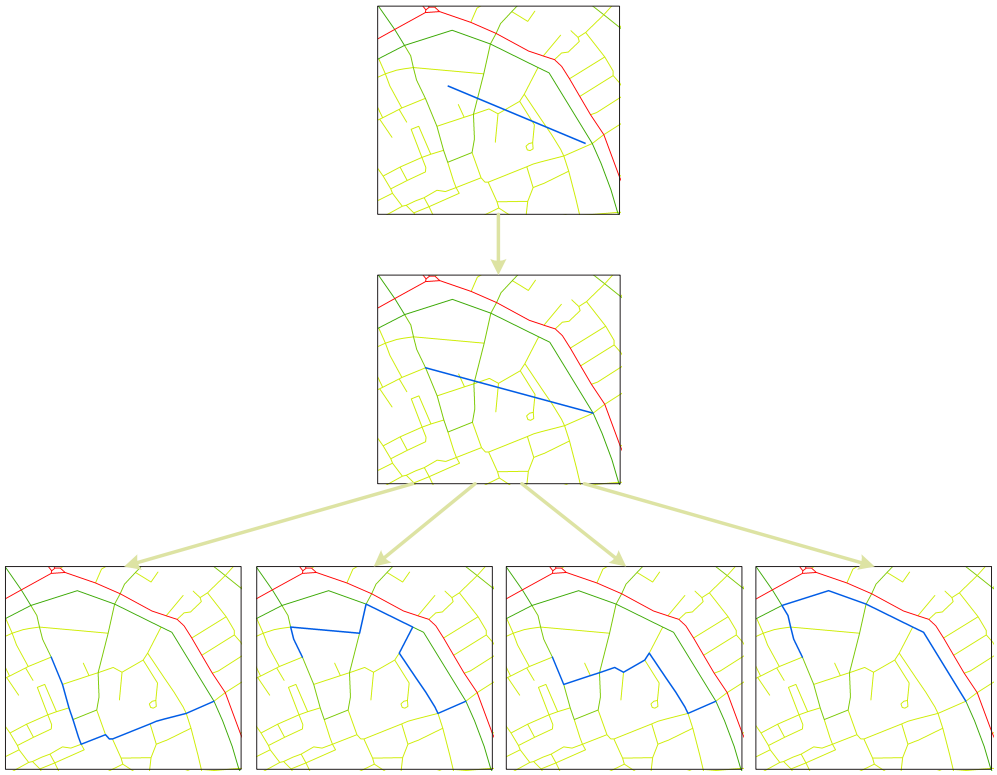
Geographic data integration for telecommunication purposes



Delft University of Technology

T.E. Bont

TNO Information and Communication Technology

Delft, 2006

Geographic data integration for telecommunication purposes

Master's Thesis in Geomatics

University:
Delft University of Technology
Section GIS technology
Jaffalaan 9
PO Box 5030
2600 GA Delft

Supervisors:
ir. E. Verbree
prof. dr. ir. P.J.M. van Oosterom

Company:
TNO Information and Communication Technology
Department Planning, Performance & Quality
Brassersplein 2
PO Box 5050
2600 GB Delft

Supervisor:
drs. A.L.J.M. Wiegerinck

T.E. Bont

Delft
June 30, 2006

Preface

This graduation thesis has been executed at TNO ICT. The focus of the department Planning, Performance & Quality is to optimize the balance between costs and quality of ICT networks and services.

In order to improve the quality of mobile telephone networks, the TIGER (Traffic Intelligence through Geographic Extrapolation Results) tool has been developed. This tool provides spatial traffic grids which indicate the distribution of the density of the calls that could be made in an area. With these grids, a planning tool can (re)design the mobile telephone network.

The quality of the output of the TIGER tool can be improved by optimization of the input data. One of the input data consist of the distribution of the potential users of the network, which are traveling by car or train. This input data set is not detailed, and has to be matched with a detailed road and railroad data set.

This thesis gives an overview of methods for matching the different representations of two independent street data sets, and implements several of these methods. The quality of the results have been evaluated by a manual comparison.

Special gratitude for Waldo Bont, Otto Visser and Roy van der Bol for their available time for reading this thesis and the interesting discussions.

Summary

Planning and designing mobile telephone networks is a complicated process, which involves large amounts of money. The TIGER tool is under development at TNO ICT. The tool calculates the spatial traffic grids, which are one of the input data sets for mobile telephone network planning tools.

The input data for the TIGER tool consists of the distribution of the potential users of the mobile telephone network. The distribution of the traveling users is only available at low detail and this data set has to be matched with the detailed road and railroad data set.

The match of both data sets should be optimized and an indication of the reliability and accuracy of the matches has to be given. In order to perform this task properly the following research question has been formulated:

> *Find the best match between generalized, low detailed schematic links and the more detailed road and railroad network and give an indication of reliability and accuracy of the results.*

This graduation project has been executed in two ways. First an examination of related researches on the topic of matching multiple representations of street data sets has been performed. Second, different variants of solution definitions have been implemented in . The results have finally been compared in order to give an indication of the reliability and accuracy.

This best variant has the highest matching rate of all of the implemented variants, determined by comparing the results of the variants mutually. The variant generalizes the matching route of the travelers with the Douglas-Peucker line simplification algorithm. The number of vertices that are left give an indication of the smoothness of the route, which is one of the characteristics of the input data set containing the distribution of the travelers.

Samenvatting

Het plannen en ontwerpen van mobiele telefoon netwerken is een complex en kostbaar proces. De TIGER tool wordt ontwikkeld door TNO ICT. De tool berekent de *spatial traffic grids*, die een van de input data sets vormt voor de planning tools voor het mobiele telefoon netwerk.

De input data voor de TIGER tool bestaat uit de verdeling van de mogelijke gebruikers van het mobiele netwerk. De verdeling van trein en autoreizigers is slechts beschikbaar met weinig detail. Om deze input data te verbeteren, moet deze huidige data set worden gekoppeld met een gedetailleerd (spoor)wegen netwerk.

De koppeling van beide data sets moet zo optimaal mogelijk worden uitgevoerd en een indicatie van de betrouwbaarheid en de nauwkeurigheid van de resultaten wordt gegeven. Om deze taak uit te voeren is de volgende onderzoeksvraag geformuleerd:

> *Vind de beste match tussen de schematic links, welke weinig detail hebben en het fysieke wegen netwerk, met veel detail, en geef een indicatie van de betrouwbaarheid en nauwkeurigheid van de resultaten.*

Dit afstudeeronderzoek is uitgevoerd op twee manieren. Eerst is een aantal soortgelijke onderzoeken bestudeerd op het onderwerp van het koppelen van verschillende afbeeldingen van wegen data sets. Ten tweede zijn verschillende varianten voor oplossingen geïmplementeerd in ArcGIS. De resultaten zijn uiteindelijk vergeleken met elkaar en met een handmatig gedefinieerde *ideale route* om een indicatie te kunnen geven van de betrouwbaarheid en nauwkeurigheid van de varianten.

De beste variant wordt bepaald doordat deze het vaakst de juiste kandidaat route selecteerde wanneer de resultaten van deze variant werden vergeleken met de andere varianten. Deze variant generaliseerd de kandidaat route met behulp van het lijn vereenvoudigingsalgorithme van Douglas-Peucker. Het aantal tussenpunten van de route die overblijven, geeft een indicatie van de gladheid van de route. De gladheid van de route is een van de eigenschappen van de input data, welke alleen de reizigers van de hoofdwegen bevat, die bestaan uit weinig bochten en afslagen.

Contents

CHAPTER 1

Introduction

The planning and redesign of mobile networks for GSM and UMTS can be executed by erecting new antennas, adjustment of the frequency distribution, aiming the directional antennas, etc. In order to design the mobile networks efficiently, e.g. to avoid unnecessary re-erecting of antennas, several planning tools have been built.

Planning of the mobile telephone network is performed based on a propagation model of the radio signal, the current network, and maps containing the distribution of potential users of the mobile telephone network. The TIGER (Traffic Intelligence through Geographic Extrapolation Results) tool develops spatial traffic grids which consists of the density of the calls. These grids are based on the distribution of potential users of the network. One of these grids contains potential users which are traveling along roads or railroads. The data containing these travelers is a highly generalized network.

The input of TIGER can be improved by matching the generalized network with a detailed road and railroad data set. Through matching both data sets, the location of the potential users of the mobile telephone network can be determined more precise, and better design decisions can be made.

In chapter 2 a short overview of the historical development of mobile telephones and the networks is given. Also the configuration of the networks will be described, combined with how a planning tool can (re)design the network. Further, the place of TIGER in this process will be examined. Chapter 3 describes the problem that has been investigated during this graduation thesis in more detail. The research questions are formulated in this chapter to lead the research in the right directions.

Chapter 4 contains a description of the available data sets and the programming environment (ArcGIS) which will be used to implement the programme in. These data sets can cause some limitations, which can affect the results of this research. The limitations or possible improvements of the input data will also be described.

This thesis is in line with some other researches on matching multiple representations of street data and are investigated in chapter 5. With these researches in mind, combined with other algorithms, different algorithms are defined in chapter 6. The variants which are implemented for determining which route is the best match to the generalized network segments are defined in this chapter.

In chapter 7 the implementation algorithm in ArcGIS is described. These results are described in chapter 8, which will contain a comparison of the results with the ideal routes (defined by a human interpretation of the data). Finally, chapter 9 will give the conclusions

of this graduation thesis and the recommendations for further research.

CHAPTER 2

Mobile telephone network

This graduation thesis has been performed to improve the input data for the TIGER (Traffic Intelligence through Geographic Extrapolation Results) tool which determines the spatial traffic grids for mobile telephone network tools. With the improved input data, a better prediction of the distribution of the potential users of the mobile network will be available.

To get a better understanding of the scope of this thesis, first a short description of mobile telephone networks will be provided. In section 2.1 a brief overview of the development of the mobile telephone and its networks will be given. Section 2.2 describes the configuration of mobile networks with its components. In section 2.3 the process of planning and redesigning a mobile telephone network is explained. Finally, in section 2.4 the TIGER tool will be examined by its input and output data sets.

2.1 History

In this section a short overview will be given of the development of the mobile telephones and mobile telephone networks:

1950 The key foundations of wireless mobile systems were invented. The concept is to reuse the same limited radio frequency in a group of cells arranged in a cellular structure to serve an unlimited numbers of users. The first generation cellular wireless mobile systems were based on analog transmission (Bi et al., 2001).

1982 Twenty-six European national phone companies began developing GSM, a new technology in a new radio band, a uniform, European wide cellular system around 900 MHz. Planning began in earnest and continued for several years (Farley, 2005).

1985 The 1G (first generation) mobile telephone networks are first introduced. Most mobile (analog) phones are large and therefore often permanently installed in vehicles as car phones (see figure 2.1).

In 1985, the Netherlands have a nationwide coverage using 50 cell sites with two different cell sizes, 5 and 20 km radius. The capacity of this system is 15.000 to 20.000 subscribers (Lee, 1995).

1990 In Europe, the idea of a 3rd generation mobile system called UMTS (Universal Mobile Telecommunications System), was developed 1990s through several European Union funded research projects.

Figure 2.1: Old mobile cell phone

3G is a wireless industry term for a collection of international standards and technologies aimed at increasing efficiency and improving the performance of mobile wireless networks. The services associated with 3G provide the ability to transfer both voice data (a telephone call) and non-voice data (such as downloading information, exchanging e-mail, and instant messaging), from Farley, 2005.

1991 Commercial GSM network (the second generation) starts operating. Every mobile contains or accesses encryption to prevent eavesdropping, authentication to prevent fraud, short messaging services or SMS, and a SIM card to easily add accounts to a handset.

2004 The first launch of the third generation (3G) of mobile systems in Great Britain.

It is announced that GSM has one billion customers.

2006 The Netherlands are covered by approximately 18.000 cell sites, serving about 16 million users.

After this global overview of the development of mobile telecommunication, the principles of the networks that are used will be further investigated.

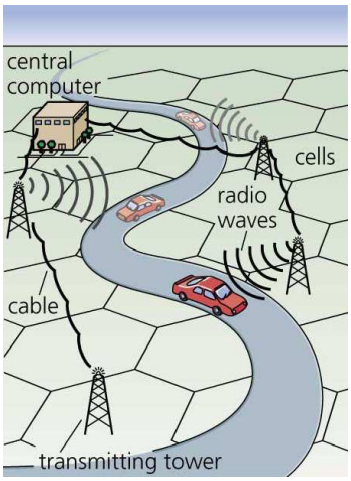


Figure 2.2: Configuration of mobile networks

2.2 Configuration

Most current mobile phones connect to a cellular network of base stations (*cell sites*). A cell in the Netherlands has typically a radius of about 250 m in urban areas, and up to 10 km in rural areas. A cellular communications system employs a large number of low-power wireless transmitters (antennas) to create the cells. A cell site is interconnected to the public switched telephone network (PSTN). With the advance of miniaturization, currently the vast majority of mobile phones are hand-held.

Mobile telephones, and the network they operate under, vary significantly from provider to provider, and even from nation to nation. However, all of them communicate through electromagnetic radio waves with a cell site base station, the antennas of which are usually mounted on a tower, pole, or building (see figure 2.2).

2.3 Planning and redesigning

In the current practice, many complex simulations and data measurements are necessary at the early stage of network planning and design to decide where to locate the base stations. Figure 2.3 gives an overview of a planning process for (re)location of base stations of the mobile telephone network. This process is executed with the following input data sets:

- *Clutter database* – contains the properties of the surface of the network area, like the building density, the heights of the buildings, presence of vegetation, etc. These properties influence the radio signal propagation, and thus the range of an antenna. The clutter database is input for the *propagation model*.
- *Propagation model* – this model calculates maps with the signal strength of the antennas for each pixel. Input for this model are the transmitting power of the

antennas, the distance of the pixel to the antenna, and possible obstacles in the terrain (from the clutter database). This model is necessary, because the stronger the signal, the more likely a mobile telephone is connected to the antenna in question.

- *Current network configuration* – the current configuration will not be totally modified, due to the expenses which are involved to do this. Therefore, for placing new antennas or aiming the antennas, it is necessary to know the location of the current cell sites and the aim of the antenna.
- *Spatial traffic grid* – this grid contains the distribution of the amount of mobile calls (in minutes per pixel). One way of designing the spatial traffic grid is by using the TIGER tool of TNO ICT. TIGER calculates the distribution over users types (e.g. citizens, car and train travelers, employees, etc.) by using the number of potential customers at a location. The TIGER tool is further examined in section 2.4.

In order to improve the mobile telephone network, these input data sets will be used in a planning tool. This tool calculates a new configuration of the antennas in the network. This thesis will investigate one of the input data sets of TIGER, the distribution of car and train travelers. Therefore, first the TIGER tool will be further examined in the next section.

