential distribution cannot be used for example, because only the variance in this function influences the mean value and the variance.

Both requirements will be amplify more. Figure 26 shows the CDF with a mean value of 1 of a Gamma distribution. A random value of 0.4 is selected on the y-axis resulting in a value of 0.92 on the x-axis. This value of 0.92 is the $X_a^{(i)}$ parameter by which the link costs will be multiplied. In this example, the link costs of this individual link decreases. As said in the requirements, it is desired to receive only positive link costs and therefore the CDF must not consist of negative values. This procedure is applied for each link in the network.

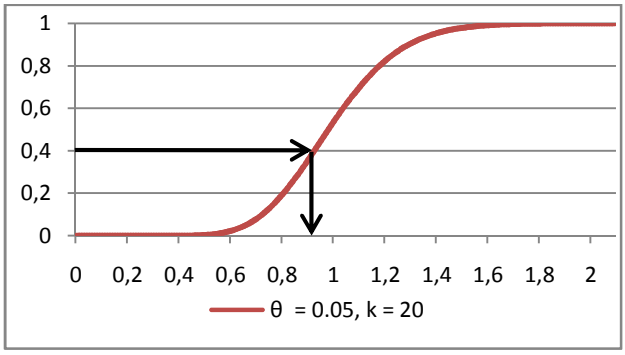



*Figure 26: gamma cumulative distribution function with the determination of the factor by which the link costs are multiplied*

An accelerated Monte Carlo approach is used in the route set generator, which means that the variance must be adaptable. Therefore, the needed distribution function has to depend on two variables, which makes it possible to have different graphs with the same mean value. This is an interesting case, because the slope of the CDF can now be influenced by which also the impact on the link cost can be influenced. A less steep graph has a larger possible change in link costs than a steep graph.

**Distributions**

Many distribution functions are available and have their own characteristics. The table below shows several distributions with their characteristics. The next abbreviations are used in the table:

$c$ = costs (e.g. travel time or distance)

$v$ = variation (the variation increases each iteration with an adjusted value)

$r$ = random

$mv$ = maximum variation = $0.5 \cdot \sqrt{12 \cdot v}$

| Randomiser | Distribution | Mean value | Random value | Alternative costs |
|---|---|---|---|---|
| Simple | Uniform | 0 | -1 < random < 1 | $c \cdot (1 + v \cdot r)$ |
| Uniform | Uniform | $1 + \left(\frac{mv}{2}\right)$ | 1 < random < 1 + $mv$ | $c \cdot r$ |
| Gamma | Gamma | 1 | 0 < random < $\infty$ | $c \cdot r$ |

By taking the two requirements into account, the Gamma distribution is confirmed to be the best option to use for the randomization. The simple randomizer could result in negative values, which is undesired. The uniform method cannot increase the variance such that there is a larger probability that link costs change.

**Gamma distribution**

The Gamma distribution depends on two variables, the scale $\theta$ and shape $k$. A random variable $X$ that is gamma-distributed with these variables is denoted like:

$X \sim \Gamma(k, \theta)$

with:

$E(X) = k\theta$

$Var(X) = k\theta^2$

The probability density function and the cumulative distribution function are written like:

$$\Pr(X = x) = x^{k-1} \frac{\exp\left(-\frac{x}{\theta}\right)}{\Gamma(k)\theta^k} \quad (4)$$

and

$$\Pr(X \leq x) = \frac{\gamma(k, \frac{x}{\theta})}{\Gamma(k)} \quad (5)$$

With:

$\gamma(k, \frac{x}{\theta})$ = lower incomplete gamma function. The incomplete gamma function is defined as an integral function of the same integrand. The lower incomplete gamma function can vary the upper limit of integration and is defined as $\gamma(s, x) = \int_0^x t^{s-1} e^{-t} dt$.

The mean value for the gamma distribution has to be *1* because the probability to increase or decrease the link costs is equally distributed. Because the mean value is set to 1, the variance is equal to the $\theta$-value, this is proved by taking $k = \frac{1}{\theta}$, such that the mean value of $\Gamma\left(\frac{1}{\theta}, \theta\right)$ is 1 and the variance is $\theta$. So $Var(X) = \theta$.

The cumulative distribution function of gamma is used to calculate the $\theta$-value that is multiplied by the link costs. The spread of these values depends on variance. An increase

in variance results in a declining gradient of the cumulative distribution function. This results in a greater spread of the random values as shown in the figure below.

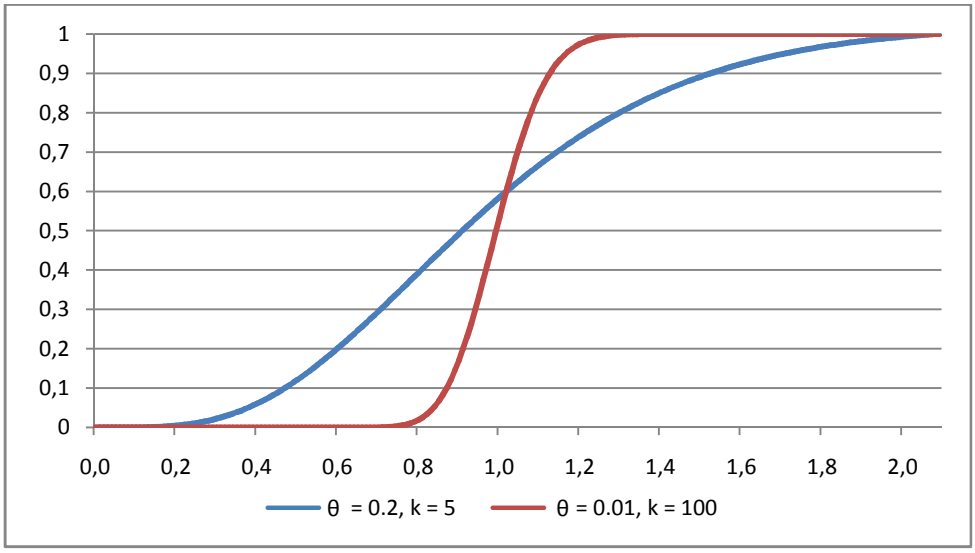



*Figure 27: gamma cumulative distribution function*

The gamma distribution with $\theta$ of 0.01, equal to the variance, results in a spread of 0.25 for the random values in 80% (0.1 – 0.9) of the cases. The graph with a $\theta$-value of 0.2 has a much greater spread, namely 1.05 for 80% of the cases.

### 5.2.2 Randomizers

The route set generator has four different types to generate alternative. The types are 'one to one', 'one to all', 'all to one' and 'all to all', which do not have to mixed up with the four generalizations of the shortest path generation mentioned in section 2.2.2.

The 'one to one' method randomizes the network for each new route per single OD-pair. This makes the type relative slow, but it also uses less memory, because each route set is processed before starting with the next route set. The other types randomize the link costs for all origins (all to one), destinations (one to all) or both (all to all) make calculating faster, but all routes have to be stored in the memory to compare the routes with the already accepted routes. The found alternative routes should not depend on chosen type, because all types use the same parameter settings and only the memory/runtime relation should change.

### 5.2.3 Route set generator

The route set generator consists of two parts. The first part is the shortest route determination, which is performed with Dijkstra's shortest path algorithm (Dijkstra, 1959). This algorithm is discussed in the literature review in section 2.2.2. Besides the shortest route, the alternative routes have to be generated as well. The accelerated Monte Carlo method is used to randomize the link costs after which the shortest path algorithm determines the newest shortest path.

The method consists of iterations in which the link costs are randomized. For each iteration the shortest path between the origin and destination is determined, which is now a potential new alternative route. A *maximum number of iterations* is determined to

terminate the route set generator. The route is checked on several filters that will be discussed in the next section.

The idea of the accelerated Monte Carlo method is to increase the variance to find routes that are different from the actual shortest route. As discussed in the last section, the variance is equal to the $\theta$-value that influences the graph's slope. The first iteration has a variance value of 0 (however exactly 0 is not possible, because it will be a negative denominator). The increase is determined with the *variance grow value parameter*.

The variance grow value parameter will be applied after a number of iterations did not find a new unique route. This number of iterations is determined with the *threshold parameter*. This threshold is used in two different ways by setting the *consecutive threshold boolean*. If the boolean is set to true, the threshold is equal to the number of subsequent iterations in which no new unique route is found. For example, if the threshold is set to 3, the variance will increase after 3 subsequent iterations without finding a new route when the boolean is set to false, the iterations without finding a new route do not have to be subsequent to increase the variance.

The table below shows an example with a threshold value of 2 and a threshold consecutive parameter set to false and true.

| Iteration | Unique route | Apply variance grow factor (consecutive threshold is false) | Apply variance grow factor (consecutive threshold is true) |
|---|---|---|---|
| 1 | Yes | | |
| 2 | No | | |
| 3 | Yes | | |
| 4 | No | Increase variance | |
| 5 | No | | Increase variance |
| 6 | No | Increase variance | |
| 7 | No | | Increase variance |

*Table 8: Apply variance grow factor by threshold value of 2 and consecutive threshold set to true and false*

**Effect of parameters**

The minimum number of iterations set the number of iterations that must be performed. A high number of iterations will probably result in many routes in the route set. Nevertheless, it is likely that increasingly fewer routes are found as a result of an increasing variance, because they will more often not accepted by the filter parameters.

The variance grow value will influence the link costs adaption. A high value results in a fast increasing variance, which takes care of finding many routes, but could miss some essential routes, because the link costs are adapted too much. A low value will find alternative route more gradually, but the parameter has to increase more often to find also the routes with a high detour.

The threshold value directly influences the times that the variance will be increased. A low value increases the variance often especially if the consecutive threshold boolean is set to false. If the boolean is set to true, it is less likely that the variance will be increased. Especially if the threshold value is high, the variance will increase slowly. A

slow increasing variance will probably result in many routes that are quite close to the already accepted routes.

**Costs**

The shortest path generation found the route with the lowest costs and not the route that is the actually the shortest. Therefore the co-called shortest path algorithm is a little bit confusing, because it is actually the cheapest path generation, but that name is not used on commonly used.

The used route set generator uses the free flow travel time as costs, which means that the shortest path algorithm will find the fastest route. The travel time as cost is assumed to create more relevant routes than the shortest route in distance.

### *5.2.4 Route filtering*

After the route set generation, each alternative route is checked by several filter criteria to prevent irrelevant alternative routes. Each alternative route is compared with the already accepted route in the current route set whereby the shortest alternative route is always accepted. An exception is the maximum detour criterion, because the comparison with the fastest route of the route set will simply result in the highest travel time detour. Each other alternative route will result in a lower detour. Once the alternative route does not satisfy one of the criteria, it will be rejected.

It is important to notice that the alternative routes are compared with each other by using the same costs as the route set generation does. Therefore, the filters are applied with the free flow travel time as costs.

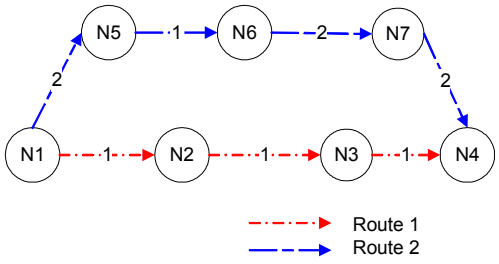
The four used criteria are discussed in this section.

**Maximum detour**

This factor takes care of new routes with a high detour in comparison to the cheapest route. It is not likely that a route with much more costs will be chosen and therefore this parameter will be applied. If the alternative route has a detour exceeding the maximum detour factor in comparison to the cheapest route (this is the cheapest route with the non randomized link costs), it will be rejected.

$$\frac{c_{alternative\ route}}{c_{cheapest\ route}} < maximum\ detour\ factor \tag{6}$$

Example




$maximum\ detour\ factor = 2$

$$\frac{C_{alternative\ route}}{C_{cheapest\ \ route}} = \frac{7}{3} = 2.33$$

The alternative route is rejected, because the route exceeds the maximum detour factor.