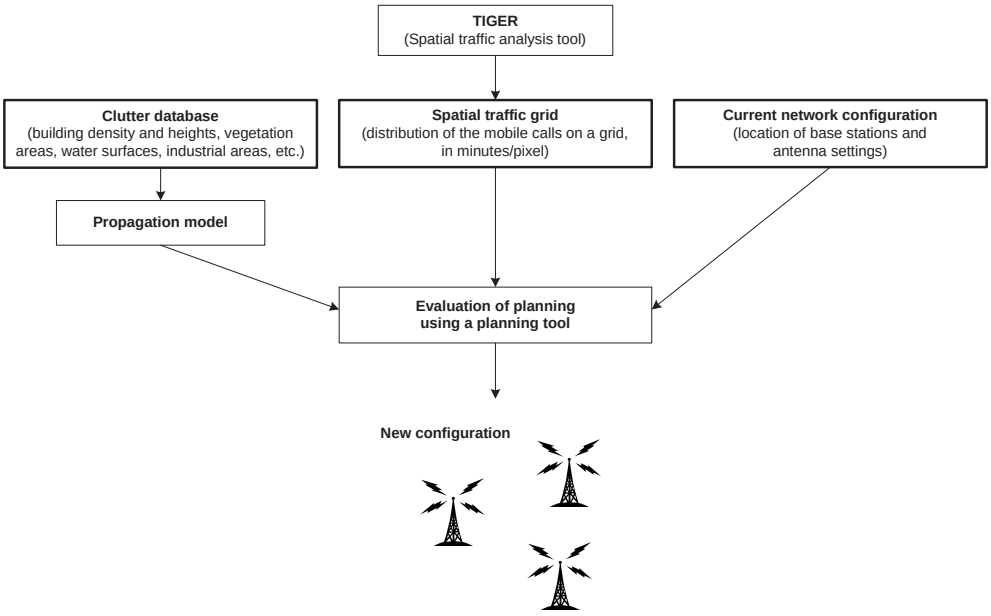


Figure 2.3: Overview of the inputs in the planning process

2.4 TIGER

Figure 2.4: TIGER logo

TIGER (Traffic Intelligence through Geographic Extrapolation Results) is a tool for operators of mobile telephone networks, for reducing the number of required base stations, and maintaining or improving the quality of the network, at minimal costs. The development of TIGER is started by KPN Research and continued at TNO ICT (figure 2.4). More information on the company TNO ICT, where this research has been executed, can be found in appendix A.

The process of TIGER is shown in figure 2.5. In order to calculate the spatial traffic grids, TIGER uses the following input data sets:

- *Traffic per base station (minutes)* – the current amount of calls (in minutes) which are handled by each base station.
- *Propagation predictions* – the signal strengths per pixel from the propagation model.
- *Demographic and vector data* – per pixel the potential users of the network are determined.

Based on a probability model a prediction is made with which base station a mobile telephone call will be connected. These probabilities are determined based on the signal strengths of the antennas from the *propagation model*. For each pixel the number of potential users has been determined, based on the *demographic and vector data*. These potential users are allocated to the base stations according to the probability model. This way, for each base station the numbers of potential users, and the measured, current traffic (the input data set) are determined.

By means of a statistical model, the numbers of potential users are translated to the amount of telephone traffic per potential user (based on which user groups the potential users belong to). Finally, this result is translated back to the location of the users, on the pixels. The resulting traffic grid is the output data set of TIGER, and input of a planning tool, which performs the planning process of the radio networks for GSM and UMTS.

This graduation research will improve the input data sets of TIGER containing the *Demographic and vector data*. With the improved input data set a better planning and redesign of mobile networks has to be accomplished. In the next chapter the research problem of this thesis is described. The background information provided in this chapter has given a indication of the scope of this project.

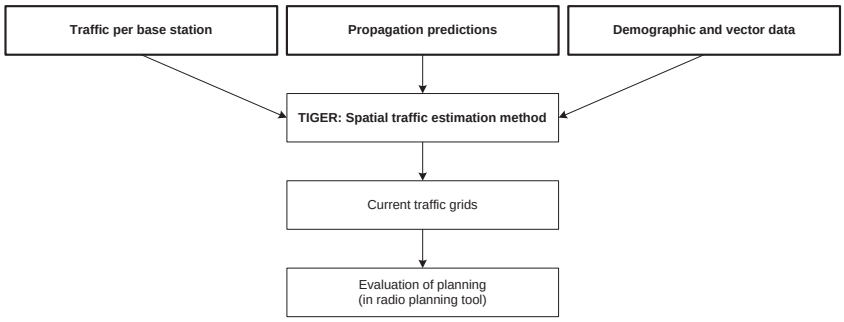


Figure 2.5: The spatial traffic estimation process

CHAPTER 3

Problem description

The context of this research has been described in chapter 2. The purpose of improving one of the input data sets of the TIGER tool, and therefore the objective of this research, is described in section 3.1. The research questions will be formulated in section 3.2.

3.1 Objective

One of the input data sets of TIGER contains the distribution of the potential mobile telephone users. This distribution has already been determined for residents and employees based on the zip codes. The distribution of road and railroad travelers is more difficult to be determined and is only inaccurately available.

The roads and railroads, where the amount of traffic is known, are approximated by straight lines. From now on these straight lines will be indicated by *schematic links*, as only the starting and endpoints of the roads and railroads are stored. In order to improve the determination of the distribution of the travelers, the location of the roads has to be defined more precise. The information for this improvement can be extracted from other map data sets.

By matching the coordinate pairs at the junctions of real roads and railroads, and to determine the route in between these junctions, which is most likely to be traveled by the cars and trains, the distribution of the travelers in the real geographic space can be determined.

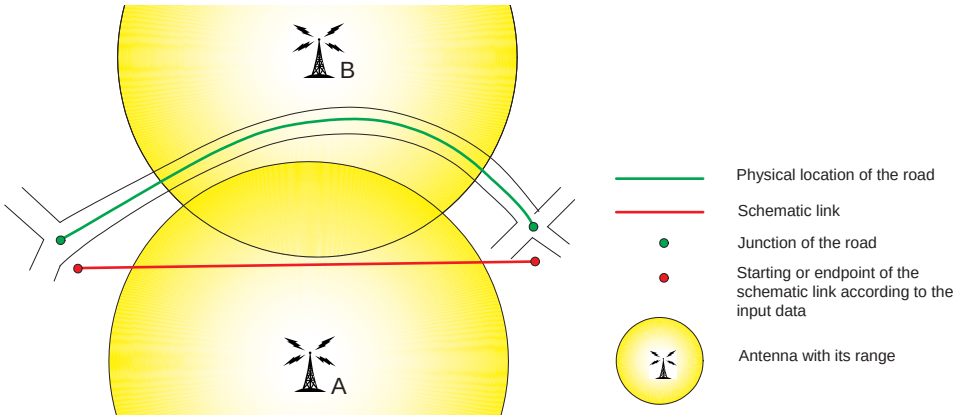


Figure 3.1: Schematic overview of the problem

In figure 3.1 a schematic overview of the problem has been sketched. In this figure the situation is shown where it will make a difference where the travelers are located in the model. In this figure, the schematic link is represented by the red line. According to this situation, the travelers on the road will be assigned to antenna *A*. In reality the travelers will travel along the road, represented by the green line.

The travelers on the road will use antenna *B*, which isn't considered in the planning and redesign of the network. Too large capacity of antenna *A*, and too small capacity of antenna *B* can cause breaking down of the connection during a call, or a bad coverage of the mobile network. Therefore, it is important to determine the location of the potential users of the mobile telephone network as accurate as possible.

3.2 Research questions

After this description of the problem, the approach to solve this problem by formulating the research questions.

Research question

In this thesis the determination of the distribution of the car and train travelers, at the true locations on the (rail)roads is examined. To determine this distribution, the following research question is formulated:

> *Find the best match between generalized schematic links and the more detailed road and railroad network and give an indication of reliability and accuracy of the results.*

The schematic links give an indication of some connection between two junctions in the road network. In the detailed road network it is possible to detect more than one route between these junctions. Therefore, a selection has to be made to choose the most likely route that is intended with the schematic link.

In some cases it is even for human interpreters difficult to select the best match. Although no ideal situation is available, an attempt will be made to give an indication of the reliability and accuracy of the results.

Focus questions

Given the research question, the following focus questions will guide the execution of this graduation thesis:

1. *Which data sets are available for the implementation of the method?*
   Which data sets (raster and vector), programmes, literature, tools and extensions from ArcGIS, etc. can be useful for the implementation of the method. This focus question will be treated in chapter 4.

2. *Which algorithms can be defined to match the schematic links with the road and railroad data sets?*
   Different algorithms can be defined to match the schematic links with the road and railroad network and to finally match the amount of travelers to this road and railroad data. The true location of the roads and railroads can be extracted from vector or raster based data. From these algorithms one procedure will be selected and implemented in ArcGIS.

There are several researches on matching multiple representations of street data sets, which have similarity to this thesis project. The results of these researches are described in chapter 5. Chapter 6 examines the theory on which the implemented procedure is based.

Based on the combination of chapter 5 and 6 different variants for matching the schematic links to the road and railroad network are defined and implemented in ArcGIS in chapter 7.

3. *What are the reliability, accuracy, and other characteristics of the implemented variants?*
   By evaluating the results from the implemented variants, an indication of the reliability and accuracy will be given. This evaluation will be performed by comparing some of the results to each other and by comparing the results of each variant with the *ideal* route. The ideal route has been determined by human interpretation of some of the schematic links. These evaluations are described in chapter 8.

With the research questions that have been formulated in this chapter, the research for improving one of the input data sets of TIGER can be executed in the following chapters. In the next chapter the preconditions of this research will be described, consisting of the input data sets and a description of the programming environment, ArcGIS.

CHAPTER 4

Preconditions

With the research questions formulated in the previous chapter, the preconditions of this project will be described in this chapter. First the input data sets will be examined in section 4.1. In this section data sets and their limitations will be described. Also a description will be given of the information that ideal data sets could contain and the available data sets did not contain, in order to determine the distribution of the travelers more correctly.

The programming environment (ArcGIS) is described in section 4.2. This environment uses special data sets and ways for storing the geographical data. The possible input data sets for ArcGIS will be described. ArcGIS also consists of different components, the one with more functionalities than the other, and different extensions. The differences of these components and the extensions which could be interesting for this thesis will be described too.

4.1 Input data

In order to be able to give an answer to the first focus question (which data sets are available), the available data will be described in this section . For future use of the developed programme it is not necessary to posses exactly the same structured data, because the structure of the data can be specified on the form of the programme, as will be described in the introduction of chapter 7.

4.1.1 General traffic data sets

The basis of this project are two data sets, one containing the information of the road travelers (figure 4.1) and the other containing the information of the railroad travelers (figure 4.2). The amount of traffic on the roads has been determined by means of traffic counting during the rush hours. This data has been based on the basic matrix of 2003 of the *Dutch National Model for Traffic and Transport*[1] and the counts of the amount of traffic on road segments, by the Dutch Transport Research Centre (part of the Rijkswaterstaat organization)[2]. The train travelers are expressed by the relative amount of traffic in the trains, and has been determined by *Netherlands Railways, Department of Commerce* in 2003.

With these data the distribution of the potential mobile telephone users traveling along the roads and railroads has to be determined. From these data sets the schematic links

[1] Landelijk Model Systeem verkeer en vervoer (LMS)
[2] Adviesdienst Verkeer en Vervoer van Rijkswaterstaat (AVV)

	A	B	C	D	E	F	G
1	Knoop 1	Knoop 2	X1	Y1	X2	Y2	Count
2	1	22520	264490	554200	264230	553990	2322
3	1	22634	264490	554200	265950	553250	0
4	2	22019	256950	568950	256060	569110	1682
5	2	22129	256950	568950	256970	568460	480
6	3	22646	266330	575350	266200	575100	0
7	3	22651	266330	575350	266470	575410	1730
8	4	21728	253460	594570	253300	594250	762
9	4	21764	253460	594570	253590	594290	1062
10	5	20962	240130	602640	239820	603270	318
11	5	21020	240130	602640	240960	603370	598
12	6	19891	234540	591750	231330	591380	176
13	6	20490	234540	591750	234730	595070	0
14	6	20560	234540	591750	235250	587550	2930
15	6	20919	234540	591750	239250	591400	16
16	7	21369	248570	577090	246560	577050	3218
17	7	21504	248570	577090	248570	580690	176
18	7	21767	248570	577090	253600	577030	0
19	8	20739	237170	576100	236850	575990	1846
20	8	20793	237170	576100	237340	575730	1168
21	9	20134	233790	582250	233150	581560	5236
22	9	20182	233790	582250	233390	582180	0

Figure 4.1: Example of the road data table

	A	B	C	D	E	F
1	BAANVAKID	X1	Y1	X2	Y2	Drukte
2	Vs-Vss	30450	385450	30500	387700	4,36
3	Vss-Mdb	30500	387700	32150	391050	5,43
4	Mdb-Arn	32150	391050	35750	391700	10,54
5	Arn-Gs	35800	391700	51100	391000	10,57
6	Gs-Bzl	51100	391000	55650	388900	10,28
7	Bzl-Krg	55650	388900	61250	387050	10,08
8	Krg-Kbd	61250	387050	66700	383450	10,36
9	Hlds-Hld	68550	443750	67900	444450	0,4
10	Hld-Msw	75750	438150	68550	443750	2,61
11	Kbd-Rb	66700	383450	69650	382200	10,12
12	Msw-Mss	76950	437100	75750	438150	5,51
13	Mss-Vdw	81200	435600	76950	437100	8,52
14	Rb-Bgn	69700	382200	79200	390000	10,08
15	Gvmw-Gv	82000	454050	81050	452450	99,11
16	Vdw-Vdg	83300	435500	81200	435600	9,67
17	Gvmw-Rsw	81050	452400	81700	450750	99,59
18	Gv-Gvc	82250	455100	82000	454050	28,71
19	Gv-Laa	83400	455000	82000	454050	73,43
20	Gvc-Vb	84550	453600	82250	455100	43,31
21	Vdg-Vdo	84400	436200	83300	435500	16,68
22	Gvc-Laa	82250	455100	83400	455000	59,23

Figure 4.2: Example of the railroad data table

Table 4.1: Overview of the attributes of the objects from the road and railroad data tables

Attributes of the road data table	Attributes of the railroad data table
Starting point number (KNOOP_1), and endpoint number (KNOOP_2);	Departure and arrival station (BAANVAKID);
RD-coordinates[a] of the starting point (X1 and Y1) of the schematic link;	RD-coordinates of the starting point, most times a station (X1 and Y1);
RD-coordinates of the endpoint (X2 and Y2) of the schematic link;	RD-coordinates of the endpoint, most times a station (X2 and Y2);
The number of vehicles between the starting and endpoint during the rush hours (COUNT).	The relative number of train travelers between the stations (DRUKTE), a value between 0 and 200.

[a] Rijksdriehoeksnet-coordinates; coordinates in the Dutch reference system.

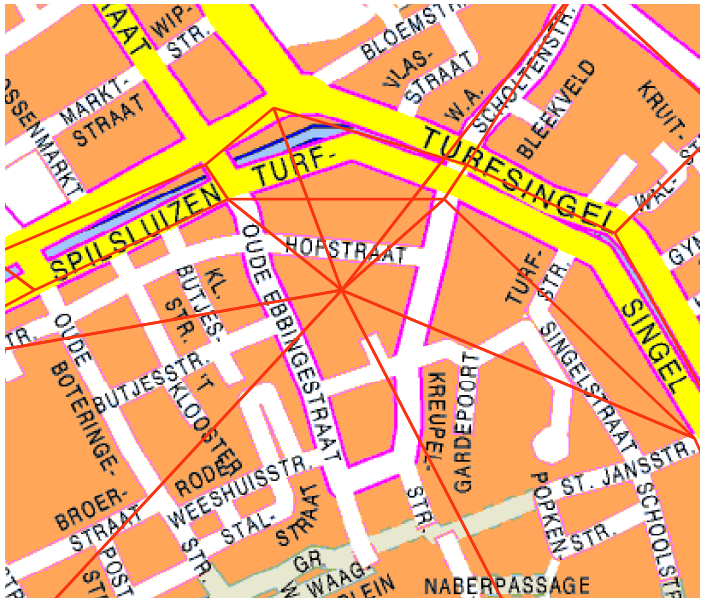


Figure 4.3: Example of a *zonal center* of the schematic links data set

are derived, 36374 schematic links for the road travelers and 429 schematic links for the train travelers, for the whole of the Netherlands. Examples of the elements of the road and railroad data tables are shown in table 4.1. The connection between the coordinate pairs of the starting and endpoints can be rendered by straight lines, which actually represent the schematic relation between the points.

In the available road data set with schematic links, some starting and endpoints seem to be poorly positioned, see figure 4.3. These vertices are *zonal centers*, which indicate where the vehicles from the zones arrive at the network. Some of these zonal center are situated in residential areas and others could be in grasslands. Although the schematic links connecting to the zonal centers don't indicate one particular route, they *do* represent a number of potential users of the mobile telephone network. Therefore, the best match has also to be determined for these schematic links. This will give some extra difficulties when matching the schematic links to the road and railroad data set.

The data set containing the schematic links of the railroad travelers doesn't contain zonal centers. However, for these schematic links the starting and endpoints are mostly not located at junctions in the railroad network, but at stations. This will affect the approach of the matching process only for a small part, as the schematic links only need to be moved to the nearest railroad section and not to a junction in the railroad network.

4.1.2 Detailed network data sets

In this section the data sets are described which contain the detailed road and railroad networks. In order to understand the difference between the different detailed data set, first the definition of raster and vector data will be given. After this definition, the available detailed data sets will be described.

Raster and vector data sets

Raster data are structured as an array or grid of cells, often referred to as *pixels* (see figure 4.4). Each cell in a raster is addressed by its position in the array (row and column number). Rasters are able to represent a large range of computable spatial objects. Thus, a point may be represented by a single cell, an arc by a sequence of neighboring cells and a connected area by a collection of contiguous cells (Worboys, 1995). In raster-based data the relationship between the pixels is not known. The only way any relation is shown, is

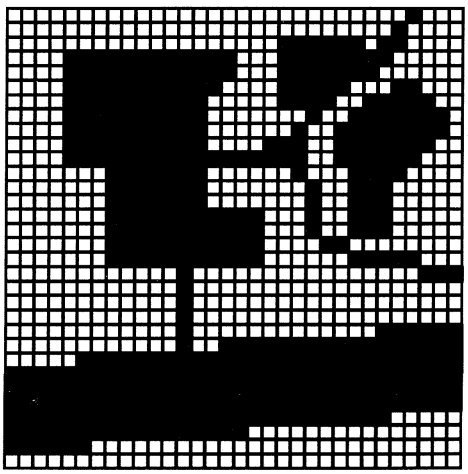


Figure 4.4: Raster data (from Worboys, 1995)

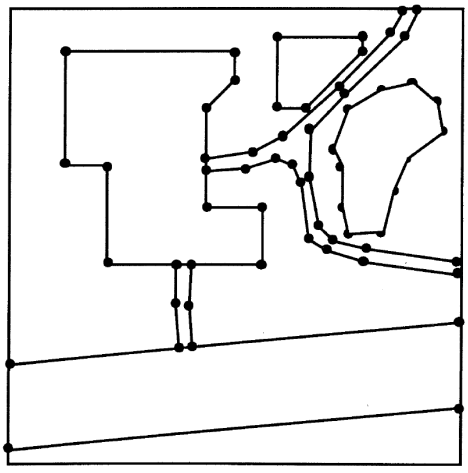


Figure 4.5: Vector data (from Worboys, 1995)

by the attribute values of the pixels and its location. Pixels with the same attribute value usually have some relation, but not necessarily.

A vector is a straight line defined by its end-points, see figure 4.5. The locations of end-points are given with respect to some reference of the plane or higher-dimensional space (Worboys, 1995). Due to these characteristics of vector data, the approach of the data will be totally different compared to the methods which are described in the previous section.

Available road and railroad data sets

To determine the location of the roads and railroads, different detailed data sets are available. These data sets are the following:

- Vector data set with all roads for cars in the Netherlands with an average spatial accuracy of 15 m.
- Vector data set with all railroads in the Netherlands with an average spatial accuracy of 40 m.
- Raster data set with road and railroad data. The raster data sets are available with different generalization levels ($1:5.000$, $1:15.000$, $1:50.000$, $1:150.000$, $1:500.000$, $1:1.500.000$).

Both vector data sets consist only of polylines, no data is available in these data sets of one-way traffic and fly-over junctions (e.g. figure 4.6). The routes between the starting and endpoints of the schematic links can consist of junctions that are not possible in the real world. It could be possible to use other data sets which contain this kind of information. In that case more conditions can be stated to the routes and thus less routes will be found to be compared with each other in order to choose the best one.

The vector data sets contain different sorts of additional attribute information for each polyline. The attributes that will be used in the programme are:

- $\textsc{RoadClass}$ – indicates the kind of road that is represented by the polyline. The value of the $\textsc{RoadClass}$ is an integer value between 0 and 5. In this data set, a $\textsc{RoadClass} = 0$ indicates a highway and a $\textsc{RoadClass} = 5$ indicating a minor road.
- $\textsc{name1}$ – gives the street name of the street that is represented by the polyline.

The raster data could be obtained by scanning paper maps or by remote sensing, but is usually derived from vector data and generalized to obtain maps with different levels of detail. Therefore, the use of raster data could be seen as redundant and could give results with unnecessary errors, because the original vector data should have been used. However, companies (e.g. TNO ICT) don't always have the disposal of the more expensive vector data, so these companies could prefer using raster data and accept the errors. The price of the vector data set is €13.910, compared to the price of the raster data set of €9.660 (the prices are from Falkplan-Andes). When a new network data set has to be purchased, it is advisable to by the vector data set as more geoprocessing tasks can be performed on vector data.

4.1.3 Characteristics of input data that could improve the results

In the description of the road and railroad network data sets the lack of information about one-way traffic and fly-over junctions has been mentioned. When this kind of information would be available the results of the matching process could be improved further. This kind of information can be stored in ArcGIS Network data sets in the following ways:

Figure 4.6: A real fly-over junction

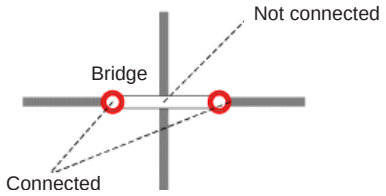


Figure 4.7: A representation of a fly-over junction (from ESRI, 2005b)

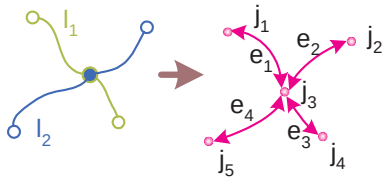


Figure 4.8: Stored without considering fly-over junctions (from ESRI, 2005b)

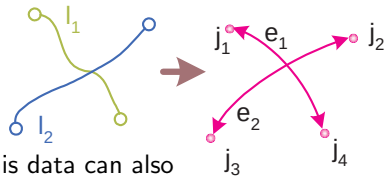


Figure 4.9: Stored with considering fly-over junctions (from ESRI, 2005b)

- *Road and railroad data set* – The current road and railroad data set doesn't have information about one-way streets and fly-over crossings. Both kinds of information can be stored in a *Network data set* in a Feature data set in the *Personal geodatabase*, more information about these data structures can be found in section 4.2. The information of one-way streets and fly-over crossings can be assigned to the data in the following way:

  - When a network data set is created, Network Analyst searches through all sources for commonly used fields, such as one-way. If it finds a one-way field in any source, it creates a one-way network attribute, and assigns values for the relevant source based, see figure 4.10 (ESRI, 2005b).
  - Building networks with endpoint connectivity is one way to model crossing objects, such as bridges, figure 4.7 gives an example. The *Streets source* assigns any vertex connectivity to allow street features to connect to other street features at coincident vertices (shown in figure 4.8). The *Bridges source*, shown in figure 4.9 assigns endpoint connectivity. This means bridges connect to other edge features only at their endpoints. Consequently, any street going under the bridges will not be connected to the bridge. The bridge will connect to other streets at its endpoints (ESRI, 2005b).
  - Falkplan-Andes provides additional navigational information for the vector data sets. This addition costs €9.900, and (as the name suggests) contains all information needed for navigation purposes, such as fly-overs and one-way streets. This data can also be imported in ArcGIS.

Source	Direction	Element
Metro_Lines	From-To	Edge
Metro_Lines	To-From	Edge
Streets	From-To	Edge
Streets	To-From	Edge
Transfer_Stations	From-To	Edge
Transfer_Stations	To-From	Edge
Transfer_Street_Station	From-To	Edge
Transfer_Street_Station	To-From	Edge

Figure 4.10: Storage of one-way streets (from ESRI, 2005b)

- *Better location of the schematic links* – The data containing the schematic links has some unusual locations of vertices, the *zonal centers*. Even human operators cannot relocate to the correct positions, not to mention how the computer should know where to relocate these vertices.

  Better information of the schematic links could result in better matching of the data sets. When looking at the whole network or smaller subnetworks, it should then be possible to match the (sub)network(s) of the schematic links in a relation of $1-1$ to the road and railroad network. With the current data this isn't possible, because too many vertices of the schematic links are situated like in figure 4.3.

4.2 ArcGIS

This thesis is executed in the programme ArcGIS 9.1. The structure of the geographical information in a *personal geodatabase* is described in section 4.2.1, the components of ArcGIS will be described in section 4.2.2, and finally some of the functionalities will be described in section 4.2.3, which could be useful for matching the data sets of this graduation thesis.

4.2.1 Design of a personal geodatabase

To manage your own spatial database, a *personal geodatabase* can be created. The spatial data type, and indexing will then be stored in a Microsoft Access database. Personal geodatabases can be accessed directly in ArcCatalog and are relational databases that contain geographic information.

An overview of the data sets which can be imported in a personal geodatabase is shown in figure 4.11. Geodatabases can contain *feature classes*, *tables*, and *feature data sets*. Feature classes can be organized into a *feature data set*, they can also exist independently in the geodatabase, from Harlow et al., 2004b:

1. *Feature classes* – are a collection of the same type of features, for example, a collection of points or a collection of polygons.

2. *Tables* – may contain additional attributes for a feature class or geographic information such as addresses or x, y, z-coordinates.

3. *Feature data sets* – contain a collection of feature classes that share the same spatial reference. The feature classes in the feature data set can be imported in a *topology*, a *geometric network*, or a *network data set*:

   (a) *Feature classes* – All feature classes in a feature data set share the same coordinate system.

   (b) *Topology* – are useful for maintaining high-quality spatial data by ensuring that the features conform to simple rules. When a geodatabase is set up with a topology, feature classes can be specified to participate in the topology and rules can be defined that control when and how features can share geometry (Pfaff et al., 2004). The relation between the features are not stored, but can be derived from the data when needed.

   (c) *Geometric network* – combines line and point feature classes to model linear networks, and maintains topological relationships between its feature classes.

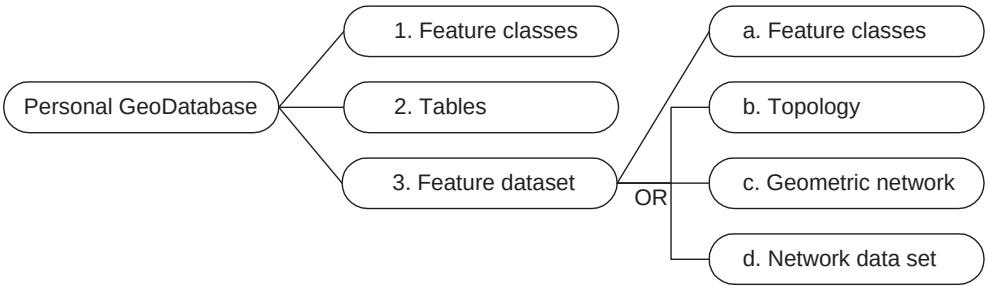


Figure 4.11: Overview of how the data can be stored in a Personal Geodatabase

Geometric networks consist of two fundamental components (Harlow et al., 2004a):

- *Edges* – are a type of network element that have a length and through which some commodity flows.
- *Junctions* – occur at the intersection of two or more edges and allows the transfer of flow between edges.

The edges and junctions in a network are topologically connected to each other. A geometric network contains the connectivity information between edges and junctions and defines rules of behavior such as which classes of edges can connect to a particular class of junction or to which class of junction two classes of edges must connect (Harlow et al., 2004a).

Edges and junctions can have any number of weights associated with them. A weight is a property of a network feature typically used to represent a cost for traversing across an edge or through a junction, in this case, the RoadClass can be used as a weight of the edges. You can use a weight filter to limit the set of network features that may be traced. A weight filter specifies which network features can be traced based on their weight values (from Harlow et al., 2004a).

A utility network is an example of a geometric network, and it is a directed network. This means the agent (for example, water, sewage, or electricity) flows along the network based upon certain rules built into the network. The path that the water will take is predetermined. It can be changed, but not by the agent. The engineer controlling the network can change the rules of the network by opening some valves and closing others to change the direction of the network (from ESRI, 2005b).

(d) *Network data set* (only available with a license for the ArcGIS Network Analyst extension) – Networks used by ArcGIS Network Analyst are stored as network data sets. The network incorporates an advanced connectivity model that can represent complex scenarios, such as multi-modal transportation networks. It also possesses a rich network attribute model that helps model impedances, restrictions, and hierarchy for the network. The network data set is built from simple features (lines and points) and turns (from ESRI, 2005b).

Once a network data set is created or an existing network data set has been edited, it must be built. Building is the process of creating network elements, establishing connectivity, and assigning values to the defined attributes.

When a network data set is built, two new objects are added to the feature data set, the network data set, and the point feature class containing all the system junctions created during the build process.

A transportation network is an example of a network data set and is an undirected networks This means that although an edge on a network may have a direction assigned to it, the agent (the person or resource being transported) is free to decide the direction, speed, and destination of traversal. For example, a person in a car traveling on a street can choose which street to turn onto, when to stop, and which direction to drive. Restrictions imposed on a network, such as one-way streets or "no U-turn allowed", are guidelines for the agent to follow. This is in stark contrast to the utility network (from ESRI, 2005b).

The road and railroad data set of this thesis has been imported in a *topology*, in a feature data set. After the implementation it came clear that using a *geometric network* would

Table 4.2: The extensions of ArcGIS

ArcGIS 3D Analyst	ArcGIS Publisher	ArcGIS Tracking Analyst
ArcGIS Business Analyst	ArcGIS Schematics	ArcScan for ArcGIS
ArcGIS Data Interoperability	ArcGIS Spatial Analyst	ArcWeb Services
ArcGIS Geostatistical Analyst	ArcGIS Survey Analyst	Maplex for ArcGIS
ArcGIS Network Analyst		

have been better, because the geometric networks built their topological relationships in advance, while a topology layer has to built the relationships on the fly (see further chapter 7). Therefore, using geometric networks would decrease the computer time considerably. In spite of this advantage, still the *topology* has been used, because this information became available after the completion of the programme.

4.2.2 Components

ArcGIS Desktop is a collection of software products that runs on standard desktop computers. It is used to create, import, edit, query, map, analyze, and publish geographic information. There are four products in the ArcGIS Desktop collection; each adds a higher level of functionality:

1. ArcReader is a free viewer for maps authored using the other ArcGIS Desktop products. It can view and print all maps and data types. It also has some simple tools to explore and query maps.
2. ArcView (€2.471) provides extensive mapping, data use, and analysis along with simple editing and geoprocessing capabilities.
3. ArcEditor (€11.530) includes advanced editing for shapefiles and geodatabases in addition to the full functionality of ArcView.
4. ArcInfo (€21.061) is the full function, flagship GIS desktop. It extends the functionality of both ArcView and ArcEditor with advanced geoprocessing, such as Geodatabase topology rules, and management of utility networks. It also includes the legacy applications for ArcInfo Workstation.

All ArcGIS Desktop products share a common architecture, so users working with any of these GIS desktops can share their work with others. Maps, data, symbology, map layers, geoprocessing models, custom tools and interfaces, reports, metadata, and so on, can be accessed interchangeably.

During the execution of this project ArcInfo has been used, although TNO ICT has only the disposal of ArcView. This way functionalities of the whole ArcGIS Desktop collection could be examined, but for the implementation of the algorithm these functionalities it was preferred not to use the functionalities which cannot be used in ArcView.

New capabilities can be added to desktop products through a series of ArcGIS extensions from ESRI. Extensions allow you to perform tasks such as raster geoprocessing, three-dimensional visualization, and geostatistical analysis. The available extensions are shown in table 4.2, each costing €4.118.