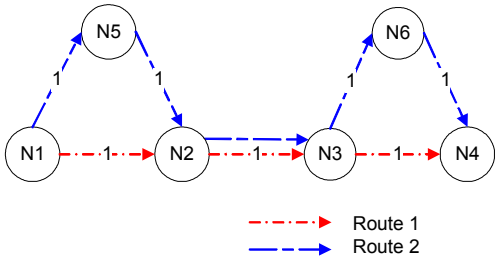
**Maximum total non-common detour**

This filter factor takes care of the non-common parts between the alternative route and the already accepted generated routes. This filter is more specific than the maximum detour, because it only takes the non-common parts into account. These are the parts (in links) wherefore the alternative route differs of the generated route.

The detour of these parts must not be too high, because this results in a route with an irrelevant detour. The value may also not be too low, because routes with relevant detours should be accepted. An alternative route that exceeds the maximum total non-common detour factor in comparison with one of the generated routes will not be accepted.

$$\frac{C_{alternative\ route\ (not\ common\ part)}}{C_{accepted\ route\ (not\ common\ part)}} < Maximum\ total\ non-common\ detour\ factor \qquad (7)$$

Example




$$Maximum\ total\ non-common\ detour\ factor = 1.5$$

$$\frac{C_{alternative\ route\ (not\ common\ part)}}{C_{accepted\ route\ (not\ common\ part)}} = \frac{4}{2} = 2$$

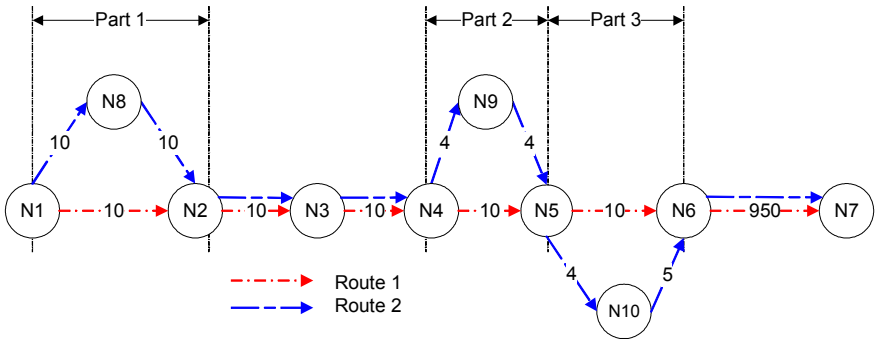
The alternative route is not accepted because the route exceeds the maximum total non-common detour factor.

**Minimum non-common detour**

This factor prevents alternative routes with a small detour on just one part of the route. An alternative route with a small detour is easily generated especially in the urban areas. For example, the choice within The Hague to drive right or left around a single house block does not matter in case of a long route and therefore one of these routes will not be accepted (the second route, because this one is compared with the already accepted route at the other side of the house block). An alternative route must exceed the minimum non-common detour factor, otherwise the route will be rejected.

$$\frac{C_{smallest\ detour}}{C_{accepted\ route}} > Minimum\ non-common\ detour\ factor \qquad (8)$$

Example




$Minimum\ non-common\ detour\ factor = 0.01$

There are three detours in the alternative route, the smallest detour occurs in part 2, because the costs are the lowest.

$$\frac{C_{smallest\ detour}}{C_{accepted\ route}} = \frac{8}{1000} = 0.008$$

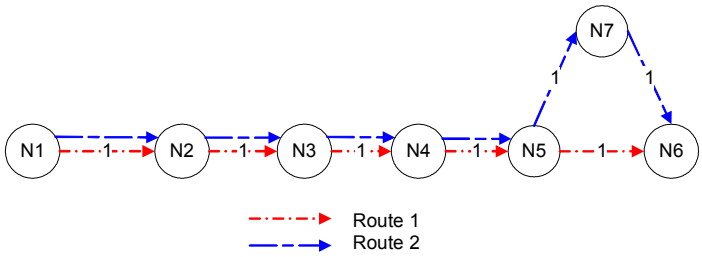
The alternative route is not accepted because the minimum non-common detour is exceeded.

**Maximum overlap**

The maximum overlap parameter prevents alternatives routes that are almost the same in comparison with each route in the current route set.

$$\frac{C_{overlap}}{C_{shortest\ route}} < maximum\ overlap\ factor \tag{9}$$

Example




$maximum\ overlap\ factor = 0.75$

$$\frac{C_{overlap}}{C_{shortest\ route}} = \frac{4}{5} = 0.8$$

The alternative route is not accepted because the maximum overlap is exceeded.

### 5.2.5 *Stopping criteria*

There are two criteria used to determine when the route set generation of a specific route set has to stop. The first criterion counts the number of routes in the current route set after adding a new alternative route. If the number of routes exceeds the *maximum number of routes*, the route set generation is finished for this route set.

The second criterion takes the variance into account. Each time that the threshold is obtained and the variance will be increased, the variance's value is checked. If this variance exceeds the *maximum variance*, the route set generation is finished for this route set.

# 6 Setting the route filter parameters

*The first section describes the observed routes as result of map matching GPS data. Section 5.2 describes the important routes which must be included in a final generated route set. At last, an analysis of the observed routes supplies relevant information for the filter parameters.*

## 6.1 The observed routes

Chapter 4 describes the map matching of GPS data with the model network. This results in map matched routes that presents the route that has been chosen by the driver. After the map matching, it was not guaranteed that the routes were all relevant to be an observed route. For example, several routes consist of two trips because the traveler made a small stop and therefore the route choice changes. All routes are checked in Section 4.7 on several criteria to be sure that the route represents an observed route. The applied criteria are briefly described below:

1. The route consists of a single trip and does not have extra stops (e.g. visiting the supermarket), which influence the chosen route;
2. The route has his origin and destination within the network without any detour outside the area;
3. Trips that have a length detour of twice the shortest route are not included;

The map matched routes that pass the criteria are selected to be an observed route. These observed routes are assumed to be relevant, which means that each route is taken by at least one person in the respondent group. There are 2136 observed routes available consisting of 1306 unique routes. Of these 1306 routes, 1001 routes are chosen only once and 305 routes chosen more than once with an average of 3.72. The routes are made between 865 different OD-pairs of which 246 OD-pairs have more than one route with an average of 2.79 routes per OD-pair.

An ideal observed route set would consist of several routes between each OD-pair, which are also made by several travelers. Unfortunately, this is not the case with this data. This is easy to explain, because of three reasons. At first, the number of OD-pairs is very high and each OD-pair is connected to a small area. Furthermore, all travelers live in Zoetermeer, which results in a selected sampling. At last, the number of travelers is not very high. The probability that two participants depart from the same OD-pair area and arrive at the same OD-pair is because of these reasons very small. By looking to groups of OD-pairs for the origin and destination, many different routes are taken by different

users. Therefore, we assume that all routes can be used, because in total they represent all different routes.

## 6.2 Filter parameters

The route set generation filters the alternative routes to prevent irrelevant routes in the route set, as discussed in section 5.2.4. The observed routes have to be part of the generated route set and therefore they must not be filtered out. This can be prevented by determining the filter parameter values such that the observed routes will be accepted. The values of the filter parameters will be determined by investigating the observed route.

It is important to notice that the parameter values may depend on the network for example. A network consisting of the entire Netherlands has another level of detail and therefore may need other parameter values to filter irrelevant routes.

### *6.2.1 Setting the filter parameters*

The route set generator uses four filter parameters. This section discusses how the value is determined for each filter parameter.

**Maximum detour**

$$\frac{C_{alternative\ route}}{C_{cheapest\ route}} < maximum\ detour\ factor$$

The maximum detour is determined by comparing all observed routes with the cheapest route between the same OD-pair. The figure below shows the cumulative percentages of the observed routes with their detour in comparison with the belonging cheapest route.

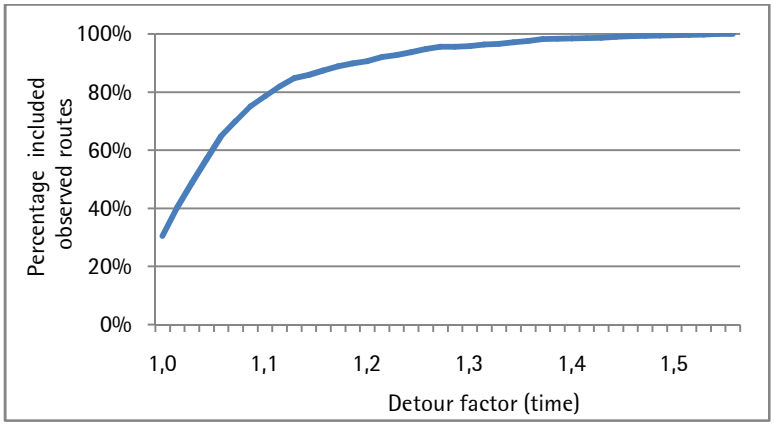



*Figure 28: the detour in time of the map matched routes in comparison with the cheapest route*

The figure shows that the highest detour between an observed route and the belonging cheapest route has a detour factor of 1.56. Because this observed route has to be included, the maximum detour must be higher and is set to a value of 1.6. There are simply no observed routes that will be excluded by using this value for the maximum detour factor.

**Maximum total non-common detour**

$$\frac{C_{alternative\ route\ (not\ common\ part)}}{C_{accepted\ route\ (not\ common\ part)}} < Maximum\ total\ non-common\ detour\ factor$$

The factor's value is determined by calculating the maximum non-common detour between each observed route and the belonging cheapest route. This results in the figure that shows the cumulative percentages of the observed routes with the belonging maximum total non-common detour.

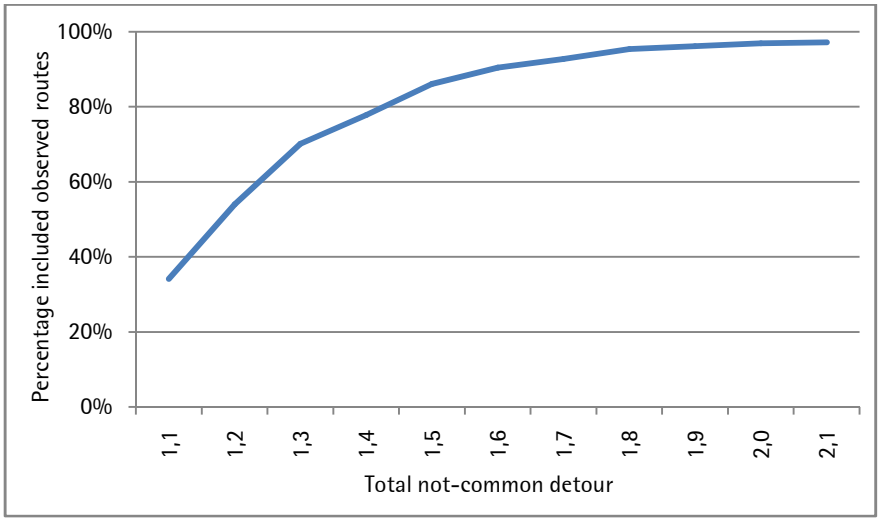
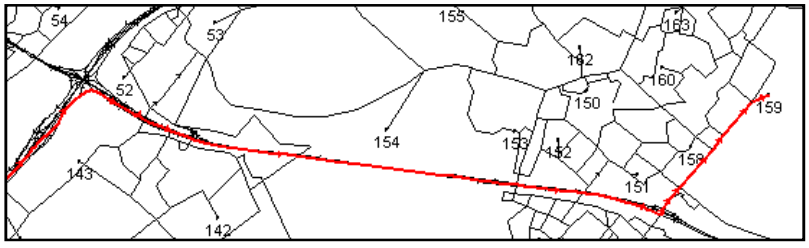



*Figure 29: the maximum total non-common detour in time of the observed routes in comparison with the cheapest route.*

An observed route with a high detour in comparison with the belonging cheapest route, is showed in the figure below. The observed route has a detour of 11.87 minutes compared with 5.84 minutes for the cheapest route. Nevertheless, the observed route is chosen by several travelers and is also likely to be chosen during rush hours. Therefore the observed route must not be filtered out through the maximum total non-common detour.

The total non-common detour of the observed route is 2.03. As showed in the figure above, there are no observed routes with a higher maximum non-common detour. The value for the maximum total non-common detour factor is set to a value of 2.1 to include observed routes like this.