Especially ArcGIS Network Analyst, ArcScan for ArcGIS and ArcGIS Spatial Analyst have promising functions. Together with other standard functionalities in ArcGIS, the functionalities of these extensions will be examined in section 4.2.3.

4.2.3 Useful functionalities

The use of different extensions has been investigated. TNO ICT has not the disposal of extensions within ArcGIS, so eventually the use of extensions will be limited to a minimum, or the benefits for using extensions should be enough to support buying the licenses. The description below contains the description of the functionalities in ArcGIS Desktop. Corresponding functionalities have been used during the implementation of the programme and are further explained in chapter 7.

Standard functionalities in ArcGIS

Tracing operations (from Harlow et al., 2004a): Using the *Utility Network Analyst* toolbar with your network, you can, among other things, do the following:

- *Find connected features* – to find all of the features that are connected to a given point through your network, use the *Find Connected task*.
- *IForwardstar* – Provides access to members that query information about adjacent elements in the geometric network.

  The *IForwardStar* interface allows to access the fundamental connectivity of a network. (*Forward Star* is a term from computer science for a data structure that stores the connectivity of a graph.) The job of *IForwardStar* is to return information about adjacent elements. An adjacent element is simply a network element connected to another network element. For example, an edge connected to a junction, or a junction connected to a junction via an edge. Quickly finding adjacent elements is the cornerstone of all network solvers.
- *Find a path* – to find a path between two points in the network, use the *Find Path task*. The path found can be just one or a number of paths between these two points, depending on whether or not your network contains loops. It is tried to use this functionality to find different routes (paths) between the vertices of the schematic links. In section 7.7 more information is given about the attempt to adjust the *Find Path task* into a function that finds different routes between the vertices.

ArcGIS Network Analyst

With a license for the ArcGIS Network Analyst extension, you can perform advanced network analyses on network data sets. You will have the full functionality of ArcGIS Network Analyst available to you, whether you're using ArcView, ArcEditor, or ArcInfo (ESRI, 2005b).

Networks used by ArcGIS Network Analyst are stored as network data sets. A network data set is created from the feature source or sources that participate in the network. It incorporates an advanced connectivity model that can represent complex situations, such as multi-modal transportation networks. It also possesses a rich network attribute model that helps model impedances, restrictions, and hierarchy for the network. The network data set is built from simple features (lines and points) and turns (from ESRI, 2005b).

Network attributes are properties of the network elements that control traversability over the network. Examples of attributes include the time to travel a given length of road, which streets are restricted for which vehicles, the speeds along a given road, and which streets are one-way (ESRI, 2005b).

Network analysis often involves the minimization of a cost (also known as impedance) during the calculation of a path (also known as finding the best route). Common examples

include finding the fastest route (minimizing travel time) or the shortest route (minimizing distance). Travel time (drive time, pedestrian time) and distance (meters) are also cost attributes of the network data set.

Restrictions – you can choose which restriction attributes should be respected for calculating the route. Restrictions, such as one-way, should be used while finding the quickest route on streets. If your network data set contains additional restriction attributes, such as weight limit or height limit, those could be used as well. In all cases, a restriction attribute is defined using a Boolean data type.

Allow U-turns – while calculating a route, Network Analyst can allow U-turns everywhere, nowhere, or only at dead ends (also known as cul-de-sacs). Allowing U-turns implies the route can double back on the same street.

If you do not have ARC/INFO or ArcView GIS turn tables but want to use turn information for network analysis, create a new turn feature class and add new turn features to store that information (from ESRI, 2005b).

The functionalities of Network Analyst could be useful when using navigational data sets, which contain one-way streets and fly-overs. The data sets used in this thesis didn't contain these attributes and therefore the use of Network Analyst would be redundant.

ArcScan for ArcGis

ArcScan is a extension for ArcView and is included in ArcEditor and ArcInfo. This extension provides tools and commands that support the conversion of raster data to vector features. This process, referred to as *vectorization*, can be performed interactively or in an automated fashion.

The interactive vectorization experience, referred to as *raster tracing*, requires that you trace the raster cells in the map to create vector features. The automated vectorization experience, referred to as *batch vectorization*, requires that you generate features for the entire raster based on settings that you specify (from ESRI, 2005d).

TNO ICT could have the choice of buying either the ArcScan licence (€4.118) or buying the original vector data set (€12.000), instead of the raster data (€9.600). Buying the raster data set and the ArcScan license together is more expensive then buying the original vector data set. Besides the costs, also vectorizing the raster data set will result in errors which are not in the original data set, preference will be given to use the original vector-based data set. This decision excludes the use of the ArcScan extension.

ArcGIS Spatial Analyst

ArcGIS Spatial Analyst provides a rich set of tools to perform cell-based (raster) analysis. The raster data structure provides the most comprehensive modeling environment for spatial analysis.

Spatial Analyst adds a comprehensive and wide range of cell-based GIS functions to ArcGIS. As a GIS modeler, this is the central toolset you'll use for analysis and modeling. Of the three main types of GIS data (raster, vector, and TIN), the raster data structure provides the most comprehensive modeling environment and operators for spatial analysis (from ESRI, 2005c).

The use of vector data sets has the preference during this research. Because of this and the time limit of this thesis, functionalities of the Spatial Analyst extension have not been examined any further.

In this chapter a description has been given of the preconditions and limitations of this research. The following chapter will examine how the research problem has been treated in related researches about matching data sets containing multiple representations of street data.

CHAPTER 5

Research on matching multiple representations of street data

In this chapter an overview will be given of the current state of the research on matching multiple representations of street data. The purpose of matching the data sets is to get information that is only available in one data set into the other data set. In this graduation research the information of the travelers is represented by the schematic links in a data set.

In order to assign the travelers to the road and railroad network from another data set, both data sets have to be matched. The data set containing the amount of travelers can be seen as a generalized data set of the road and railroad data set. Therefore, the approaches which match multiple representations of street data can be used as basics of this research.

The automatic matching of features from different representations of street data can rely on several algorithms using semantic information (e.g. attribute items), geometric information (e.g. shape) and topological information (e.g. association) of the geographic feature (Féchir and Waele, 2006).

How the approaches can be defined will be described in the following sections. First in section 5.1, the parameters will be described which can be used to give values to the matching rate of the candidate edge to the reference edge.

After this section, different approaches to use quality criteria for automatic selection of correct matching pairs will be investigated. Quality criteria are measures for the resemblance of a single aspect of the matched objects, e.g. the length of the matched lines, similarity of names, etc.. The three approaches are the following (from Mantel and Lipeck, 2004):

- An approach is to define a *threshold value* and discard all matchings, with quality measure below the threshold. After discarding, all remaining possible matchings, which are not in conflict with any other any more, can be confirmed (section 5.2).
- One can compute all the measures of each criterium for all possible matchings and then compute the *best combination* of parameters, that is the combination with the highest sum of measures (section 5.3).
- The last approach can be refined to an *iterative method*, by starting the selection with a high threshold value and then decrease it stepwise until a minimum threshold

value is reached (section 5.4).

The quality criteria are evaluated in section 5.5. In general, all the researches on matching multiple representations of street data conclude that full automatic matching is almost impossible, because in some cases even human operators can hardly give the solution for the best match. The result is highly dependent on the homogeneity of the data.

5.1 Possible parameters

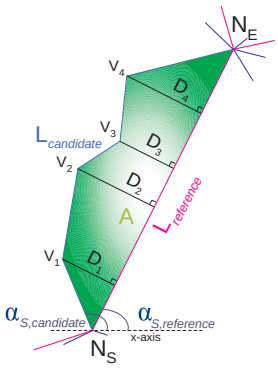


Figure 5.1: Overview of the used parameters

In this section all edge and line similarity-parameters will be defined (from Volz, 2006, Féchir and Waele, 2006, and Zhang et al., 2005) in order to be able to further investigate the approaches. A visual overview of the parameters is given in figure 5.1 for a better understanding of the parameters and the similarity measures that are derived from these parameters, which will be described later in this section. The following parameters are used as basic measures to be able to derive the similarity measures and to give a value to the matching rate:

- The *length* (L) of both the candidate feature and the reference feature. Either the difference or the ratio of these lengths is taken.
- The *angle* (α) between the matching reference and the candidate at the starting and end vertices.
- The *line distance* (D) determined by the shortest distance from the intermediate vertices (≥ 0) of the candidate to the other reference (the schematic link).
- The number of *incident edges* (I) of the start and end nodes of the two edges.
- The *area* (A) of the polygon enclosed by the matching reference and candidate.

In the description of the experiments some similarity measures are derived from the parameters above. With these measures a value can be given to the candidate edge which could be representing the reference edge. In case of this study, the reference edge is called the schematic link and the candidate edge is one of the routes. The following similarity measures are derived (from Zhang et al., 2005 and Volz, 2006), and calculated where possible for the case of figure 5.1:

- *Length difference* (ΔL):

$$\Delta L = L_{candidate} - L_{reference} \tag{5.1}$$

- *Angle difference* (α) between two edges, determined at the starting node as the difference between the larger and the smaller angle against the x-axis:

$$\alpha = |\alpha_{S,candidate} - \alpha_{S,reference}| \tag{5.2}$$

- *Average line distance* $(\bar{D})$, determined as the average distance of the distances of all vertices of two input edges:

$$\begin{aligned} \bar{D} &= \frac{\Sigma D}{\#D} \\ &= \frac{D_1 + D_2 + D_3 + D_4}{4} \end{aligned} \tag{5.3}$$

- *Adjacency relations* of start and end nodes (ΔI), determined as the difference of the number of incident edges of start and end nodes of the candidate data set and the reference data set. The higher detailed data set (the candidate data set) will have more or equal adjacency relations:

$$\begin{aligned} \Delta I &= (I_{N_{S,\text{cand}}} - I_{N_{S,\text{ref}}}) + (I_{N_{E,\text{cand}}} - I_{N_{E,\text{ref}}}) \\ &= (3-2) + (5-3) = 3 \end{aligned} \tag{5.4}$$

- *Angle difference* ($\Delta\alpha$), determined by the angle between the straight line connecting the two end points and the x-axis:

$$\Delta\alpha = |\alpha_{S,candidate} - \alpha_{S,reference}| - |\alpha_{E,candidate} - \alpha_{E,reference}| \tag{5.5}$$

- *Maximal chord difference* ($\max D$), determined as the maximum distance of the vertices to the other edge:

$$\begin{aligned} \max D &= \max\{D_1, D_2, D_3, D_4\} \\ &= D_2 \end{aligned} \tag{5.6}$$

- *Area* (A) of the polygon enclosed by the matching reference and candidate.

With only these similarity measures a selection still can't be made, as only one measure will not give an good indication of the correctness of the match. These parameters will be used in the remaining sections of this chapter and in chapter 6 for defining variants for selecting the best matching route, which represents the schematic link.

5.2 Threshold value

The first approach for determining a quality measure of a match is by defining threshold values for multiple similarity measures. With these threshold values two approaches can be derived (Féchir and Waele, 2006):

- An association is created between the selected features and the source feature. This is done by measures based on several tools and their right parameters.
- Measures or tools will detect irrelevant features and delete the association with the source feature.

The set of possible matching candidates contains not only correct matching, but some incorrect suggestions. In order to exclude the false candidates threshold values are determined by experienced operators and testing values(Zhang et al., 2005):

- $\Delta\alpha < T_\alpha(e.g.15°)$
- $\Delta L < T_L(e.g.20m)$
- $\max D < T_D(e.g.12m)$
- $AL = \frac{A}{\Delta L} < T_{AL}(e.g.10m)$

If necessary, other criteria can be also defined for specific data types. The target polylines fitting for all of these criteria will be confirmed as the promising matching candidates, while others will be rejected.

5.3 Combination of parameters

The combination of different quality measures through a weighted sum, can give an indication of the quality. With the calculated evaluation values the best match can be selected as the match highest/lowest evaluation value.

The absolute edge similarity values are mapped to a corresponding evaluation value by explicit mapping rules. The scale of the evaluation values ranges from 0 (no similarity) to 10 (maximum similarity) for each indicator. The different partial evaluation values are finally aggregated into a total edge evaluation value (T_{ev}^1) using the following weighted sum approach (from Volz, 2006):

$$T_{ev}^1 = \frac{3 \cdot \Delta L + 3 \cdot \Delta\alpha + 2 \cdot \bar{D} + 4 \cdot D_{VH} + 4 \cdot \Delta I}{16} \tag{5.7}$$

The evaluation values are weighted by a factor which was specified on the basis of the operator's expertise regarding the influence of the different geometric and topological similarity values on the total similarity. The aggregation of similarity measures is a difficult problem that can be further optimized within this approach.

Another way of weighting different criteria according to their relative contributions is defined by Zhang et al., 2005. The candidate with the largest weighted sum is regarded as the best candidate, whilst all the others are discarded. The weighted sum of measurements is represented by the variable T_{ev}^2:

$$T_{ev}^2 = \frac{W_\alpha \cdot (1 - \frac{\Delta\alpha}{T_\alpha}) + W_L \cdot (1 - \frac{\Delta L}{T_L}) + W_D \cdot (1 - \frac{\max D}{T_D}) + W_{AL} \cdot (1 - \frac{AL}{T_{AL}})}{W_\alpha + W_L + W_D + W_{AL}} \tag{5.8}$$

Equation (5.8) contains some weighting factors ($W_\alpha, W_L, W_D, and W_{AL}$) which still have to be determined in order to calculate T_{ev}^2. Zhang et al., 2005 gives no solution for these factors.

5.4 Iterative

One of the approaches using an iterative process is by using the threshold values from section 5.2 (T_α, T_L, T_D and T_{AL}). When these thresholds are to large, more than one candidate feature will be selected. Stepwise decreasing the threshold values will finally result in one remaining candidate (from Zhang et al., 2005).

Another approach could be defining the four thresholds to exclude the incorrect matching candidates. Finding the proper value of these thresholds is important because too small values may result in low matching rate and too large ones perhaps lead to many matching errors. To solve this problem, an iterative learning component has been introduced. The initial thresholds are empirically set on a quite small value. After the first initial step, the thresholds can be increased, every time checking the matching rate. When the matching rate is high enough, the best match can be selected by, for example, determining one of the evaluation values of equations (5.7) and (5.8) (Zhang et al., 2005).

5.5 Evaluation of the results

All of the approaches described above are said to be not fully automatic. Especially if the spatial data to be matched are not homogeneous, complicated cases can occur that

can sometimes even only hardly be solved by human operators, i.e. a complete automatic matching is generally unrealistic (Volz, 2006).

Also the other articles notice that after the automatic matching a manual postprocessing should be done to interactive check and correct the results. In order to analyze the result, one must display the two different databases as well as the links between them. It is sometimes hard to understand the geometric relative configuration of objects among all this displayed information (Mustière, 2006).

At this moment, nothing can assure the reliability of the associations between the features of the data sets. It is only possible to check the reliability manually. However, a global indicator, based on the satisfaction degree of some of the parameters calculated during each linking process, could help to retrieve the suspicious features. This indicator is stored in the relation table to guide the manual checks (from Féchir and Waele, 2006).

To improve the accuracy, further information is needed. One of the topological attributes defined in our matching algorithm is represented by the number of incident edges of a certain point (Zhang et al., 2005).

With the results of these researches in mind, the problem will now be executed with the available data sets. The situation is not exactly the same, but some similarities can be found. In chapter 6 the approach of the problem will be discussed by defining different possible solutions.

CHAPTER 6

The matching process

Although in the beginning of this research the intention was to implement two matching algorithms, one which retrieves the detailed network data from raster data sets, and the other which retrieves the data from vector data sets. During the implementation of the first method, it turned out that the implementation was more time consuming than predicted due to unforeseen difficulties for finding different routes in the road and railroad network representing the schematic links (see further section 7.7), so the implementation of the method based on raster data was eventually canceled. The results of this method would probably be less accurate, because the information from raster data is harder to retrieve fully automatically. The algorithms that were defined in advance, before the implementation in ArcGIS, can be found in appendix B.

In this chapter only the global procedure for matching the schematic links with the detailed (the vector based) road and railroad data set will be described. For more detailed information, on the implementation of the programme, see further chapter 7. The global matching procedure is the following:

1. Match vertices of the schematic link to junctions of the road and railroad network (section 6.1),
2. Find different candidate routes (section 6.2), and
3. Select best match (section 6.3).

6.1 Match vertices to network

For the schematic links of the road data, first preparatory step has to be taken to relocate the starting and endpoint of the schematic links to junctions in the road network. The following steps should be executed in preparation, see also figure 6.1:

1. Determine the distance of all junctions in the road network within some buffer around the starting or endpoint of the schematic link.
2. Select the nearest junction, with or without using the RoadClass as a weighting factor.
   The weighting factor makes it possible to select a junction that is further away, but belongs to a major road, instead of a nearer junction belonging to a minor road. This is desirable because the schematic links represent mainly major roads.
   Additionally, the number of incident edges of the junction can be taken into account.

This number should always be larger or equal to the number of incident schematic links in this junction.

The process of moving the vertices and later finding the best route can be accelerated when the lines of the road and railroad network first are joined as polylines. Only the lines can be joined which meet in a point where no other lines meet, i.e. the point has exactly two incident edges. Further, both lines should have the same street name (NAME1). When the union of the lines has been performed, less data have to be examined in order to find the right junction or route.

6.2 Find different candidate routes

After matching the vertices of the schematic links with the road network, different candidate routes on the road network between the vertices of the schematic link have to be determined. Determining different routes between the vertices of the schematic links can be performed by traveling along the road network from one vertex to the other.

Depth-first search algorithm

Formally, the Depth-first search algorithm is an uninformed search that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hadn't finished exploring. In a non-recursive implementation, all freshly expanded nodes are added to a LIFO (Last In First Out) stack for expansion.

In figure 6.2(a) the algorithm is explained for a simple graph, which represents a road or railroad network. The procedure for building the spanning tree of figure 6.2(b) can be used to determine the different routes from S to E, is as follows:

1. The algorithm starts at the starting point of the schematic link, in figure 6.2(b) indicated by S.

2. The tree is expanded by adding the child nodes of the starting node (nodes 1 and 4). If one of the child nodes is the same as one of the parents, this node will not be added. This condition avoids the creation of loops in the candidate routes.

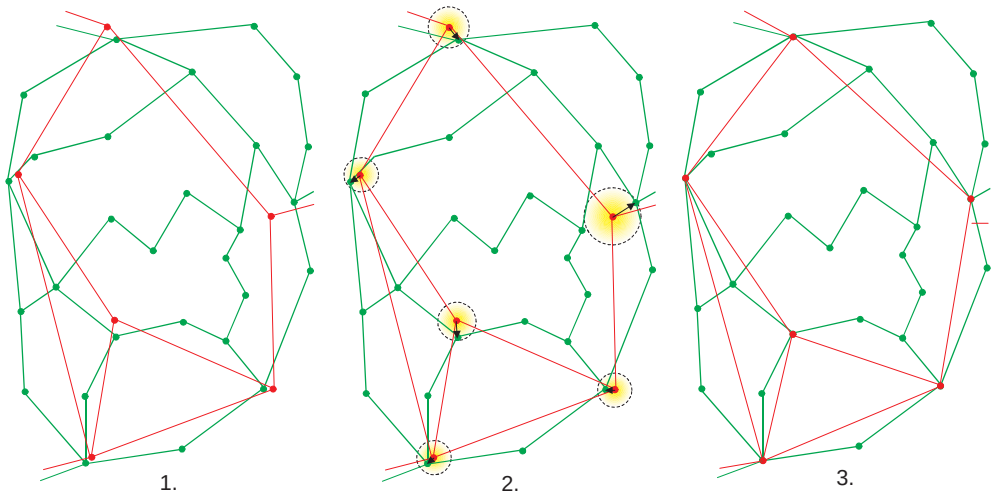
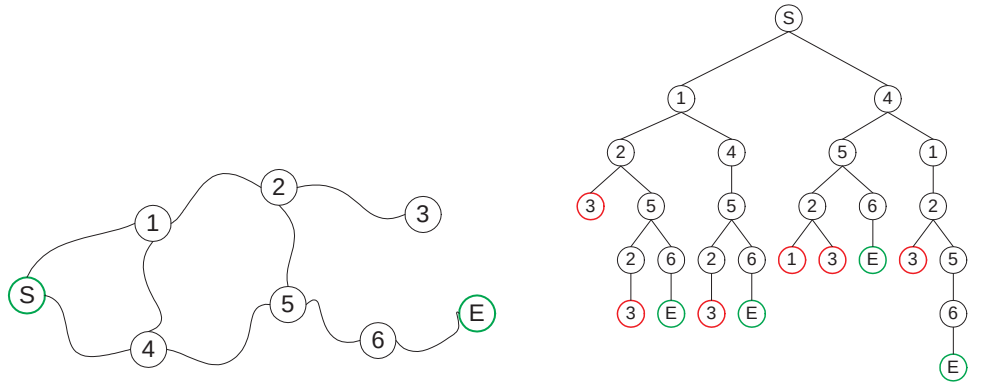


Figure 6.1: Matching the vertices of the schematic links with the junctions in the vector-based data



(a) Example of a peace of road network between vertices of a schematic link

(b) Spanning tree derived with depth-first search of the network

Figure 6.2: Explanation of the Depth-first search algorithm

3. Take now one of the child nodes as the search node (e.g. node 1).

4. Add nodes 2 and 4 as child nodes of node 1. Again, to avoid loops in the candidate routes, don't add nodes that exists as a parent in the tree (thus don't add node *S*).

5. Next again take one of the child nodes as the search node (e.g. node 2) and repeat steps 3 and 4 until there are no child nodes to explore. If the search node is a node without child nodes, set the parent node of the current search node as the search node.

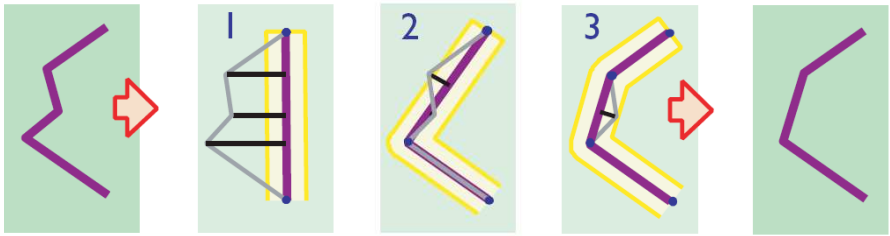
For finding different routes between the vertices of a schematic link, this procedure can be executed. In case of the network in figure 6.2(a) only four routes are stored. The other endings in the spanning tree didn't end in the endpoint, and are thus not stored.

6.3 Select best match

There are various ways for selecting the best route. Below a short description of different selection criteria will be given which have not been described in chapter 5. For more detailed descriptions see appendix B.2.

The criteria all compare the routes and selects the route that has the best value that is calculated based on the criteria. When matching the candidate route to the schematic link, one or more of these criteria could be used:

- The area formed from the schematic link and the route (appendix B.2.1).
- The length of the route (appendix B.2.2).
- The angles at the starting and endpoint, between the schematic link and the route (appendix B.2.3).
- The maximum distance from the intermediate points of the route to the schematic link (appendix B.2.6).
- The shape of the route after generalization (appendix B.2.4).
- The smoothness of the shape of the route (appendix B.2.5).



(a) start (b) 1 (c) 2 (d) 3 (e) end

Figure 6.3: Generalization with the Douglas-Peucker algorithm (from Zeiler, 2001)

Line simplification

Line simplification algorithms are normally used to reduce the number of points required to represent a digitized line. The major objective of simplification algorithms has been the creation of accurate representations of lines using a minimum of processing time and storage. For a series of practical considerations, the superfluous data gathered in the digitization stage should be eliminated. Different algorithms for line simplification have been developed, like *Jenks'*, *Reumann-Witkam*, *Opheim*, *Lang*, and *Douglas-Peucker* line simplification algorithm (for more information about these methods see McMaster, 1987).

Since the programme is implemented in ArcGIS, and ArcGIS contains a predefined function *generalize*, this function has been used for generalizing the routes. The *generalize* function implements the *Douglas-Peucker* line simplification algorithm, so this algorithm will be explained more thorough.

The Douglas-Peucker line simplification algorithm has been developed by David Douglas and Thomas Peucker in 1973. The Douglas-Peucker line simplification algorithm has been widely used over many years for data cleaning and simplification. The algorithm functions in the following manner:

1. The start and endpoints of a line are connected by a straight line segment (figure 6.3(a)).

2. Perpendicular offsets for all intervening points are then calculated from this segment, and the point with the greatest offset is identified (figure 6.3(b)).

3. If the offset of this point is less than some preassigned tolerance, then the straight line segment is considered adequate for representing the line in simplified form. Otherwise, the point is selected, and the line is subdivided at this point of maximum offset (figure 6.3(c)).

4. The process is then recursively repeated for the two parts of the line until the tolerance criterion is satisfied (figure 6.3(d)).

5. Selected points are finally chained to produce a simplified line (figure 6.3(e)) (Vaughan et al., 1991).

Specifying a low tolerance value results in little line detail being removed whereas specifying a high tolerance value results in all but the most general features of the line being removed (Whyatt and Wade, 1988).

Sliver polygons

Most variants for calculating the evaluation value are derived from the theory of calculating sliver polygons. Sliver polygons are small and relatively narrow polygons formed as a result of overlaying two or more layers of vector data. These are formed due to small differences in the way that identical lines have been digitized and it also occurs along map borders when two maps are joined, as a result of inaccuracies of the coordinates in either one or both maps. Of course not all these properties are applicable as properties of the likeliness that a route matches the schematic link. Properties of sliver polygons are (from White, 1997):

- *Area* – Slivers are small.
- *Shape* – Slivers are long and thin.
- *Number of arcs* – Slivers generally have only 2 bounding arcs while real polygons rarely have only 2.
- *Junctions* – Slivers terminate in 4 junctions, but 3 arc junctions are more common in real polygons.
- *Chaining* – Slivers tend to occur in chains.

In this chapter only the theoretical background and global procedure have been described. In chapter 7 the implementation of the procedure will be described, containing more detailed information on the steps which are described above and the pretreatment steps.

CHAPTER 7

Implemented procedure for matching network data with schematic links

After examining of different ways for approaching the problem, the algorithms for matching both data sets have been implemented. The programme itself has been added on a CD in appendix C. The description of the procedure of the programme of this research is described in this chapter. First, the startup of the programme, together with the parameters which are necessary for running the programme, will be described in section 7.1. Section 7.2 gives an overview of the executed steps of the programme. These steps are further examined in the sections that follow this section, sections 7.3 to 7.8.

7.1 Starting the procedure

The user has to specify the characteristics of the input data on a form (shown in figure 7.2) that will appear when the application is started in ArcGIS. This way the programme can also be used with different input data. On the form the user can also specify which operations have to be performed on the data. Some of these operations are time consuming to perform, like the union of the polylines of the street data (although with proper spatial data, which contains the topological structure, this can be done fast). However, some of these operations don't need to be executed for each calculation.

According to the numbering in figure 7.2 the following variables and options can be specified before the programme is run (recommendations for values of the variables will be given in the description of the programme). The right side of the form (numbers 9.-14.) contains the meta information, which is dependent on the attribute names in the input data sets:

1. *Layer schematic links* – The name of the layer which contains the schematic links. This layer should be a *FeatureClass* in a *PersonalGeodatabase*.

2. *Vector data set roads or railroads* – The name of the data set which contains the road or railroad network. This layer has to be a topology layer.

3. *Save output data set in* – The place where the output files will be stored, such as

the file containing the distances of the moved vertices of the schematic links, and the file which contains the input of this form.

4. *Maximum distance to move vertex* (MAXDISTANCE) – The maximum distance (in meters) to move the vertices of the schematic links to a junction on the road and railroad network. When the nearest junction is further away then this value, the vertex is not moved. The schematic links containing this vertex are thus not relocated to the network and therefore no route can be found.

5. *Maximum number of routes per schematic link* (MAXROUTES) – The maximum number of candidate routes that will be created. This index will be used to avoid endless calculations for one schematic link. When this number has been reached, the calculation for this schematic link will be abruptly ended, independently whether the best route has been found or not.

6. *Maximum number of polylines* (MAXPOLYLINES) – This number determines the maximum number of polylines from the road or railroad network of which the route can consists of. When the route that is created is longer than this number, this route is considered as a false potential candidate route and will be discarded.

7. *Maximum detour of the route* (MAXDETOUR) – The value of this parameter indicated the maximum allowable detour of the route. During the process of finding a route, the end of the route should always come nearer to the endpoint of the schematic link, with a tolerance as large as this value. In the example of figure 7.1, the route will not be discarded, because all calculated distances are decreasing as the vertices the route meets are coming closer to the endpoint of the schematic link. If the route doesn't get closer to the endpoint of the schematic link, a tolerance is allowed as large as the value of MAXDETOUR.

8. *Steps to be executed* – These checkboxes are used to indicate which steps of the programme will be executed. The checkbox for *select best route* can only be checked when the checkbox for *find different routes* has been checked too, as the selection takes place right after the creation of a route.

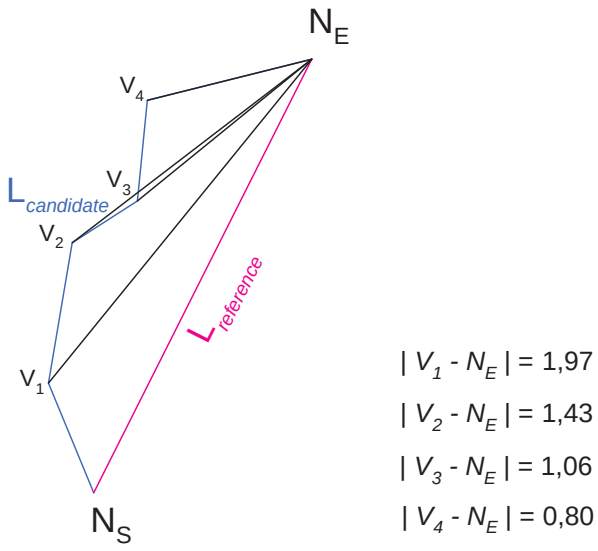


Figure 7.1: Explanaition of the MAXDETOUR-variable

9. *Starting and endpoint number* – The field name of the schematic links layer (which is indicated at 1.) containing the point number.
10. *Starting and endpoint* X- *and* Y-*coordinates* – The field name which contains the attribute values of the X- and Y-coordinates of the starting and endpoints in the layer of the schematic links.
11. *Schematic link count* – The field name containing the number of travelers of the schematic links.
12. *Street names* – The field name of the road or railroad network data set which contains the street names.
13. *Roadclass* (ROADCLASS) – The field name containing the roadclass of the road and railroad data set.
14. *Major and minor roadclass value* – Values that indicate the major (primary) roads and the minor (local) roads. All values of the roadclass are presumed to be within this interval.
15. *Save form* – When clicking this button, the input values of this form will be stored in a text-file named *Specifications.txt*, in the folder of 3.
16. *Save & execute* – The form will be saved and the programme will start executing the steps, which are checked in 8. using all values that are specified on the form.