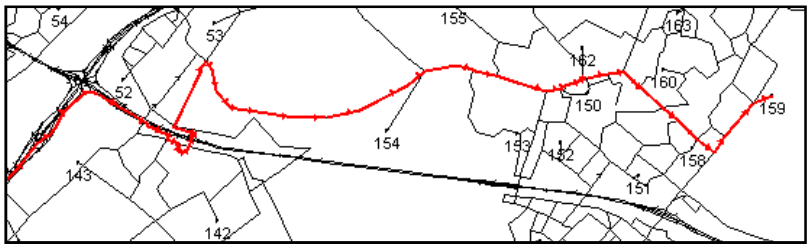



*Figure 30: the cheapest route (5.84 min) and the observed route with a detour (11.87 min)*

**Minimum non-common detour**

$$\frac{C_{smallest\ detour}}{C_{accepted\ route}} > Minimum\ non-common\ detour\ factor$$

The minimum non-common detour value must be lower than the smallest detour between two routes, which are both relevant and must be accepted.

The factor is determined by investigating the two closest off-ramps on the main roads. These off-ramps results in a small detour, but had essential impact on the main roads and results therefore in two relevant observed routes. The figure below shows the detour passage of the routes.

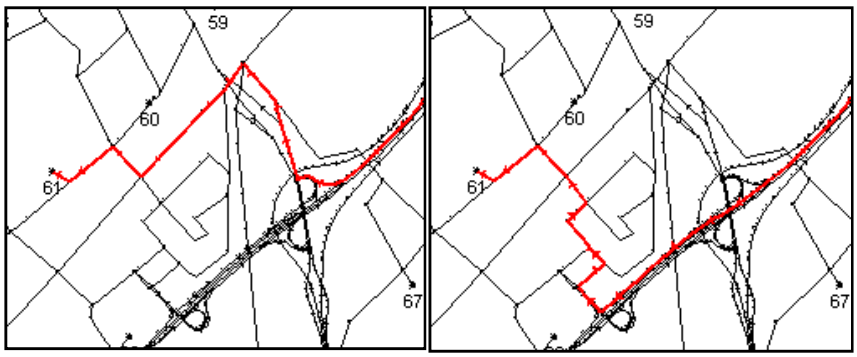



*Figure 31: two routes to the same destination using a different off-ramp*

The total costs of the first observed route including the longest route to this destination in the network are 18.8 minutes. In comparison with the other route, the minimum detour is 2.26 minutes. This results in a factor of 0.12 as non-common detour factor. Because, both routes must be accepted, the actual parameter value must be lower than 0.12 and is set to 0.10. This value will accept the second observed route for this off-ramp.

**Maximum overlap**

$$\frac{C_{overlap}}{C_{shortest\ route}} < maximum\ overlap\ factor$$

An observed route that must be included but has much overlap with an earlier accepted route is mostly a route that includes another off or on-ramp of the highway. The figures below show the two different off-ramps of the city Zoetermeer. We assume that the rest of the route is similar. The route with the second off-ramp will not be accepted if the total route costs of the route with the first off-ramp are very high, because the overlap will exceed the maximum overlap factor.

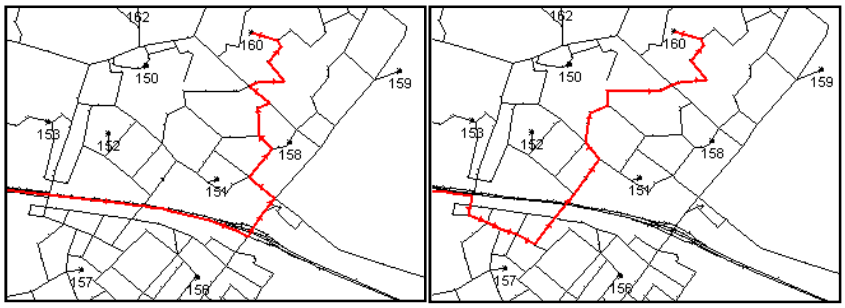



*Figure 32: two routes with the same destination using a different off-ramp*

The maximum cost of the first observed route is 21.5 minute; this is measured of the OD-pair that is the farthest away in the network. The overlap of both routes is 15.7 minute, resulting in an overlap factor of 0.73. Some other routes with a high overlap are also compared with the total route costs, but do not exceed the value of 0.73. The maximum overlap factor is set to a value of 0.75, which makes sure that all routes with a high overlap, but have a different important part of another observed route, are accepted.

Especially this value depends on the network. In case of a large network with large distances, this value must be higher to prevent irrelevant routes. For this relatively small network, the parameter is set correctly to include both relevant routes.

### 6.2.2 Summary

The values for four filter parameters are determined in the previous section. These values are such determined in such way that they will not filter out the relevant observed routes. The determined values are presented in the table below.

| Filter | Value |
|---|---|
| Maximum detour | 1.6 |
| Maximum total non-common detour | 2.1 |
| Minimum non-common detour | 0.10 |
| Maximum overlap | 0.75 |

*Table 9: filter parameter with the determines values*

# 7 Calibration of route set generation

The final purpose of this research is to calibrate the route set generation. The calibration is performed by comparing the observed routes with the generated routes and tries to include as much as possible observed routes within the generated route sets without creating generated route sets including irrelevant routes. This calibration will be discussed in this chapter.

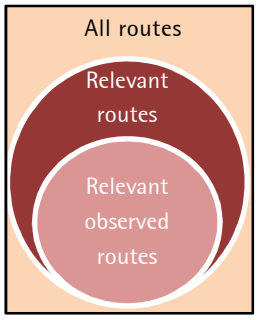
*Section 7.1 describes the approach to calibrate the route set generation. Thereafter an analysis to the randomization is performed in an intermezzo. The calibration of the route set parameter is discussed in section 7.2, followed by section 7.3 with an analysis of the routes which are not included in the final route sets. Section 7.4 supplies the conclusions and at last the limitations and recommendations are described in section 7.5.*

## 7.1 Approach

The approach will first discuss how the observed routes will represent all relevant routes, so that the generated routes sets are really comprehensive. After this, a distance measure is introduced to compare the observed routes with the generated routes. The next section adds two additional criteria for the generated route sets. The last section discusses how the parameters are calibrated and which parameters are also known through earlier analysis.

### *7.1.1 Comprehensive route set*

A comprehensive route set is defined as a route set that includes all relevant routes. The relevant routes are routes that are taken by at least one person of the respondent group. Unfortunately, the observed routes do not contain all the relevant routes, because the data source is simply too restricted. It is only possible to receive all relevant routes if all travelers' trips are known. As not all travelers trips are available, it is quite certain that not all relevant routes have been observed.




There will be assumed that the observed routes can also represent all other relevant routes in the network and therefore if all the observed routes are included in the comprehensive route set, the relevant routes are also included though the effort to include the observed routes. This effort is the calibration of the route set generation parameters whereby as much as possible observed routes must be included.

This assumption will be explained by an example. Figure 33 shows two OD-pairs with four relevant routes, the dashed routes are also observed. The observed route with a travel time of 12 minutes between A1 and B1 will be included during the parameter calibration. This result can be reflected on the relevant route between A2 and B2 with the same travel time, but which is not included in the GPS data set. It is likely that the parameter settings that include the observed route with a travel time of 12 minutes between A1 and B1 will also be included between A2 and B2.

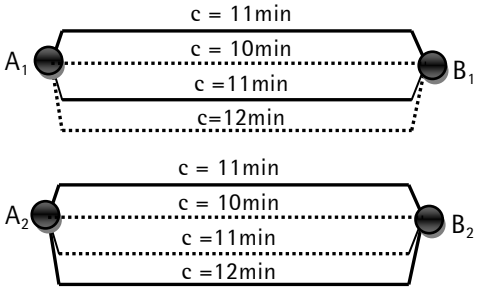



*Figure 33: comparing two route sets*

The large effort to include observed routes with a high travel time will result that other relevant routes with a high travel time will also be included in the route set. The negative effect of this could be the inclusion of irrelevant routes. These routes have to be excluded by using the filters. The filters have to be adjusted so that they will accept the relevant routes and remove the irrelevant routes. Because some observed routes might be difficult to include in a route set and result in many irrelevant routes, a full match is not feasible.

### 7.1.2 Distance measure

For each OD-pair $(i,j)$ in a predefined set of OD-pairs $OD$ there are actually chosen relevant routes $R_{obs}^{ij}$. The $R_{obs}^{ij}$ have to be compared with the generated routes $R_{gen}^{ij}(p)$ depended on the parameters $p$ to determine if $R_{obs}^{ij}$ is included in the generated route set. It is likely that an observed route is not completely included in this set, but has a small detour in comparison with one of the routes in the generated route set. In this case, a 100% match between the $R_{obs}^{ij}$ and $R_{gen}^{ij}(p)$ will never be possible, because the route set generation filters will not accept a generated route that is nearly the same to another route in the generated route set (overlap filter).

For this, it is not desired to require a full match between $R_{obs}^{ij}$ and $R_{gen}^{ij}(p)$. A deviation between the routes that does not occur on the main roads has to be accepted too. This is an important assumption to notice that is made because the traffic research is directed on main roads and small differences on local roads are considered to be not important.

The purpose is to find the $p$ that maximizes the number of $R_{obs}^{ij}$ included in the $R_{gen}^{ij}$ so we want to find the $p$ that maximizes $\Delta(R_{obs}^{ij}, R_{gen}^{ij}(p))$. The $\Delta$ is called the distance measure, which will be discussed in this section.

**Explanation of the distance measure**

Let $r$ be an element of an observed route $R_{obs}^{ij}$ and let $s$ be an element of a generated route $R_{gen}^{ij}$. We define $\delta(r,s) = 1$ when $r$ and $s$ are considered to be equal and $\delta(r,s) = 0$ otherwise. This equality is confirmed if the overlap between $r$ and $s$ exceeds a threshold value, where overlap means the percentage of common links, weighted according to the distance. The choice to use the distance, and not the travel time, will be underpinned in the next paragraph.

In order to describe these overlap, we have to define $r$ and $s$ in more detail. Let us describe $r$ as a route, i.e. an ordered set of links, $r = \{l_1^r \dots l_n^r\}$ and $s$ as another route, $s = \{l_1^s \dots l_m^r\}$..

Further, it is supposed that each link $a$ has a certain costs indicated by $\lambda(a)$. The overlap between $r$ and $s$ is $r \cap s = \{a | a \in r \wedge a \in s\}$. The costs of a route $r$ are the costs of all links in $r$, or $\lambda(r) = \sum_{a \in r} \lambda(a)$. So the relative weighted overlap $O(r,s)$ of route $s$ compared to route $r$ is:

$$O(r,s) = \frac{\lambda(r \cap s)}{\lambda(r)} \cdot 100\%$$

**Used costs for the relative weighted overlap**

We have to find a threshold value for the relative weighted overlap, wherefore we can say that the observed route is equal with the generated route. A high value for this has to prevent that routes, which are apparently different, are not accepted. For example, if the threshold for the relative weighted overlap is set to 90%, both routes have to be equal for 90% in the costs to accept the observed route as equal to the generated route. Although, an observed route is accepted if $O(r,s)$ exceeds a threshold with one of the routes of the generated route set. These costs are set to the distance and this paragraph describes why not using the travel time. So, actually a link $a$ has a length indicated with $\lambda(a)$.

A disadvantage of using travel time is the point that irrelevant roads (e.g. the local roads) have a relative high free flow time and with this influence $O(r,s)$ largely. The influence of travel time on a route with no overlap at the local roads is large if this route consists further of a highway with overlap. Travel time at the highway is relative low by which the irrelevant detour with a high travel time results in a low match quality. The figure below shows a relevant situation.