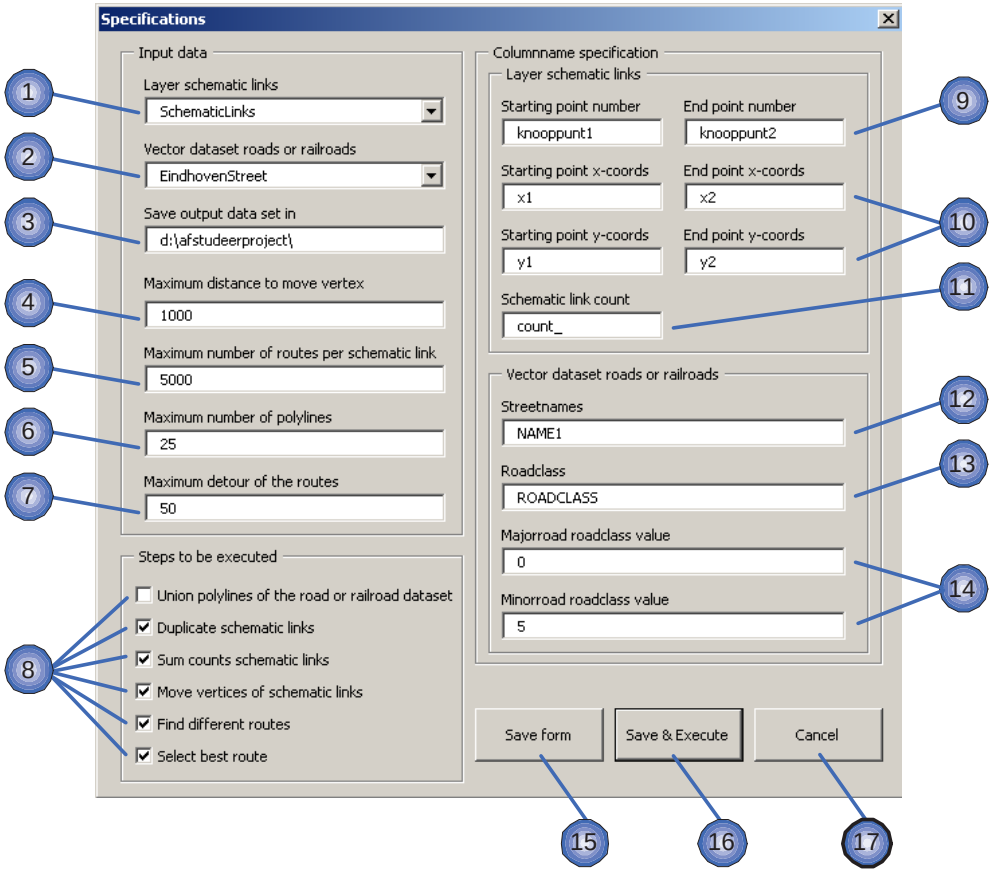


Figure 7.2: The form for specifying the input

17. *Cancel* – Clicking this button will close this form, *without* saving the values on the form.

When all these parameters are filled in, the programme can be run. The next chapters will describe what steps the programme will execute.

7.2 Global overview of the procedure

When the road and railroad data sets are vector based, the process of matching the data will globally follow the steps as shown in figure 7.3:

- The first step (*Union polylines of street layer*) is to thin out the road and railroad data set by joining streets which are in line of each other and have both the same street name.
- The second step *duplicates* the original layer containing the schematic links in order to avoid making changes in the original data.
- Then COUNTs of the schematic links which share the same starting and endpoints, but represent opposite driving directions, are *summed*, to decrease the number of schematic links.
- The vertices can now be to junctions of the road and railroad network, in the step called *Move vertices of schematic links*. If no junction can be found, the schematic link will not be matched to any route of the road or railroad network. This schematic link will remain as it is.
- After these pretreatment steps, the different candidate *routes* can be determined that could be representing the schematic links.
- For each route a matching value is calculated, that indicates the matching rate. This value is compared with de values of each other candidate route and the candidate route with the best matching rate is chosen as the *best match* for representing the schematic link.

Especially the last step, selecting the best route, can be performed in different ways. These alternatives have been examined and different approaches have been found, most of these approaches are based on how humans match the data by looking at it. With these alternatives the reliability and accuracy can be determined. As shown on the form in figure 7.2 it is the choice of the user to use all or some of the steps that are shown in figure 7.3.

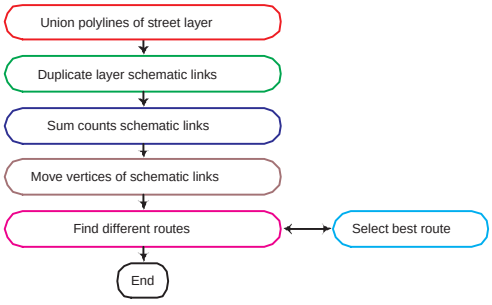


Figure 7.3: Overview of the procedure

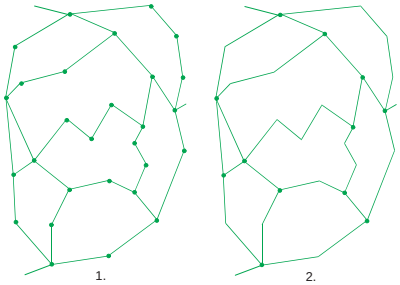


Figure 7.4: Purpose of joining polylines

7.3 Union polylines of road and railroad data

The road and railroad data consists of lines that are placed in line. Most of these intermediate junctions are no real crossings of two or more different roads. In this step these polylines are joined. This step can decrease the amount of polylines stored in the layer with road data, see figure 7.4. There are two conditions that have to be fulfilled, then the polylines will be joined:

1. The polylines have to be connected in the start or endpoints and not in a point in between.
2. Both polylines also have to have the same street names, because then both polylines represent the same street and separately storing the street by two polylines is redundant.
3. No other polylines should begin or end in the point where the polylines are connected according to 1.

After this check the next polylines are selected and compared, as can be seen in figure 7.5. When the amount of road data is limited, this procedure will only take a limited amount of time to be executed, but it will increase fast to long calculation times when the amount of data is larger, because every polyline of the road and railroad network has to be checked whether or not it could be joined with its neighbors. Good spatial data structures could prevent the computer time from increasing fast, because when storing the topology, the number of incident edges of each polyline is known in advance and not all polylines have to be checked anymore, only the polylines with one incident edge in each or one of its end points.

The road and railroad data set will not be updated as frequently as the data containing the schematic links. Therefore, this action can be performed once, and the results can are stored in another layer. This new layer can be used every time the programme is run, and the *union layer* step can be left out. This procedure has the advantage that when the correct junctions are related to the vertices of the schematic links, the road in between these junctions can be easily reconstructed.

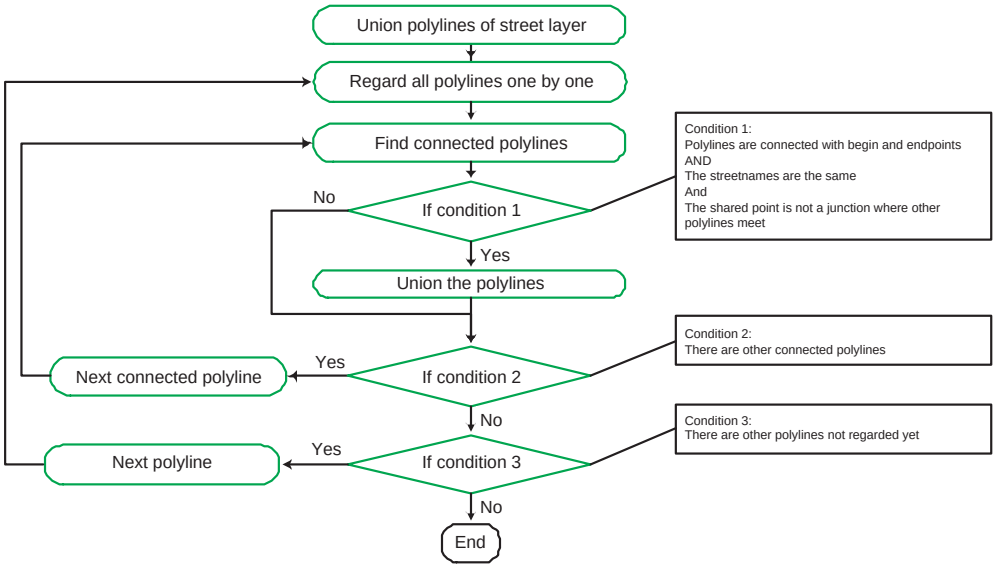


Figure 7.5: Union polylines of road data

7.4 Duplicate layer

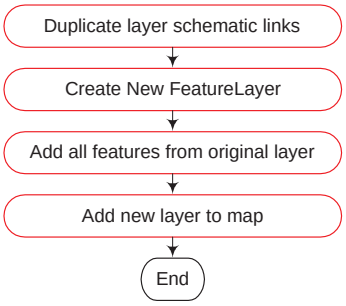


Figure 7.6: Duplicate layer of schematic links

This step is recommended to use, because it avoids changes in the original input data. When different parameters have to be compared, this can be easily done, because each time the original data is copied. The procedure is quite easy and follows the steps as shown in figure 7.6.

7.5 Sum counts

Sum counts of the schematic links is executed for schematic links that are between the same vertices, but are counts of different directions. For the application of this programme only the distribution of the total amount of traffic has to be determined, so the counts of overlapping schematic links can be added to the first and the other can be removed from the layer, this can also be seen in figure 7.7.

In this step the starting and end vertices will be compared. When one schematic link is determined from vertex A to vertex B and another from B to A, the COUNT-values will be summed. By executing *Sum counts* redundant calculations are prevented, because the following steps in the procedure are similar when the schematic links are similar.

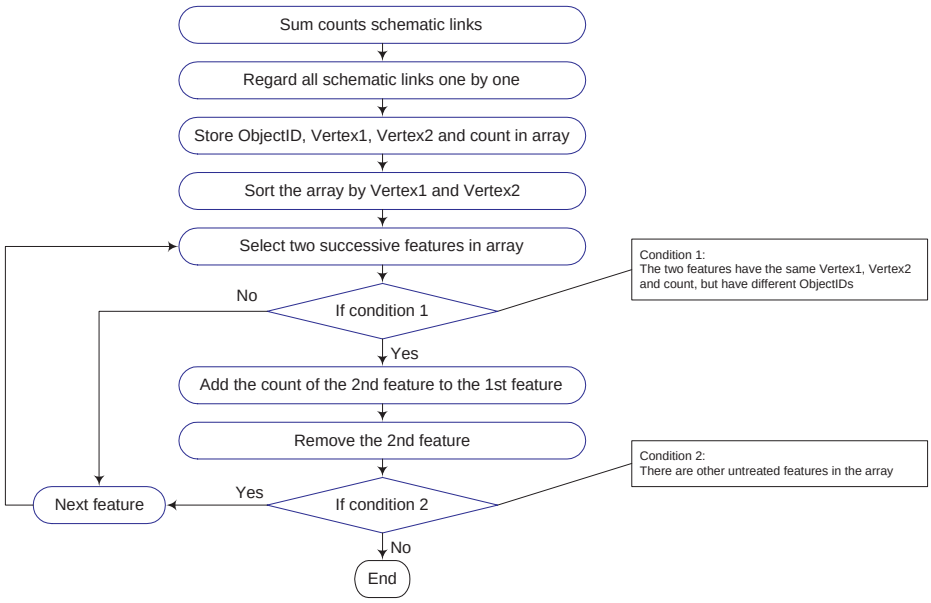


Figure 7.7: Sum counts of schematic links

7.6 Move vertices

The next step in the procedure is *Move vertices*, which links the vertices of the schematic links to the junctions in the road or railroad data, see figure 7.8 where the original schematic links are red and the moved links are green.

First all vertices with their coordinates are stored in an array and then for each vertex the nearest junction is determined (see figure 7.9), and taking also the ROADCLASS of the junction into account by multiplying the distance with the ROADCLASS:

$$\text{RelativeDistance} = Distance(Vertex \leftrightarrow Junction) \cdot (\text{RoadClass} + 1) \quad (7.1)$$

The road and railroad data set contains RoadClass-values in the range of 0 to 5. When the distance is multiplied with only the RoadClass, multiplication with 0 can occur. In that case, the vertex will always be moved to the junction with RoadClass = 0, what is not desirable. Therefore, 1 has been added to the RoadClass. In figure 7.8 the vertices are not moved to the nearest road junctions, but to junctions which are further away, with lower RoadClasses.

In figure 7.8 can also be seen that for each vertex the nearest junction is determined, not considering the relationships between the vertices. This can result in starting and endpoints of schematic links that are not connected through the road or railroad network. For these schematic links not candidate routes can be found in the next step. In order to maintain the data of possible users of the mobile telephone network, which should be traveling in this area, the schematic links for which no candidate routes can be found will remain at the moved locations.

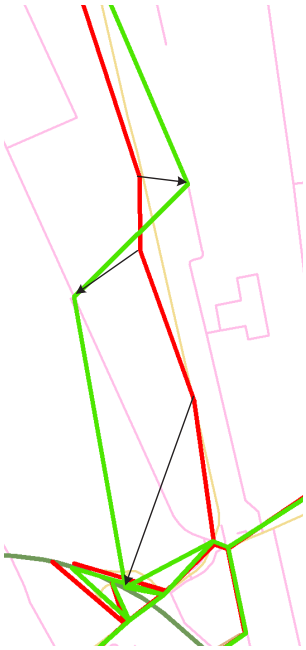


Figure 7.8: Result of move vertices

The storing of the vertices point numbers avoids redundant calculations, because most of all vertices of the schematic links are used more than once. The RoadClass is taken into account because the schematic links mostly connect junctions of the major roads. Thus even though a junction of a minor road is closer to the vertex, still a junction of a major road can be selected as the best matching junction for this vertex of the schematic link. After this step all vertices of the schematic links have been moved to junctions on in the road or railroad network. When no RoadClass is available in the road and railroad data set, a constant value will be used. Consequently, all roads and railroads are considered equally.

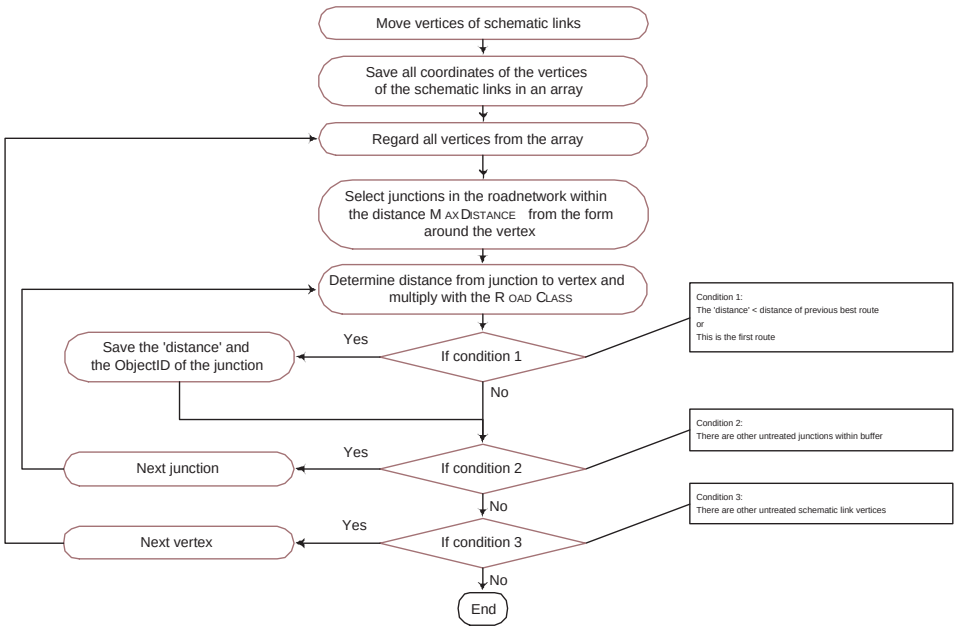


Figure 7.9: Move vertices of schematic links

7.7 Find different routes

In order to find the best matching route between the moved vertices of the schematic links, all possible different candidate routes between the vertices have to be determined. The procedure of this step is described in figure 7.10. The basics for finding the routes is derived from the Depth-first search algorithm described in section 6.2. In this algorithm the spanning tree of a graph is systematically determined, based on the order of the nodes.

The different routes are constructed by beginning in one vertex of the schematic link and *travel* along the road and railroad network according to the procedure described above for building a spanning tree. This network is a layer of polylines and can be described in two ways:

1. collections of geometric features, and
2. a graph of topological elements (nodes, edges, faces, and their relationships).

Therefore finding the child nodes in the network can also be executed in two ways:

1. determining the edges that share the node, or
2. directly from the topological graph, as the relationships of the nodes and edges are already stored.

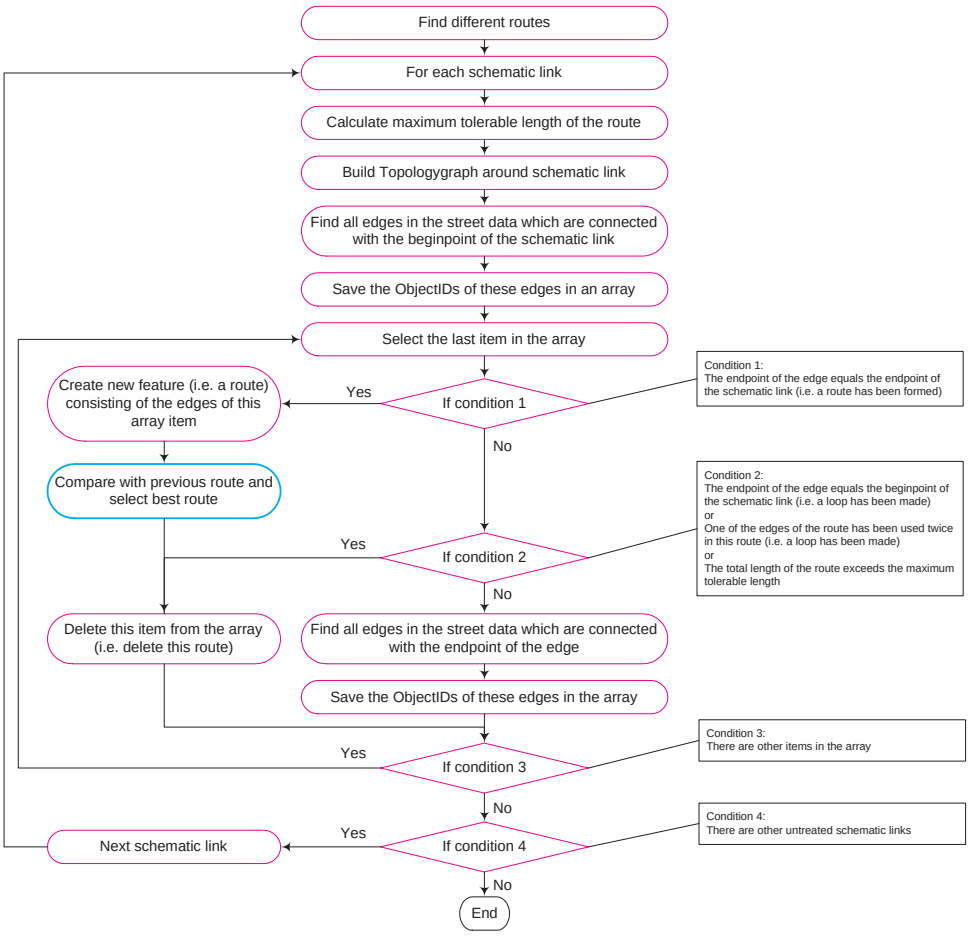


Figure 7.10: Find different routes

In the programme, the properties of the topological graph have been used, because of reduction in calculation time. For that reason figure 7.10 contains an intermediate step called *Build topologygraph around schematic link*. This is necessary to build the topological graph in order to find the incident edges in the nodes, which are the child edges as mentioned in the description of the Depth-first search algorithm above.

The purpose of building the topological graph of only a small part of the network is to allow multiple users use the same data. Earlier the ArcInfo Coverage only allowed single-user editing, because of the need to ensure that the topological graph was synchronized with the feature geometries. Also the coverage had the disadvantage when edits were made to features in a topological data set, a geometric analysis algorithm had to be executed to rebuild the topological relationships.

The new geodatabase topology model of ArcGIS 8.3 and further implements a mechanism that stores features using the simple feature geometry but enables topologies to be used on this simple, open data structure (from ESRI, 2005a).

When searching for routes also the part of the network is examined that is not in between the vertices. To avoid infinite searches for routes, some conditions have been imposed on finding the next edges in the route (see also figure 7.10):

- To avoid creating loops in the candidate routes, it is not allowed for a child node to be the same as one of the (indirect) parents. This is checked by using the function *IPoint.Compare(other IPoint)*, which compares the X- and Y- coordinates and the ID of this point (in that order) with that of the other point. With this function the current child node is compared with each other node in the unfinished route. Whenever one of the other nodes is the same, the route is discarded.

- The distance from the new child node to the end node should be smaller than the distance from its parent to the end node, with a tolerance of the value MAXDETOUR. This is calculated with the function *IProximityOperator.ReturnDistance(other IGeometry)*. This function always returns the minimum distance between two geometries, thus the Euclidian distance between these points is determined, and not the distance over the network. These geometries can be points, lines or polygons, in this case two points.

- The total distance of the route should be maximal twice the Euclidian distance between the starting point and the endpoint, otherwise too many routes will be found, what will result in too long calculation time. The length of the route can be determined by summing all lengths of the polylines from which the route is built up. The lengths of polylines are returned by the function *IPolyline.Length*.

- The number of nodes should not exceed the maximum number of nodes as specified on the form (figure 7.2), MAXPOLYLINES.

- The edges of the route should only have been used once. Currently, the search for routes is performed independently on the routes representing other schematic links. Therefore, the routes representing different schematic links can share the same polylines.

If at least one of these conditions is not satisfied, this unfinished route is considered as invalid and will be discarded. The parent of the last node is selected and the search is continued. When a route has been found that starts in one vertex of the schematic link and ends in the other vertex, it will be determined how good the route is. How this is determined, is described in section 7.8.

Problems in implementing the step for finding different routes

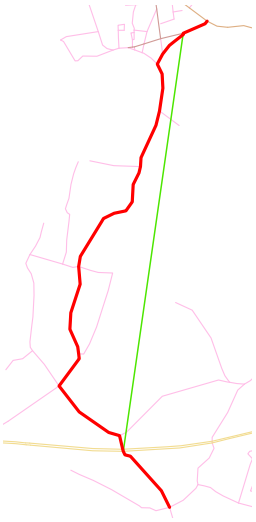


Figure 7.11: Result of *Pathfinder*

The search for different routes on the road or railroad network, which connect the vertices of the schematic links, was more difficult than presumed. ArcGIS provides different ways for finding the shortest path on a network, but no function is available to detect different routes.

The predefined functions only find the shortest routes. Different approaches for finding the shortest paths have been examined and adjusted in order to eventually find the routes. Despite of all these attempts, none of these functions could be adjusted in such way that they could also save the other routes that have been found. Below, the functions are described which have been adjusted in order to find the routes in between the vertices of the schematic links:

- *Pathfinder* – This COM-object demonstrates several network interfaces and presents a simplified interface for finding the *shortest path* through a set of points. You can register and use this DLL as is or customize it further for your application (from ESRI Developer Network, 2006c). The algorithm of the *Pathfinder* has been included on the CD of appendix C. After customizing the *Pathfinder* in order to get it working, the results consisted of different routes through the network that contained the vertices of the schematic links. The positive result was that different routes were found to connect the vertices of the schematic links.

  However, the negative result was that the vertices of the schematic links were not the starting and endpoints of the routes. Instead, at each ending of the route, one polyline was added to the route. This is also shown in figure 7.11. In this figure, the route is indicated by the red line, and the original schematic link by a green line. I was not able to correct the implemented code to remove these extra polylines of the routes, due to the complicated way the pathfinder is implemented. Also for this reason, I was not able to re-implement the code in such way that all candidate routes were stored, and not just the shortest.

- *Create Shortest Path Feature Task* – According to the description of (ESRI Developer Network, 2006a), this task creates a new polyline feature that correspond to the Shortest Path between the two input points. The sample is using the Dijkstra-algorithm with a weight corresponding to the Euclidian distance between nodes of the graph. The sample has been added on the CD of appendix C.

  This description seems promising, but adjusting the function to find more than one route hasn't been accomplished. The implementation of the *Feature Task* was complicated is such way that I was not able to understand the process of the task. As a result, the task couldn't be adjusted and still only the shortest paths could be found.

After these two attempts of using predefined functions from ArcGIS, it has been decided to implement the whole path finding algorithm according to the procedure described before in this section. Some parts of the *Pathfinder* and the *Create Shortest Path Feature Task* were useful to understand the approach of topological networks, in order to find the connected polylines of a junction.

7.8 Select best route

In this final step in the procedure for determining the best match between the schematic link and the road and railroad data, the routes are compared to each other and the *best route* is selected. The procedure for selecting the best route is described in figure 7.12. This selection can be executed in different ways, as will be described in this section and are called the different variants.

In order to be able to select the *best* routes from the candidates which have been found in the previous step, first the properties of the ideal route have to be described. The data that is represented by the schematic links is obtained by counting the number of travelers between junctions of the major roads and the relative amount of passengers in the trains. Especially for the schematic links of the road data, it is necessary to give a more precise description of which roads are represented by the schematic links as the road network data is dense. As a consequence a lot of routes can be found that could match with the schematic link and a choice has to be made. The properties of the *ideal* route are:

1. Only the travelers of major roads are counted, thus the route should have a ROAD-CLASS that is as small as possible, when a major road is represented by a small number and a minor road is represented by a high number.

2. The route should consist of as less as possible junctions, because major roads generally consist of less junctions then minor roads.

3. The route should be as straight as possible, as major roads generally have less curves.

With the parameters which have been described in chapters 5 and 6 the following *evaluation values* (*EV*) can be calculated, which are measures for the matching rates of

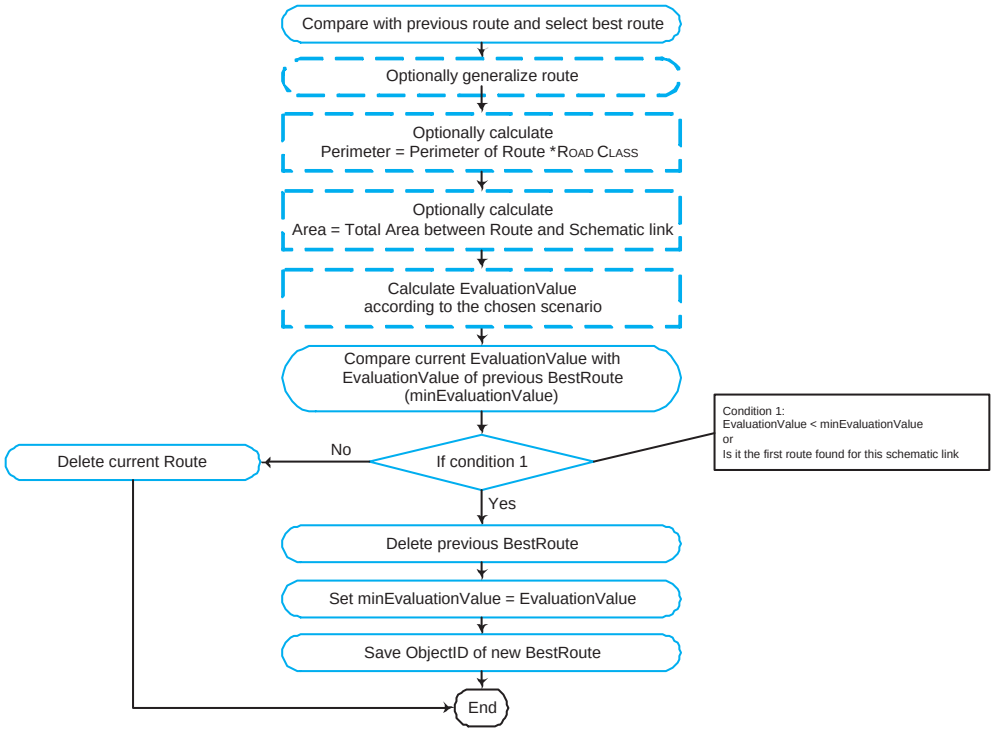


Figure 7.12: Select best route

the candidate routes. The candidate route which is represented by the smallest evaluation value is considered as the *best* match with the schematic link.

- The Douglas-Peucker line simplification algorithm, described in section 6.3, can be used as a measure for the smoothness of the candidate routes in the following two ways:
  - By comparing the number of (intermediate) points of each candidate route after generalization:

$$EV_1 = \#\text{points of generalized route} \tag{7.2}$$

  - The ratio of the number of intermediate points of the generalized route and the number of intermediate points of the original route. This value will be low when the original route has a high degree of generalization, which indicates a smooth original route:

$$EV_2 = \frac{\#\text{intermediate points of generalized route}}{\#\text{intermediate points of original route}} \tag{7.3}$$

- The properties of sliver polygons, which are described in section 6.3, can be used as an indication of the likeliness that a route matches the schematic link. Especially the first two properties, the area and the shape, are the properties that will be used for determining this likeliness. The area will be calculated as the absolute area, i.e. the *absolute* areas created by the schematic link and the candidate route when they intersect (see figure 7.13), will be summed not considering the positive and negative areas. The detection of sliver polygons is performed by calculating the *sliver value*. In the literature different ways can be found for calculating the value:
  - By the ratio of the area of the polygon and its perimeter. With this definition, a lower sliver value will indicate a smaller area and thus the polygon will more likely be a sliver polygon, thus (from ArtWork, 2006):

$$EV_3 = \frac{2 \cdot \text{Area}}{\text{Perimeter} \cdot (\text{RoadClass} + 1)} \tag{7.4}$$

  - By the so-called *Thinness Ratio*. The Thinness Ratio is the relation between the square of its maximum elongation in X- and Y-axis and the area of the

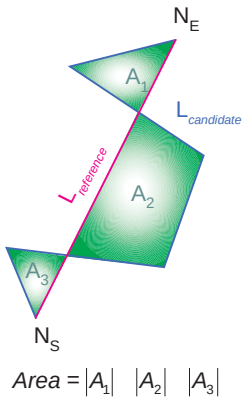


Figure 7.13: Determination of the *area* when the candidate route and the schematic link intersect

polygon (from Arc4You, 2003). The maximum elongation is the same as the length of the schematic link, which is for each route the same. Therefore, this value will always be the same, and is thus not usable. Instead of the maximum elongation, the length of the route can be used:

$$EV_4 = \frac{(\text{Length route} \cdot (\text{RoadClass} + 1))^2}{\text{Area}} \tag{7.5}$$

- In graph theory, the single-source shortest path problem is the problem of finding a path between two vertices such that the sum of the weights of its constituent edges is minimized. In this thesis project, the different routes have already been found. So instead of determining the path length during the search for a path, the path lengths will be determined afterwards:

  $$EV_5 = \text{Length route} \tag{7.6}$$

  A variation on function (7.6) also takes the RoadClass-value into account. As the route can be composed of edges with different RoadClasses, each segment has to be treated separate and all parts of evaluation values have to be summed:

  $$EV_6 = \Sigma_{i=0}^{\#\text{polylines}}(\text{Length edge}_i \cdot (\text{RoadClass}_i + 1)) \tag{7.7}$$

- From the results of the researches on matching multiple representation of street data, described in chapter 5, another two evaluation values can be defined:

  - Volz, 2006 has defined an approach for matching multiple representations of street data. This approach uses the average line distance, determined as the average distance of the distances of all vertices of two input lines:

    $$EV_7 = \frac{\Sigma D}{\#\text{vertices of route}} \tag{7.8}$$

  - A variation of this evaluation value can be determined by the maximum distance of the intermediate points of the route to the schematic link:

    $$EV_8 = \max D \tag{7.9}$$

With the implementation for matching the data set containing the schematic links and the data set containing the road and railroad network, the implemented variants will be evaluated in chapter 8.