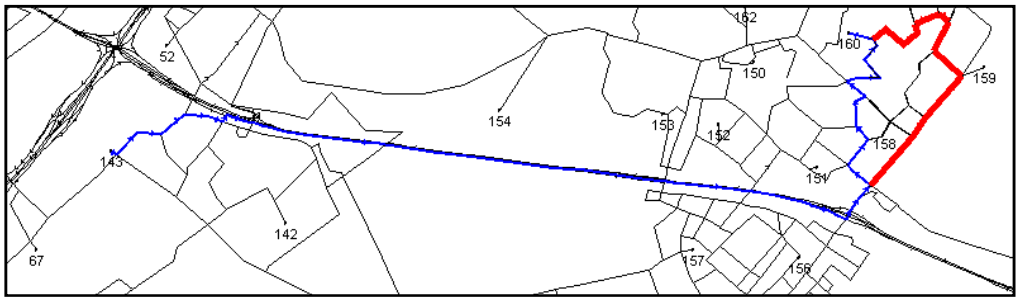



*Figure 34: influence of travel time by comparing a generated route (thin) with an observed route (thin - bold)*

The observed route (thin bold) and generated route (thin) are assumed to be similar, because there are no differences on the main roads. $O(r,s)$ is now calculated with the travel time and distance as costs:

$$O(r,s) = \frac{\lambda(r \cap s)}{\lambda(r)} \cdot 100\% = \frac{6.99}{10.33} \cdot 100\% = 68\%$$

With:

$\lambda(a)$ = indication of the travel time as costs of each link $a$

$$O(r,s) = \frac{\lambda(r \cap s)}{\lambda(r)} \cdot 100\% = \frac{10.4}{13.19} \cdot 100\% = 79\%$$

$\lambda(a)$ = indication of the distance as costs of each link $a$

The result of the calculations shows the large influence of the travel time and with this, the lower needed threshold value for the $O(r,s)$ to accept the observed route. A low value for the $O(r,s)$ is not desired, because observed routes that are not similar to the

generated route could also be easily accepted. Therefore, the distance is chosen as costs for the relative weighted overlap.

**Determination of threshold value**
It is difficult to determine the threshold value of the relative weighted overlap precisely, because the correctness depends on the fact that the main roads must be matched correctly and a percentage does not take this into account. A route consisting of main roads satisfies a higher match percentage in contrary to a road with a large detour on irrelevant local roads.

The situation presented in the previous section by Figure 32 is used to determine the threshold value. The length of the total route is 26.78km and the non-common part is 7.11km, which means that $O(r,s) = 73.5\%$. The alternative road is interesting, because it effects the main road in another way (e.g. less vehicles on the highway, because they use an off-ramp earlier). Therefore, the alternative route is not similar to the shortest route and must not exceed $O(r,s)$. $O(r,s)$ has to be larger than 73.5% to distinguish these two routes.

With the information of the last paragraph, the threshold value is set to 75%, which means that observed routes are considered to be equal to a generated route if they have a similar length of at least 75%, so $O(r,s) > 75\%$. This value is an equilibrium value between the two considerations. Unfortunately, it cannot prevent that some routes might be accepted while they actually differ on the main roads (should not be accepted) or at the other side differ on local roads for more than 25% of the route length (should be accepted).

### *7.1.3 Additional criteria*

There are two additional criteria used besides the distance measure with relative weighted overlap of at least 75%. Both criteria must be obtained to generate a relevant route set.

**Important observed routes**
The observed routes will be used to calibrate the route set generation, which will be described in the next chapter. During this calibration, it is likely that not all observed routes will be included in the generated route sets. Zantema et al. (2007) experienced that during their route set calibration the non-motorway routes were difficult to include in the generated route sets. This was caused by the high free flow travel time of non motorway routes in comparison to the "fast" highway. Therefore the used route set generator had difficulty to include the non-motorway routes.

In order to prevent this problem during this research, five OD-pairs will be selected that have an observed motorway and observed non-motorway route. Both routes between these OD-pairs must be included during the route calibration.

*Figure 35: areas represented by OD-pair 1*

The five OD-pairs are selected by analyzing the often taken observed motorway and non-

motorway routes between two areas. The OD-pairs represent the areas where the routes depart and arrive as shown in Figure 35. For all the OD-pairs there are at least 25 observed routes for the motorway and non-motorway together.

The figure below shows the 5 origins and 5 destinations that constitute the five OD-pairs indicated with the numbers 1 till 5.

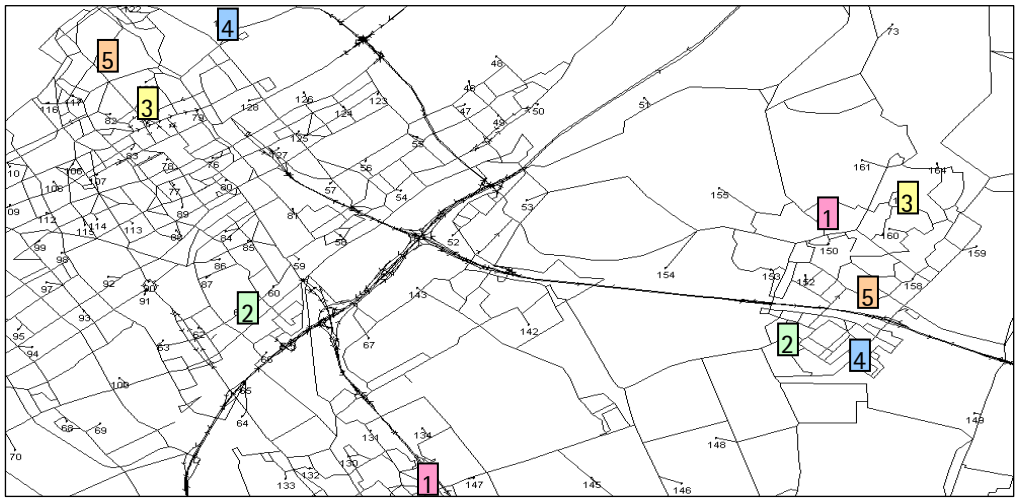



*Figure 36: overview of the selected five origins and destination that constitute the five OD-pairs, which must include a motorway route and a non-motorway route in the route set of this OD-pair.*

**Average number of routes in generated route set**

One of the stop criteria is the maximum number of routes in a route set. The performed analysis in section 6.1 shows that the observed routes, which are chosen more than once, have average of 2.79 routes per OD-pair. Several OD-pairs consists of six routes, but more routes in a route set are really exceptional. An analysis of the ten selected observed routes shows more than six routes will result in irrelevant routes, because there are not so many relevant routes. Therefore, there is decided that the average number of routes in a generated route must not be higher than 5 routes. The single exceptions could exceed this number, because there are enough OD-pairs with at the most three relevant routes.

The average number of routes will also used as criteria in case two parameter sets both include the ten selected observed routes and has an equal value for the distance measure.

### *7.1.4 Parameters and settings*

The parameters of the route set generation and their effects are discussed in section 5.2.3. The calibration of these parameters and some assumptions are described in this section.

**Randomization**

The most favorable type of randomizing is the 'all to all' method as described in section 5.2.2, because the lower runtimes. The different randomizing types will be tested making sure that there is no quality difference. This will be a separated analysis performed in the next section.

**Route set generation**

The route set generation consists of four parameters, which are shown in the table below.

| Parameter name |
|---|
| Minimum iterations |
| Variance grow value |
| Threshold |
| Consecutive threshold |

*Table 10: parameters for the route set generation*

The consecutive threshold is set on true to be sure that there are no alternative routes found. Especially for the low threshold values (e.g. 2 and 3), the variance will grow very fast and could result in many routes with a detour, but jump over the routes which are quite close to the cheapest route.

The other three parameters have to be calibrated by performing route set generations and comparing the observed routes with the generated routes. At first, it is relevant to estimate the number of iterations. This number is estimated by checking how many of the ten selected observed routes are included in the generated route sets. The table below shows the cases to be tested.

| | OD-pair 1 | | OD-pair 2 | | OD-pair 3 | | OD-pair 4 | | OD-pair 5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Iterations | Mw | NMw | Mw | NMw | Mw | NMw | Mw | NMw | Mw | NMw |
| 10 | | | | | | | | | | |
| 20 | | | | | | | | | | |
| 30 | | | | | | | | | | |
| 40 | | | | | | | | | | |
| 50 | | | | | | | | | | |
| 75 | | | | | | | | | | |
| 100 | | | | | | | | | | |

*Table 11: route sets to be generated to test the inclusion of motorway and non-motorway routes for the five selected OD-pairs*

The number of iterations, in which the 10 selected routes are included, will give a good indication for the above parameters, but it is likely that these parameters will change to further optimize the route set generation and so the number of iterations will change too.

The second step is to investigate a relation between the threshold and the variance, which could simplify the calibration. With this result, several parameter combinations will be tested about the including of the ten selected observed routes.

The two stopping criterion will be used to fine tune the generated route sets, which includes the ten observed routes. Applying both parameters will probably result in a decreasing number of routes in the generated route sets. Although, the parameters might filter some observed routes, a good assessment must be made about these contradictions.

## Intermezzo Randomization analysis

The four types of randomizing are compared with each other by using the percentage of observed routes that are included in the generated route sets. For this analysis, the default route set generation parameters are used as presented in section 7.1.3, only the

minimum number of iteration depends to present the differences in inclusion of observed routes. Figure 37 shows the percentages of included observed routes for several minimum iterations.

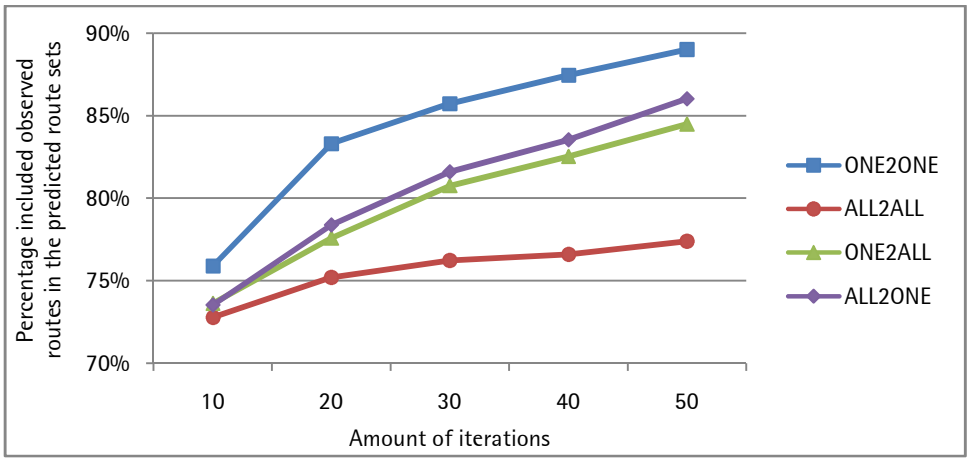



*Figure 37: The percentage matched routes with a variance of 0.02 and a threshold of 3 for 900 OD-pairs*

In contrast to our expectations described in section 7.1.3, Figure 37 shows large differences between the four randomizations types. These differences could be caused by change by the randomizing of the link costs, but it seems to be that the 'all to all' type consequently has a lower number of included observed routes. The 'one to one' method shows a constant high value of the included observed routes.

The reason for the difference has been found when different numbers of OD-pairs were used for the route set generation. All types without the "one to one" method do only increase the variance if no alternative routes are found for all routes starting from the origin (one to all), arriving at a destination (all to one) or for all origins and destinations (all to all). These methods have a lower runtime, because they are checking all origins and destination once. This results in a negative effect on the route generation. The more OD-pairs are included in the route set generation, the smaller the chance that no alternative route is found. Therefore the variance will not increase and no new alternative routes will be found. Only the 'one to one' method is checking the alternative routes per OD-pair and increases the variance correctly.

The figure below shows the average number of routes in a route set for the four randomization types. With the knowledge that more routes most of the time result in more matched routes, the results of the previous figure are explained.

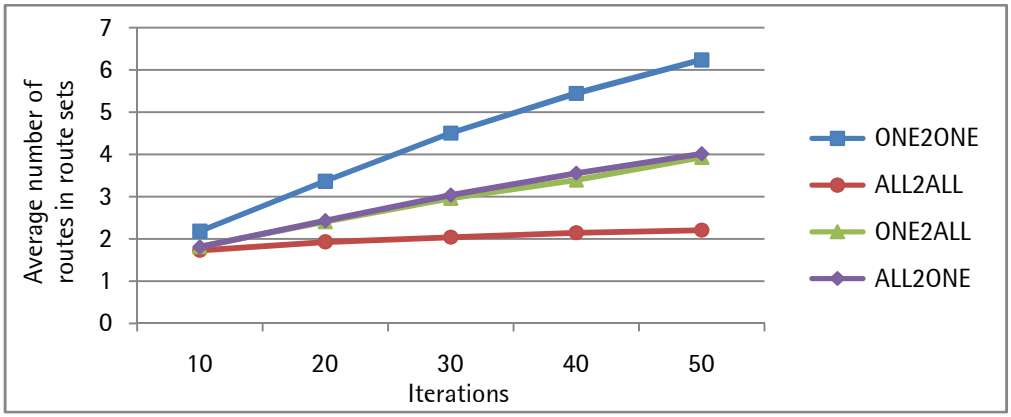



*Figure 38: the average number of routes in the route set depending on the number of iterations*

The figure below shows the decreasing number of routes in the route set, which is generated with the 'all to all' type depending on the number of OD-pairs. It is clearly seen that more OD-pairs results in fewer routes in the route sets.

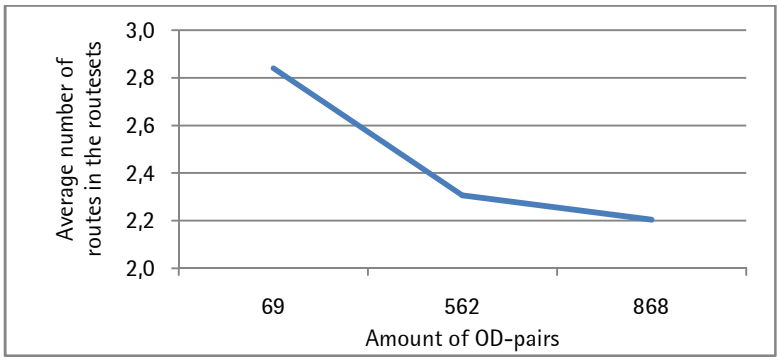



*Figure 39: the declining average number of routes in the route set depending on the number of OD-pairs*

The conclusion of the randomization analysis is that the types are surprisingly differing in results. Because it is no option to improve the "all to all" method during this research, the "one to one" type will be used for the rest of this research, because this is the only type that increases the variance correctly and with this gives correct results.

## 7.2 Parameters calibration

The parameters used by the route set generation are all calibrated in this section. The route set generation is performed with the already determined values for the filter parameters in section 6.2. Besides this, the five selected OD-pairs of section 0 have also an important role during the calibration.

### *7.2.1 Route set generation parameters*

The calibration of the route set generation parameters is performed in this section. The theoretically background of the parameters was presented in section 5.2.3.

**Number of iterations**

An estimation of the number of iterations is performed by checking the inclusion of the ten selected observed routes in the generated route sets. The five OD-pairs are shown in Figure 36 on page 55. Except the variable number of iterations, the parameter values do not change. The default parameter values are for threshold (3) and variance grow value (0.02). The table below shows the observed routes that are included depending on the number of iterations.

| | OD-pair 1 | | OD-pair 2 | | OD-pair 3 | | OD-pair 4 | | OD-pair 5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Iterations | Mw | NMw | Mw | NMw | Mw | NMw | Mw | NMw | Mw | NMw |
| 10 | x | | x | | x | | x | | x | |
| 20 | x | | x | | x | x | x | | x | |
| 30 | x | | x | | x | x | x | | x | |
| 40 | x | x | x | | x | x | x | | x | |
| 50 | x | x | x | | x | x | x | | x | |
| 75 | x | x | x | | x | x | x | | x | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 100 | x | x | x | x | x | x | x | | x | x |

*Table 12: inclusion of the 10 selected observed routes in the generated route sets depending on the number of iterations (Mw: motorway, NMw: non motorway)*

As expected an increase of the number of iterations results in more included routes. For the highest number of iterations only one non motorway is not included. A route set generation with more than 100 iterations is not desired, because of high runtimes. Despite this, the non motorway is even not included when 300 iterations are performed. This clearly indicates that other route set generation parameters have to be changed.

**Relation between the variance grow factor and threshold parameter**

The variance grow value and threshold parameter have to be calibrated by comparing the observed routes with the generated route sets. It should be practical if these parameters are related to each other to simplify the calibration. A higher threshold results in a slower increasing variance, but in more attempt to find new routes at the same variance. A high variance takes care of many new routes and the threshold must be lowered to increase the variance further. Figure 40 shows the relation between five threshold values and two variance grow values.

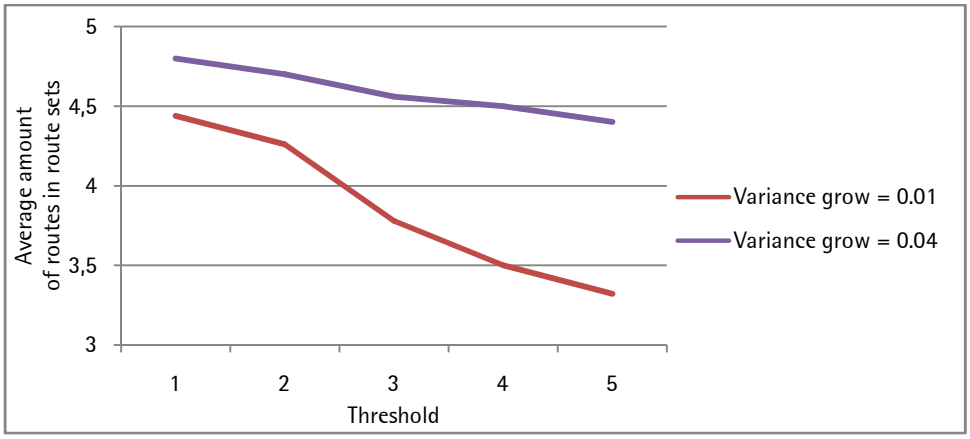



*Figure 40: Average number of routes in route set for several variance grow and threshold values performed with 50 iterations.*

A linear relation between the threshold and variance grow will simplify the calibration a lot, because one parameter can be determined by setting the other. The possible linear relation between these values is investigated by comparing the average reached variance after the route set generation is performed. In case of a linear relation between these parameters, the reverse combination must result in the same final variance. The final variance is the variance used for the last iteration. For example, a route set generation with a variance of 0.01 respectively 0.03 and threshold of 1 respectively 3 should result in the same final variance is no new routes are found.

The two tables below show the final variance and percentage of matched observed routes for different values of the variance and the threshold.

| Threshold | Variance grow | Average number of routes | Routes matched | Average increased variance |
|---|---|---|---|---|
| 1 | 0.01 | 2.29 | 76.1% | 0.15 |
| 2 | 0.02 | 2.42 | 77.5% | 0.13 |

| | | | | |
|---|---|---|---|---|
| 3 | 0.03 | 2.57 | 79.0% | 012 |
| 4 | 0.04 | 2.54 | 79.3% | 0.12 |
| 5 | 0.05 | 2.65 | 77.8% | 0.11 |
| 6 | 0.06 | 2.70 | 78.2% | 0.10 |

*Table 13: relation between the threshold and variance grow value for several combinations performed by a route generation of 20 iterations*

| Threshold | Variance grow | Average number of routes | Routes matched | Average increased variance |
|---|---|---|---|---|
| 1 | 0.01 | 4.01 | 83.9% | 0.37 |
| 2 | 0.02 | 4.25 | 85.0% | 0.31 |
| 3 | 0.03 | 4.25 | 84.0% | 0.27 |
| 4 | 0.04 | 4.31 | 84.5% | 0.24 |
| 5 | 0.05 | 4.22 | 85.6% | 0.23 |
| 6 | 0.06 | 4.21 | 83.9% | 0.21 |

*Table 14: relation between the threshold and variance grow value for several combinations performed by a route generation of 50 iterations*

Unfortunately, the tables show no relevant relation between the both parameters. The average increased variance decreases for lower variance grow values and higher threshold values. The percentage of matched routes and the average number of routes differs for all values. This makes it not possible to calibrate only one parameter and knowing the other optimal parameter.

**Inclusion of non motorway and motorway routes**

The five selected OD-pairs are used to determine the parameter combinations that include the motorway and non-motorway routes. The difficulty is that this depends also on the number of iterations. The results in the beginning of this section show that 100 iterations includes 9 routes, therefore it is assumed that with some other parameter combinations all the 10 routes are included.

As showed in the previous two tables, there is no correlation between the matched percentages and the used parameters, which means that each combination has to be tested. The table below shows the number of selected routes matched for 30 combinations of the threshold and the variance values.

| Variance grow<br>Threshold | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|---|---|---|---|---|---|
| 0 | 10 | 10 | * | * | * |
| 1 | 9 | 10 | 9 | 8 | * |
| 2 | 7 | 10 | 9 | 9 | 8 |
| 3 | 9 | 9 | 8 | 9 | 9 |
| 4 | 7 | 8 | 8 | 10 | 9 |
| 5 | 7 | 8 | 8 | 10 | 10 |

** overflow error because of the fast increasing variance*

*Table 15: the number of routes that are included of the ten selected routes for the route sets generated with 100 iterations*

The table above shows the number of selected observed routes that are matched. Seven parameter settings include all selected ten routes (motorway and non-motorway). Furthermore, four parameter combinations cannot be tested because the variance grows too fast, which results in an overflow error.

The number of iterations, for which all ten observed routes are included, result in a quite high number of average routes in a route set. The route set with the best relation between the matched score and the number of routes in the route set is chosen as optimal. Figure 41 shows the relations between the matched percentage and the average number of routes in the route sets.

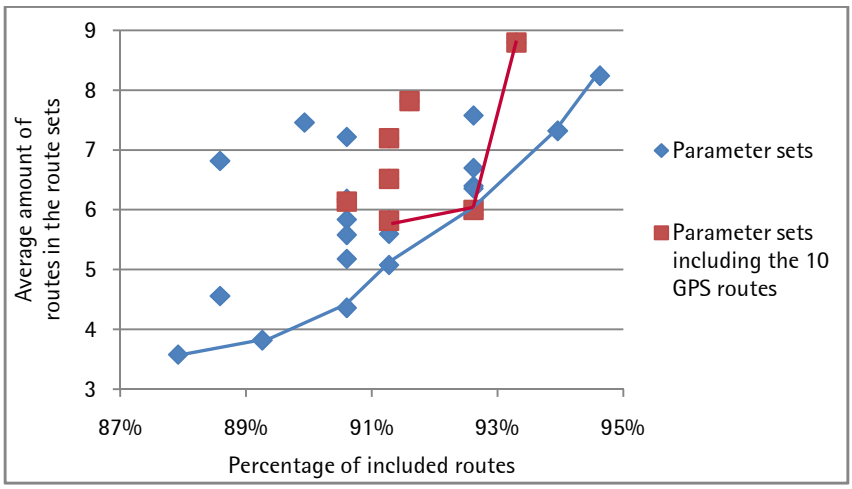



*Figure 41: parameter settings with their average number of routes in the route sets and the percentage of included routes. The route set generation is performed with 100 iterations for 50 random chosen OD-pairs.*

A Pareto frontier is available consisting of 7 optimal parameter sets, but not all the sets include the ten selected observed routes. Another small Pareto front shows the three optimal parameter combinations that do include the ten selected routes. Both frontiers consist of the optimal combinations, all other combinations results in less favorable results.

Figure 41 shows only one parameter set as part of the Pareto frontier, which includes the 10 selected observed routes (average number of routes is 6 and 92.6% of the routes is included). Although, this parameter set seems to be optimal, the number of routes has to be lowered, because the average number of routes is mostly higher than the maximum of 5 routes. This is done by performing two final criteria for each OD-pair: the maximum number of routes and the maximum variance. The first criterion stops the generation of new routes between an OD-pair when the criterion's value is reached. The maximum variance does exactly the same, although it is now checking the used variance.

The table below shows the seven parameter sets, which include the 10 selected observed routes. The second part of the table shows the optimal variable settings that still include the 10 selected observed routes and despite this also result in a lowest value for the average number of routes in a route set. The belonging quality and average number routes per route set are shown in the third part of the table. The variable settings are determined by first changing the number of iterations and then decreasing the number of maximum routes and variance per OD-pair. In some cases, the variable parameters are adapted to increase the quality and do not exceed the average number of five routes for the route sets.

| | Variance grow | Threshold | Iterations | Maximum variance | Maximum routes | Quality | Average number of routes |
|---|---|---|---|---|---|---|---|
| 1 | 0.01 | 0 | 75 | 0.45 | 6 | 89.3% | 4.84 |
| 2 | 0.02 | 0 | 80 | 0.75 | 6 | 92.6% | 5.46* |
| 3 | 0.02 | 1 | 75 | 0.45 | 5 | 87.2% | 4.32 |
| 4 | 0.02 | 2 | 70 | 0.35 | 5 | 87.9%** | 4.18 |
| | | | 90 | 0.35 | 6 | 90.6% | 4.92 |
| 5 | 0.04 | 4 | 90 | 0.35 | 6 | 90.6% | 4.72 |
| 6 | 0.04 | 5 | 90 | 0.35 | 5 | 90.6%** | 4.24 |
| | | | 100 | 0.30 | 6 | 91.3% | 4.72 |
| 7 | 0.05 | 5 | 90 | 0.30 | 6 | 91.3%** | 4.46 |
| | | | 90 | 0.30 | 7 | 92.6% | 4.80 |

** the average number of routes cannot be lower, because the 10 selected observed routes won't be included in this case.*

*** the quality is improved in the row below by changing the parameter settings. This results also in higher average number routes per route set.*

*Table 16: parameter settings with optimal settings to include the 10 selected observed routes, but also have as few as possible average routes in the route sets. The second row for some cases shows the increase of the average number of routes in the route sets, but below the five, and including more observed routes in total.*

The parameter setting with a threshold of 5 and a variance of 0.05 is most optimal. This set includes 92.6% of the routes for the sample of 50 OD-pairs. The parameter combination with a variance of 0.02 and a threshold of 0 results also in match of 92.6%, but has a higher average number of routes.

The seven optimal settings (as low average number of routes in route set as possible) and the three improved settings (increase the match percentage with a higher average number of routes in route set) are shown in the figure below.

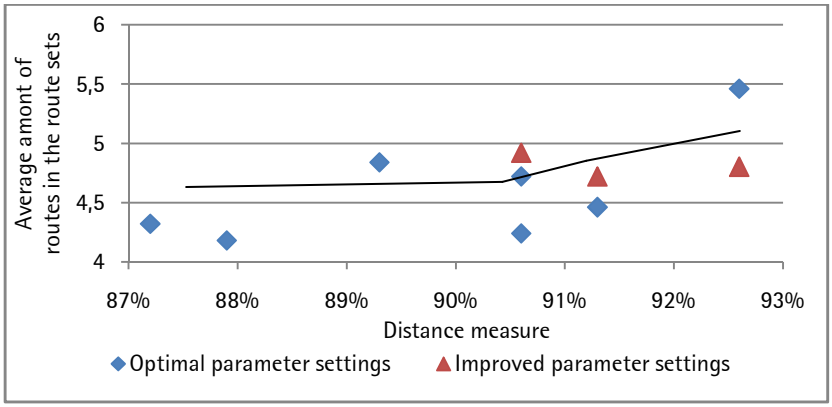



*Figure 42: optimal and improved parameter settings with their average number of routes in the route sets and the percentage of included routes. The route set generation is performed with 100 iterations for 50 random chosen OD-pairs.*

The figure shows clearly that the last parameter combination, the most right triangle, of the Pareto front is optimal. It includes the 10 selected observed routes, has an average number of routes in the route sets less than 5 and the matching percentage is quite high. With this last calibration, the optimal parameters are known and presented in the table below.

| Parameter | Value |
|---|---|
| Route generation | |
| Variance grow | 0.05 |
| Threshold | 5 |
| Threshold consecutive | true |
| Iterations | 90 |
| Stop criterion | |
| Maximum variance | 0.30 |
| Maximum routes | 7 |

*Table 17: Calibrated parameters*

A final run for all 868 OD-pairs results in a matched percentage of 88.63%. This difference is coincidence because the 50 OD-pairs, wherefore the parameter settings were calibrated, were selected randomly. A run for the other optimal parameter settings included in the Pareto front shows the same decrease matching percentage.

## 7.3 Analysis of not included observed routes

Approximately 11% of the observed routes are not included in the generated route set. It is interesting to investigate this group. The assumption is that routes with a high detour are often included in the route sets. The figure below shows the percentage matched observed routes in relation to the detour in free flow time of the observed route in comparison to the shortest route.

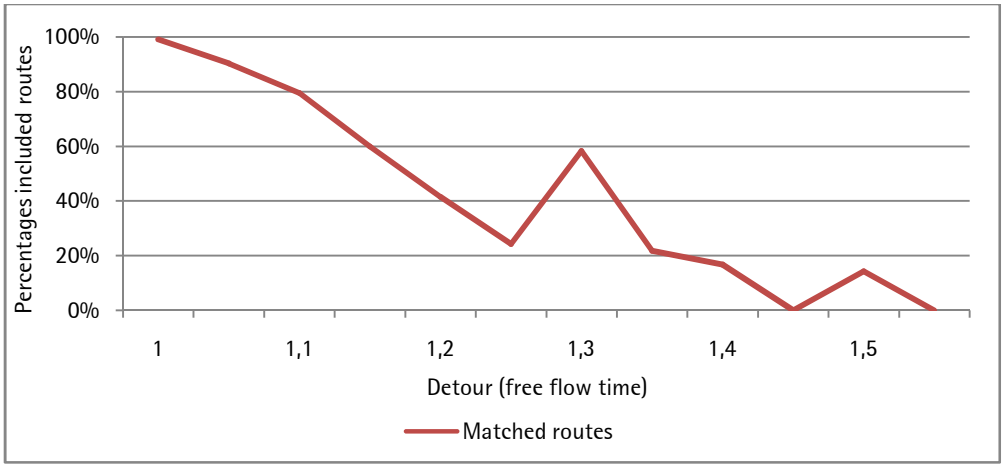



*Figure 43: percentages matched and not matched routes in relation with the detour*

Although, the graph is descending, it is important to take the number of routes into account that are available for each detour. The table below shows these numbers of routes and the number of routes that are included.

| Detour | 1 | 1.05 | 1.1 | 1.15 | 1.2 | 1.25 | 1.3 | 1.35 | 1.4 | 1.45 | 1.5 | 1.55 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Total number of routes | 713 | 268 | 142 | 60 | 41 | 33 | 12 | 23 | 6 | 8 | 7 | 3 |
| Number of included routes | 6 | 26 | 29 | 24 | 24 | 25 | 5 | 18 | 5 | 8 | 1 | 0 |

*Table 18: the total number of routes and the included routes distributed to the detour*

Despite the optimalization progress, the route set generation with the optimal parameters, does not include most routes with a high detour. The more the detour increases, the fewer observed routes are included in the route sets. Despite this result, most routes are matched correctly and especially for the routes with a detour less than a factor of 1.1.

The difficulty to include the routes with a higher detour can be indicated as a problem of the used route set generator, because the optimal parameters are used. The table below supplies a global view of the reason why the observed routes are not included in the route sets.

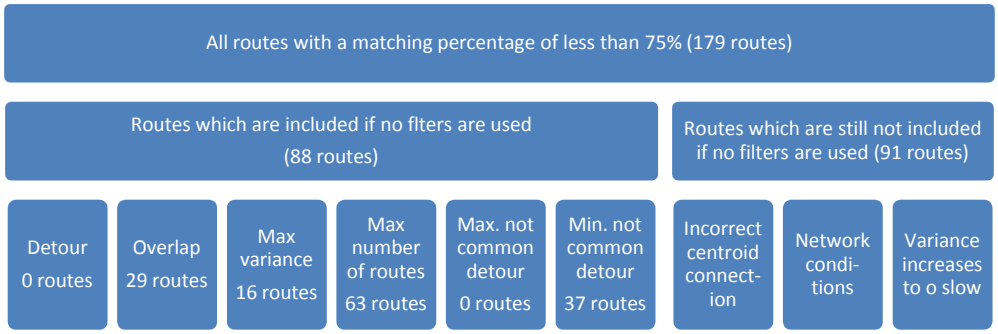



*Figure 44: overview of the routes with a matching percentage of less than 75%*

Approximately half of the excluded routes will be included by not applying the filter restrictions. Especially the maximum number of routes filter removed many routes.

### Influence of the filters

The main reason of the high number of routes in each route set is found at the filters. These filters are not strict enough, because the irrelevant routes are still included in the route sets. At the other side, the filters cannot be more strict, because relevant routes won't be accepted. With the chosen filter values, some outstanding observed routes are already not accepted.

Research found out that the filters works correct, but the used costs are not practical to exclude irrelevant routes. The route set generator uses the free flow time to compare the generated routes. The free flow time is relative low at local roads in relation to the highways. Therefore, the overlap of irrelevant routes is in distance relative large, but in travel time lower than the maximum overlap factor. The two figures at the right show an example of an accepted alternative route and the new irrelevant alternative route.

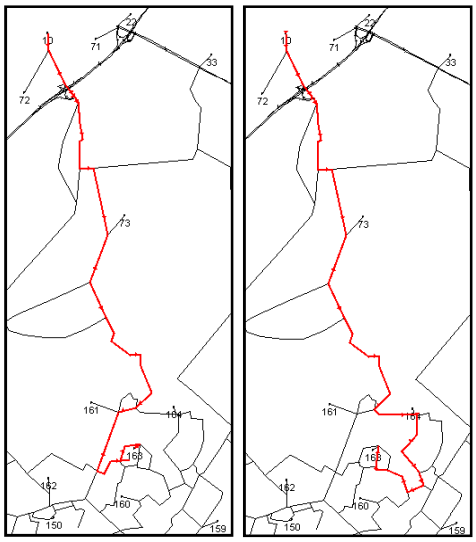



*Figure 45: the accepted and correct alternative route (right) and the alternative route (left) which is incorrectly accepted if the travel time is used is as cost for the filter.*

The accepted alternative route has a distance of 11.64km and a free flow time of 10.04 minute. The irrelevant alternative route has a distance of 12.45km and a free flow time of 14.22 minute. The overlap of the routes is

9.50km in distance and 7.02 minute in free flow time. This results in the following calculations:

$$\text{Overlap factor (free flow time)} = \frac{C_{overlap}}{C_{shortest\ route}} = \frac{7.02}{12.45} = 0.490$$

$$\text{Overlap factor (distance)} = \frac{C_{overlap}}{C_{shortest\ route}} = \frac{9.50}{12.45} = 0.763$$

The overlap factor measured in free flow time is much lower than the maximum overlap factor of 0.75 and the route will be accepted. If the overlap is measured in distance, the new route will be unaccepted, because the factor is higher than 0.75. The same problem occurs for the other filters, which explain the including of irrelevant routes.

**Other causes why routes are not included**

The observed routes, which are still not included if the filters are not used, can be divided in three categories. The first group consists unfortunately of the routes that are not map matched correctly, because the centroid connection resulted in incorrect detours. These detours are below 75% overlap. This category of the routes is not interesting, because an incorrect route cannot be compared with the route within the route set.

The other two categories are more interesting. The network conditions result in several observed routes that are not included in the route set, because the link speed was too low in comparison with the actual roads. Thereby, the network conditions do not take into account that the ramp meter at the A12 is not always in action. The link speed is always low for this link, which has much influence on the free flow time of some alternative routes.

The last category routes can be explained by a slow increasing variance. Because the filters are applied afterwards, many routes are created and the variance is increasing slowly with a threshold of 5. In case of 100 iterations still many routes are not unique, which lowers the possibility to increase the variance, because the threshold is never reached. In the example of Figure 46, 43 unique routes are generated and therefore the variance only increases two times, resulting in a final variance of 0.15. This explains why the relevant observed routes with a detour of 1.4 are probably not included, because the change for this is relative low.


*Figure 46: many generated routes at the destination without filtering which prevent that the variance increases*

In general a few disadvantages are found according the route set generation.

1. Because of the filters, an accepted incorrect alternative route can never be replaced by the correct alternative route. An incorrect alternative route could have an incorrect detour, because the detour links has lower costs. Especially when the variance increases to find this alternative route, this happens easily.
2. It seems to be that the number of links in a route influences the route choice negatively. If an optimal route consist of relative long links and one of these links receives high costs, a not optimal detour will be found.

## 7.4 Conclusions

The purpose of the route set calibration was to find the optimal parameter settings for the route set generation. These optimal parameters have been found and a route set generated with these parameters includes 88.63% of the observed routes in first instance, have an average of 5.03 routes in each route set and includes the ten selected observed routes.

The routes that are not included can be divided in two categories, the routes that are excluded by the filter parameters because these were too strict and the routes that are not even included without using filters. The purpose of the filters is to exclude all routes that are irrelevant. However, this is very hard as these routes often partially overlap with routes that are relevant. This problem partly seems to occur because the free flow time is used for filtering. Especially at local roads, the free flow times are a significant part of the entire routes travel time, which results in an acceptation of routes with detour that is not useful. It is interesting to use the distance for filtering routes, because the local roads are mostly just a small part in distance of the entire route.

The exclusion of the routes, which are not even included without using filters can be attributed to three reasons. First, the centroid connection leads to route deviations in distance which do not obtain the relative weighted overlap threshold of 75%. This occurs especially for relative short routes (e.g. with a length of 10 kilometer) in combination with a start of end centroid in a rural area (e.g. the nearest centroid being 3km off). We cannot expect that these routes will be included, which makes the matching percentage of 88.63% too low. The observed routes are included for several percentages more than the 88.63% if these routes are not taken into account. A solution for the centroid connection has to be found to map match routes more correctly and with this improve the route set calibration. The other exclusions are assigned to the network conditions, these are not similar to the real conditions, e.g. speeds that are too low or are not dynamic like with the operation times of a ramp meter. At last, the variance is only increasing if no unique routes are found without applying the filters. Therefore many irrelevant routes are created and the variance stays low resulting in a low probability that routes with a higher detour are included.

The application of the Monte Carlo method proves to include many relevant routes, but shows one important drawbacks. The generation of an accepted incorrect alternative route near the correct alternative route results in the fact that this last one will be excluded by the filters. This indicates that the filtering afterwards has to be more comprehensively by removing accepted alternative routes if a better alternative route is found.

Summarizing, the comparison of observed routes with a route set generation model is very interesting to analyze the performance of the route set generator. The difficulty to include useful routes with a high free flow time detour is confirmed by this research. The used filters cannot exclude all routes that are not useful, as they would otherwise also exclude useful routes. This makes it difficult to make a good assessment. At last, the calibration indicates some interesting cases about the operation of StreamLine and Monte Carlo method in general.

## 7.5 Limitations and recommendations

The calibration of the route set was quite difficult through the many available parameters and some difficulties in StreamLine.

There are a few limitations for the found optimal parameters.

1. The parameters are calibrated for this specific data set and network and they are likely not to be optimal for other networks. Despite this, they will probably result in quite good results. It is interesting to investigate the optimal parameters for a city and a network of the entire country of the Netherlands;
2. The calibration is only performed for one mode, namely passenger cars. The weight of the routes is determined for vehicles and trucks for example will have other attributes. They won't easily make a detour on a provincial road when driving on the highway. Some traffic models use a separate model to generate the truck's routes. The calibration of other models could lead to new insights;
3. The influence of centroid connection has to be solved. The added links to the observed routes could result in illogical routes that are not likely to be included in the route set, but are used for the calibration. It may be good to mark the origin and destination of a traveler with two centroids or assign centroid to a final area. Another option is to use a network with many more centroids. Both options are likely to improve the quality of the route comparison.

Furthermore, there are a few soft and hardware limitations to calibrate the route set generation easily.

1. The fast randomizer types cannot be used, because they do not generate logical route sets. This resulted in the fact that the relative slow 'one to one' type is used and it took much time to generate route sets;
2. An overflow occurs when the variance increases too fast, therefore some parameter settings could not tested;
3. Some of the used settings (e.g. fixed seed order) were difficult to change, different installations have to be used. Besides this, some data output (e.g. reached final variance) was not available and has to be calculated manually.

At last, there are a few recommendations that could improve the route set generator in StreamLine:

1. The routes are filtered by using the free flow time as costs. It could be interesting to use the distance to filter the routes, because the spatial relations are better taken into account by using these costs;
2. The number of iterations could be lowered if a new alternative route is directly checked on all filters instead of checking afterwards. Now, only after a number of not unique routes the variance is increased;
3. The differences between the randomizer types have to be investigated;
4. It seems that the StreamLine route set generator leaks memory when performing several sequent runs;
5. The different randomizers used to generate the same route sets, but obvious differences are found in the average number of routes per route set;

# 8 Conclusions

## 8.1 Research objective

The research objective of this thesis is to calibrate the route set generator of StreamLine by map matching GPS data. Two main questions were formulated to support the accomplishment of the research objective:

1. What is the best map matching algorithm to obtain routes from GPS data and how can it be implemented?
2. What is the performance of the route set generation with the "optimal" parameter values and how to gain these values?

The two main questions results in two parts of the research. At first, the map matching algorithm has to be selected and implemented. A literature research resulted in the choice to implement Marchal's (2004) map match algorithm. After the implementation, the algorithm's parameters are calibrated by using a part of the GPS data. The optimal parameters are used to map match the entire set of GPS data. The map matched routes have been checked on some criteria to be sure that they are relevant in the calibration of the route set generation parameters.

The calibration of route set generation is divided in two parts. At first, the filter parameters are determined by analyzing the GPS data. After this, the other parameters are calibrated by using a distance measure to determine whether the observed routes are included in the generated route sets. Thereby two additional criteria are applied that have to be satisfied by the generated route sets.

## 8.2 Conclusions

### *8.2.1 Map matching*

The map matching algorithm of Marchal (2004) works well for the used dataset and the network. The method to keep paths, which are likely to be correct, in the memory is very essential to obtain good map matching results. The algorithm map matches 89% of the routes correctly with a computation speed 450 times faster than the collection time (450 GPS points/s). This research found out that parameter values used by Marchal are not optimal for our case, because they could result in an incorrect score determination when the GPS point are positioned too far behind the matched links. The found parameter values, shown in table below, result in a high match percentage with the lowest deviation while maintaining an acceptable computation time.

| Parameter | Value |
|---|---|
| α | 0.9 |
| Path size | 15 |

*Table 19: optimal parameter values*

The selection of relevant observed routes for route set calibration is partly influenced by the connection to centroids. Especially when the centroids are located far from the last GPS location, the connection to centroids creates unlikely chosen routes. Thereby, it is important to realize that it is only relevant to find small errors at roads, which influence the route choices on the roads that are interesting for traffic research. This is the case for road works, short stops and network deviations.

### *8.2.2 Calibrating the route set generation*

The relevant observed routes are used to calibrate the route set generation. The observed routes are easy to use to determine the values for the filter parameters. By using these routes, we are sure that the observed routes are not removed by the filter parameters self. The table below shows the determined filter parameter values.

| Filter | Value |
|---|---|
| Maximum detour | 1.6 |
| Maximum total non-common detour | 2.1 |
| Minimum non-common detour | 0.10 |
| Maximum overlap | 0.75 |

*Table 20: filter parameters with determined values*

The comparison of observed routes with the generated route sets shows the drawbacks of using the travel time as filter costs. The free flow time is relative low at local roads in relation to the highways. Therefore, irrelevant routes on local roads are simple accepted because the overlap is less than the maximum filter value through the high free flow time on the irrelevant roads.

The actual route set generation calibration resulted in a match of 88.63% of the observed routes, which are included in the generated route sets. The generated route sets have an average of 5.03 routes per route set.

It is difficult to include routes with a high detour in free flow time; approximately only half of the observed routes with a detour of more than 1.25 is included in the generated route sets. The actual matching percentage is a several percent higher, because the centroid connection has lead to several observed routes that were not likely to be included. The found parameters for the route set generation in StreamLine are shown in the table below.

| Parameter | Value |
|---|---|
| Route generation | |
| Variance grow | 0.05 |
| Threshold | 5 |
| Threshold consecutive | true |
| Iterations | 90 |

| Stop criterion | |
|---|---|
| Maximum variance | 0.30 |
| Maximum routes | 7 |

*Table 21: route set generation parameter values*

A distance measure is used to determine whether an observed route is similar to a route in the generated route sets. Unfortunately, it is difficult to set the threshold value for the overlap correctly. We set the threshold value for the weighted overlap value to 75%, but this value does not guarantee that all relevant routes are accepted and irrelevant routes are not accepted.

There are four reasons found why observed routes are not included in the generated route sets. At first, several observed routes are filtered by two route set generation filters (overlap and minimum not common detour) and by the stopping criteria (the maximum number of routes and the maximum variance). These values cannot be adapted, because otherwise too many irrelevant routes will be accepted. The second reason is the dynamic network conditions of the ramp meter at the A12. The model network assumes a very low speed for this link all the time, but actually the ramp meter is only working during the rush hours. This results in many generated routes, which avoid this on-ramp, although the observed routes do include the on-ramp because it was not working. At last, the variance increases too slowly, because only in case of sequent not unique routes without using filters, the threshold value will be obtained. There are many unique routes that would be filtered that it takes much time before this happens.

### *8.2.3 Further research*

Nowadays, little research is performed on the calibration of route set generation by using map matched routes. Because of this, several issues have been identified for further research.

a) The influence of centroid connection is large. An approach to connect observed routes more logical to centroids by a new method is needed. Another option could be the addition of centroids at the start or end of an observed route. Solutions like this could greatly increase the quality of the route set calibration;
b) The quality and runtime of the map matching algorithm could be improved by using more intelligence about the network. If the GPS points are correctly matched to a highway link several times in a row, it is quite certain that this path is the best, but meanwhile all other paths are still investigated and extended at each junction. This causes many irrelevant computations. It is interesting to assign some route types for which a minimum number of correct matched GPS point could remove all other paths.
c) The free flow time as cost for the route set generation filters do not remove all irrelevant routes. It would be interesting to replace the free flow time with the distance, which means that the different routes on local roads are less often accepted.
d) The route set generation might be optimized when the threshold consecutive parameter is set on false. This will result in a faster increase of the variance and could result in fewer computations.
e) Even through the Monte Carlo method generates quite good route sets, they are not optimal, because relevant routes are excluded and irrelevant routes are in the set. It seems to be that this drawback is generally accepted. The influence of the route set generator on the results of the traffic model should therefore be investigated;

f) GPS data is more often available and consists of valuable traffic data. It is interesting to investigate the other possibilities of these data, like for example the calibration of route choice or junction modeling. The data could also be used to generate networks or find network errors and to do research on travel times, departure times, bottlenecks and short cuts.

# References

Bierlaire, M., and Frejinger, E. (2008). Route choice modeling with network-free data, Transportation Research Part C: Emerging Technologies 16(2), 187-198.

Bliemer, M.C.J., PHL Bovy & H. Li (2007). Some properties and implications of stochastically generated route choice sets. Proceedings TRISTAN VI Conference 2007, Phuket, Thailand

Bovy, P.H.L. & Stern, E. (1990). Wayfinding in Transport Networks. Dordrecht: Kluwer Academic Publishers.

Brakatsoulas, S., Pfoser, D., Salas, R., and Wenk, C. (2005). On Map-Matching Vehicle Tracking Data. *Proceedings 31st VLDB Conferenc*e, 853-864.

Dial, R.B. (1971). A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research 5*, *2*, 83–111.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, S. 269–271.

Fiorenzo-Catalano, M.S. (2007). Choice set generation in multi-modal transportation networks. Delft: TRAIL Research School / Delft University of Technology.

Frejinger, E. and Bierlaire, M. (2007). Capturing correlation with subnetworks in route choice models. Transportation Research Part B 41 (3), 363–378.

Floyd, Robert W. (June 1962). "Algorithm 97: Shortest Path". Communications of the ACM 5 (6): 345.

Hart, P.E., Nilsson, N.J., Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2): 100–107.

Hoogendoorn-Lanser S. (2005). Modelling travel behaviour for multi-modal transport networks, TRAIL Thesis Series T2005/4, TRAIL, The Netherlands.

Jan O., Horowitz A. and Peng Z. (2000). Using GPS data to understand variations in path choice. *Transportation Research Record, 1706*, 145–151.

Kant, P. (2008). Route choice modelling in dynamic traffic assignment. *Master thesis.*

Marchal, F., Hackney, J., Axhausen, K.W. (2004). Efficient map matching of large global positioning system data sets: Tests on speed monitoring experiment in Zürich. *Transportation Research Record, 1935*, 93–100.

OmniTRANS International (2008). *StreamLine Framework, Technical Design Document.* Deventer: Rsm.

Ortúzar, J. de D. and Willumsen, L.G. (2001). Modelling Transport, *third ed.*, John Wiley & Sons, Chichester.

Quddus, M.A., Noland, R.B., Ochieng, W.Y. (2007). Current map-matching next term algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies, 15,* 312-328.

Rich, J., Mabit, S., Nielsen, O.A. (2007). Route choice model for Copenhagen: a data-driven choice set generation approach based on GPS data. In: Proceedings of the 6th Tristan Conference, Phuket, Thailand.

Taylor, G., Brunsdon, C., Li, J., Olden, A., Steup, D., Winter, M. (2006). GPS accuracy estimation using map-matching techniques: Applied to vehicle positioning and odometer calibration. *Computers, Environments, and Urban Systems, 30,* 757–772.

Yin H. & Ouri W. (2004). A Weight-based map matching method in Moving Objects Databases. *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM), June,* Santorini Island, Greece, 21-23.

Yin, H., Wolfson, O. (2004). A weight-based map-matching method in moving objects databases, Scientific and Statistical Database Management. *Proceedings of the International Working Conference, 16,* 437–438.

Zantema, J. Amelsfort, D.H. van Bliemer, M.C.J. Bovy, P.H.L. & Bussche, D. (2007). Route Set Generation Within a Dynamic Modeling Framework: A Comparison of Methods and GPS Route Choice Data. *Proceedings of the European Transport Conference ETC, October,* Noordwijkerhout, The Netherlands.

# Appendices

## Appendix A

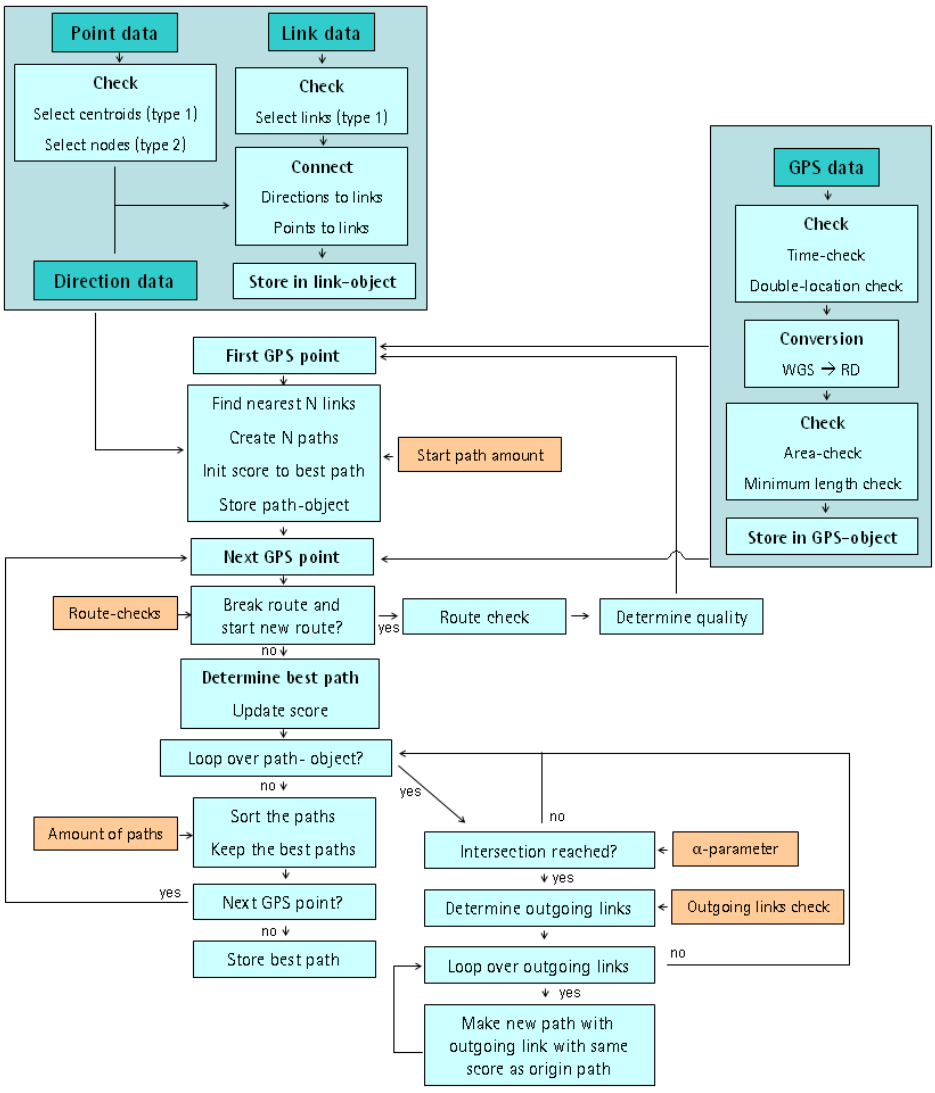



*Figure 47: Flow chart of the map matching algorithm*

The flow chart above supplies a global view on the map matching algorithm in OmniTRANS. There are two separated parts that are not described in the literature review. The first one is the GPS data check that decreases the GPS data file size. This is done by a simple Matlab algorithm performing a user, time and double location check. The remaining GPS points are separated in an array of GPS points, in which each route array is checked on several criteria (e.g. minimum length and a "courier" check) as described in the map matching chapter. The second part describes the generation of a link-object. The OmniTRANS network consists of points, links and directions. This data is for each link aggregated into a link-object as feature of the object-oriented programming in Ruby. This object is used to find the nearest links quickly during the performance of the map matching algorithm.

## Appendix B

Not all routes are relevant for the calibration of the route set generation. Some routes are not accepted because of one important reason. These routes include a "strange" detour in comparison with the direct route. This detour is not accountable, but has an extern reason like dropping the kids at school or road workings.

The route check script is developed to check routes in the OmniTRANS network quickly. The script is self-learning, which means that frequently occurred errors can be changed once. The script consist of some toolboxes to make choices. The script shows the first route of the route set and a "accept"-toolbox with two choices, "yes" and "no". The "yes" button means the route is accepted and the next route is showed. When the user selects "no", a new "change"-toolbox appears with two choices "yes" and "no". The "change" option supplies the possibility to change the route and accept this route. After the route is changed, the user can perform this change for all coming routes. Naturally, the "no" button do not accept the route. All checked routes are marked to prevent double checks when the script is used several times for the same route set. The script's chart flow is showed below.

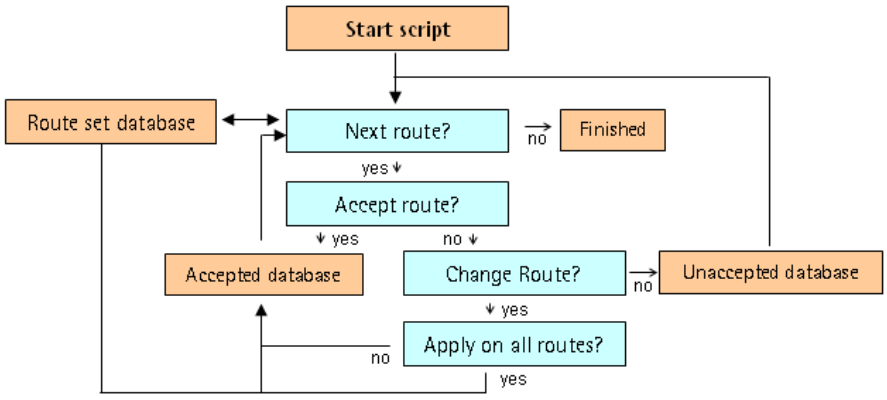



*Figure 48: Flow chart of check-script*

## Appendix C

A low value for the α-parameter results in incorrect map matching results. Marchal (2004) used a α-parameter of 0.5 to select new links at a junction, because u-turns are more available in foreign countries. The following example will show the reversal of a low α-parameter.

The two figure below show the route chosen by the driver and some GPS points along his route numbered from $P_1$ till $P_{21}$.

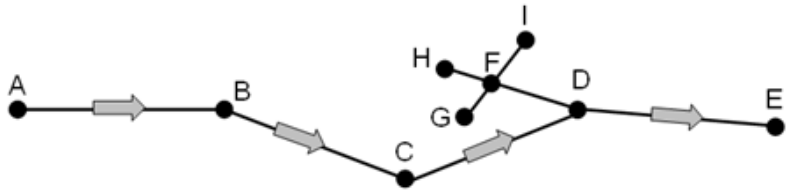



*Figure 49: Chosen route by the driver*

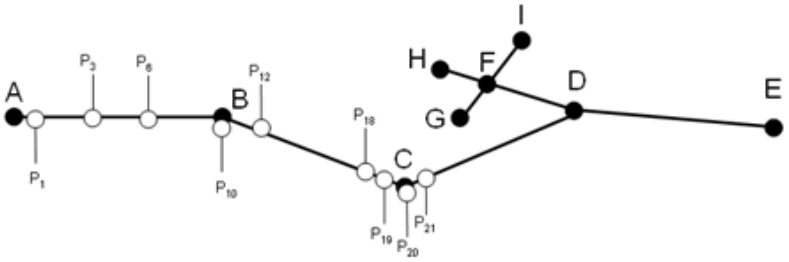



*Figure 50: GPS points among the route*

A value of 0.6 is assumed in this example. This means that at 60% of the link length new links are selected that are topological connected with the junction. After three links of 1km, the virtual vehicle is at 0.6*3 = 1.8km of the link length, but already 3km of links is map matched. The fourth link is still receiving its score from the car's position. The map matching algorithm will result in an incorrect map matched route as shown in the figure below, because the score of fourth incorrect selected link (D-F) increases. The position of the vehicle is closer to this link than the correct link, because the vehicle is located too far back.

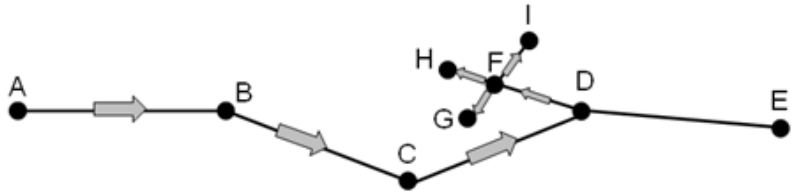



*Figure 51: Incorrect map matched route*

The following checks show the determination of the scores at proves that the link selection is going incorrect.

**Check 1**
Selected GPS point: P3

- Link set 1: A-B
  Length A-B: 1000m
  P1 - P3: 300m
  Score: 3

**Check 2**
Selected GPS point: P6

- Link set 2: A-B (+ B-C)
  Length A-B: 1000m

P1 - P6: 600m (new link added) Score: 6
→ 1000*0.6=600m

**Check 3**
Selected GPS point: P12

- Link set 3: A-B, B-C (+ C-D)
  Length B-C: 1000m
  P6 - P12: 600m (new link added) → 1000*0.6=600m
  Score: 12

**Check 4**
Selected GPS point: P18

- Link set 4: A-B, B-C, C-D (+ D-F)
  Length C-D: 1000m
  12 - P18: 600m (new link added) → 1000*0.6=600m
  Score: 18
- Link set 5: A-B, B-C, C-D (+ D-E)
  Length C-D: 1000m
  P12 - P18: 600m (new link added) → 1000*0.6=600m
  Score: 18

**Check 5**
Selected GPS point: P19

- Link set 4: A-B, B-C, C-D, D-F
  Length D-F: 400m
  P18-P19: 100m
  Score: 19
- Link set 5: A-B, B-C, C-D, D-E
  Length D-E: 1000m
  P18-P19: 100m
  Score: 18

**Check 6**
Selected GPS point: P21

- Link set 6: A-B, B-C, C-D, D-F (+ F-H)
  Length D-F: 400m
  P18-P21: 300m (new link added) → 400*0.6=240m
  Score: 21
- Link set 7: A-B, B-C, C-D, D-F (+ F-G)
  Length D-F: 400m
  P18-P21: 300m (new link added) → 400*0.6=240m
  Score: 21
- Link set 8: A-B, B-C, C-D, D-F (+ F-I)
  Length D-F: 400m
  P18-P21: 300m (new link added) → 400*0.6=240m
  Score: 21
- Link set 5: A-B, B-C, C-D, D-E (maximum of three paths → correct link set will be deleted)
  Length D-F: 1000m
  P18-P21: 300m
  Score: -1