CHAPTER 8

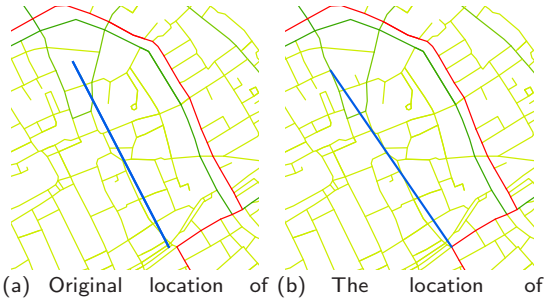
Results

The programme as mentioned in chapter 7 has been executed for a region of the Netherlands which contains both rural areas and urban environments, i.e. environments with both short and large schematic links. This will give representative results in order to be able to extend the results to the whole country.

With this evaluation an overview will be given of the results for determining the distribution of the potential mobile telephone network more accurately, in order to improve the input data for the TIGER tool.

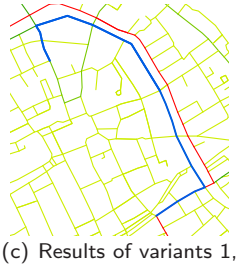
Although better input data could improve the results even more (as mentioned in chapter 4), an indication of the quality of the implemented variants will be given. Finally, the *best* variant for matching both input data sets will be indicated.

In section 8.1 the routes selected by the variants are first compared to the *ideal route*, and then the results of the variants will be compared with each other. During these comparisons also the reliability will be mentioned of the results.

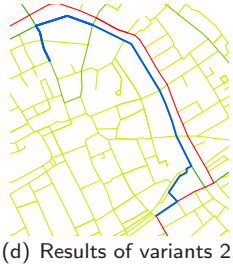


(a) Original location of schematic link 1

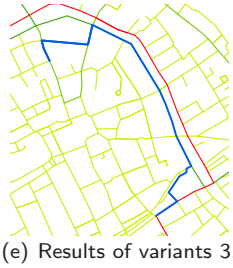
(b) The location of schematic link 1 after moving its vertices



(c) Results of variants 1, 6, and 8



(d) Results of variants 2, and 7



(e) Results of variants 3, 4, and 5

Figure 8.1: Results of a schematic link

The computer time for executing the variants are mentioned in section 8.2. In this section also is explained how to keep the computer time within reasonable limits. The expenses for using the programme of this research will be summed up in section 8.3.

8.1 Evaluation of the variants

In this section the results of the implemented variants will be manually compared, not automatically as was intended at the start of this graduation research. Figure 8.1 shows the results of the implemented variants for one schematic link. The other results which are used in the comparison in this section can be found at the end of this chapter, figures 8.3 to 8.6.

This sample survey is very small due to the time limit of this research. One can either take a larger number of schematic links to compare or one can compare the results of the variants to the schematic links that have been manually moved for an early improvement of the input data set for TIGER.

The selection of the best route for each variant has been performed automatically, by selecting the routes which have the lowest *Evaluation Value* (EV_i). The evaluation values of the variants have been calculated by the following equations:

$$EV_1 = \#\text{points of generalized route} \tag{8.1}$$

$$EV_2 = \frac{\#\text{intermediate points of generalized route}}{\#\text{intermediate points of original route}} \tag{8.2}$$

$$EV_3 = \frac{2 \cdot \text{Area}}{\text{Perimeter} \cdot (\text{RoadClass} + 1)} \tag{8.3}$$

$$EV_4 = \frac{(\text{Length route} \cdot (\text{RoadClass} + 1))^2}{\text{Area}} \tag{8.4}$$

$$EV_5 = \text{Length route} \tag{8.5}$$

$$EV_6 = \Sigma_{i=0}^{\#\text{polylines}}(\text{Length edge}_i \cdot (\text{RoadClass}_i + 1)) \tag{8.6}$$

$$EV_7 = \frac{\Sigma D}{\#\text{vertices of route}} \tag{8.7}$$

$$EV_8 = \max D \tag{8.8}$$

Characteristics of the ideal route

The *ideal route* is determined subjectively by human interpretation of the following properties, as already have been stated in section 7.8:

- The route should have a RoadClass that is as small as possible, because only the travelers of major roads are counted (when a major road is represented by a small number and a minor road is represented by a high number).
- The route should consist of as less as possible junctions, because major roads generally consist of less junctions then minor roads.
- The route should be as straight as possible, as major roads generally have less curves.

By searching for the route which fulfils the above properties best, the *ideal routes* have been determined for some schematic links, of which the results from the variants were different.

Comparison with the ideal route

The results of the variants can give an indication of the quality of the evaluation values. Especially when comparing the results of the variants to the *ideal route*, the variant with the highest matching rate can be pointed out. The ideal routes have also been indicated at the figures in this appendix, as well as in table 8.1(a). In this table, the variants which give the *ideal route* as the best match are indicated.

Table 8.1(b) indicates the number of times a variant has selected the *ideal route*. This table indicates variant 1 as the best variant for matching schematic links to the street network. Variant 1 determines the number of vertices of the generalized route. Comparing this variant with the properties as mentioned above, gives the following results:

– This variant doesn't consider the RoadClass of the route. A variation of the variant could be by multiplying the evaluation value of this variant with the mean Roadclass.

\+ The number of vertices of the route are considered, as the generalized route will have few vertices when the original route also has few vertices.

\+ The Douglas-Peucker line simplification algorithm gives an indication of the smoothness of a route, because only a few vertices will remain after generalization of a smooth route.

The other variant which gives good results is variant 6, which determines the shortest path with the least weight of the path. This path can also be determined by the Dijkstra algorithm, which determines directly the shortest path. This path can also be found by using the *Create Shortest Path Feature Task* as described in section 7.7. In the programme of this thesis first all routes are determined, in order to be able to execute all the variants.

Using the *Create Shortest Path Feature Task* will shorten the computer time, as the road network has to be examined less thoroughly. Only the route will be created with the lowest value, while all the other candidates are discarded during the search for the shortest route. This in contrast to the programme, where all routes are examined until they either have formed a route or are discarded because one of the threshold values has been exceeded.

Table 8.1: Comparison of the results of the schematic links with the *ideal* route

(a) Per schematic link indication of which variants have found the *ideal* route

Ideal route	Variants with this result
fig. 8.1(c)	1, 6, 8
fig. 8.3(f)	6
fig. 8.4(c)	1, 2
fig. 8.5(c)	1, 5, 6
fig. 8.6(c)	1, 6

(b) The frequency of variants which have found the *ideal* route

Frequency	Variant
4	1
3	6
2	2
1	5
1	8

Comparison with each other

A different comparison of the routes which are selected by the variants is, by determining which variants often find the same routes.

Figure 8.2 has been constructed in the following way. For each pair of variants, the frequency of choosing the same route has been determined. For example, variant 2 selects 3 times the same route as variant 1. The frequencies have been determined by comparing the results of figures 8.1, 8.3, 8.4, 8.5, 8.6, to each other.

In figure 8.2 can be seen that variants 1 and 2, and 3 and 4 frequently select the same routes. Below these pairs of variants are compared to each other, in order to explain the similar results of the variants:

$1-2:$ Both variants first generalize the route. Variant 1 uses the number of vertices of the generalized route directly for the evaluation value (EV_1), where variant 2 divides the evaluation value of variant 1 by the number of vertices of the original route (EV_2).

One can state with this comparison that the division by the number of vertices of the original route has only a small influence on the relative evaluation value determined for each route. In other words, a route with a lower EV_1 compared to another route, will most probably have a lower EV_2 than the other route.

$3-4:$ These variants both use the ratio between the area and the perimeter/length. However, the evaluation value of variant 3 (EV_3) has the area in the numerator of the devision, while the evaluation value of variant 4 (EV_4) has the area in the denominator.

Although the results of the ratios EV_3 and EV_4 result in selecting the same routes could be coincidence, one should consider that only a very small sample survey has been performed. In order to be able to make a better statement about the results of these variant, a larger number of schematic links should be examined.

8.2 Computer time

The computer times of table 8.2(a) are dependent on settings of the programme and on the input variables from table 8.2(b):

- The number of input polylines of the road network (in this case 60.000 polylines, what 5% of the total data set is). The computer time of *Union polylines of street layer* is greatly dependent on this amount.

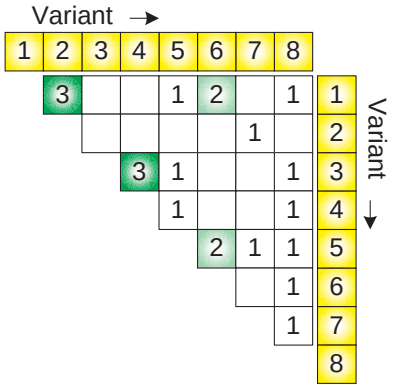


Figure 8.2: The number of matches of routes for corresponding links in the variants

- The number of schematic links, in this case nearly 300 links. All other parts of the programme, except the *Union of polylines*, are dependent of this number.

- The lengths of the schematic links have an influence on the computer time, because with longer schematic links, longer routes have to be found, which take a lot more time. Finding long routes involves searching a greater part of the street network, and thus a longer computer time.

- The variables from table 8.2(b) have the following influence on the computer time:

  - MaxMoveDistance – A larger value will involve a larger area to be searched for the *best* junction. With the current value, all vertices were moved to a junction from the street network, and the computer time is very reasonable.
  - Max#Routes – The value used here has never been reached. The number of routes which are found is also dependent on the next two values, because a larger area of the street network will then be examined.
  - Max#Polylines – With the current value of 25, not all schematic links are matched with a route on the road network. However, when more time is available for computations, a value of about 50 could be set, and probably all schematic links will be matched.
  - MaxDetour – This value has been set low, in order to shorten the computer time necessary for finding the routes. With the current value, not all schematic links will be matched with a route. For a real situation, this value should be about 100, in order to find routes for all schematic links.

The computer time of the *Union polylines of street layer*-step are long, because of the amount of polylines which have to be checked. Each polyline of the street network is examined on whether or not they are in line with another polyline, which has the same streetname, and no other polylines join the junction in which the compared polylines meet.

Finding and selecting the routes is also time consuming when the values of MaxDetour and MaxPolylines are not properly chosen. It is preferable to set these values quite small, and later set the values higher for the unmatched schematic links. This prevents long computer time for every schematic link.

The programme would run faster when a geometric network was used instead of a topology layer. A geometric network builds the network once, storing all relationships between the edges and points. In a topology only the topological rules are stored, while the relationships have to be determined on the fly. This process is time consuming and unnecessary to perform for each schematic link separately.

Table 8.2: Computer time per step from the procedure of the programme with the used input variables

(a) Measured computer times

Step	Computer time
Union	13,9 hr.
Duplicate	0,15 min.
SumCount	0,03 min.
Move	1,70 min.
FindRoutes and Select	3,3 hr.

(b) Input variables

Variable	Value
MaxMoveDistance	2000
Max#Routes	5000
Max#Polylines	25
MaxDetour	50

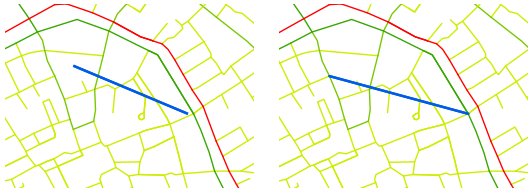
8.3 Expenses

All of the implemented variants use the same functionalities and input data. The expenses that are made when executing the programme are:

- The vector data set: €12.000;
- A single-user license for ArcInfo: €23.061.

A license for ArcView will not be sufficient, as functionalities for determining the incident edges through the topology of the network are not available with this license. The output of this programme could be improved by using the navigational additional data set, provided by Falkplan-Andes for €9.900.

These expenses give no indication of the charge for labor. For the evaluation of the results still a human operator is necessary to give an indication of the accuracy and reliability of the output of this programme. The charges for labor will be a considerable part of the additional expenses.
In the following chapter the conclusions of this graduation thesis will be drawn as well as possibilities for further research will be mentioned.



(a) Original location of schematic link 2

(b) The location of schematic link 2 after moving its vertices

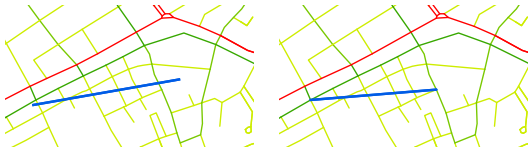
(c) Results of variants 1, and 2

(d) Results of variants 3, and 4

(e) Results of variants 5, 7, and 8

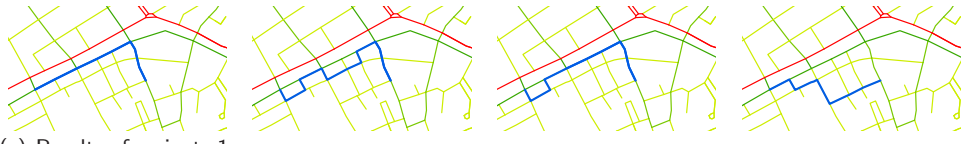
(f) Result of variant 6

Figure 8.3: Results second link



(a) Original location of schematic link 3

(b) The location of schematic link 3 after moving its vertices



(c) Results of variants 1, and 2

(d) Result of variant 3

(e) Result of variant 4

(f) Result of variant 5

(g) Result of variant 6

(h) Result of variant 7

(i) Result of variant 8

Figure 8.4: Results of third link

(a) Original location of schematic link 4

(b) The location of schematic link 4 after moving its vertices

(c) Results of variants 1, 5, and 6

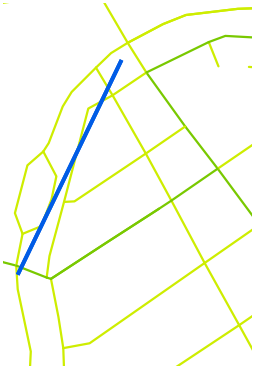
(d) Result of variant 2

(e) Result of variant 3

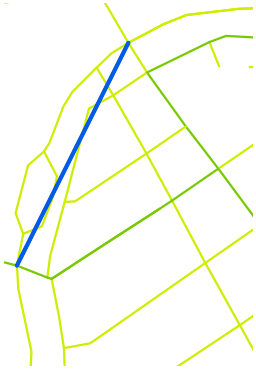
(f) Result of variant 4

(g) Results of variants 7, and 8

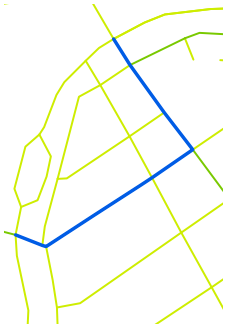
Figure 8.5: Results of fourth link



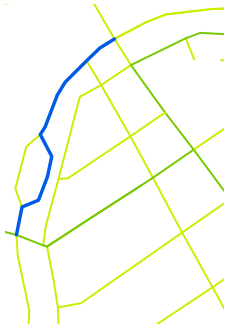
(a) Original location of schematic link 5



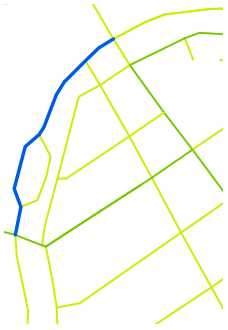
(b) The location of schematic link 5 after moving its vertices



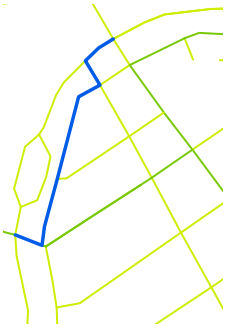
(c) Results of variants 1, and 2



(d) Results of variants 3, 4, and 8



(e) Results of variants 5, and 6



(f) Result of variant 7

Figure 8.6: Results of fifth link

CHAPTER 9

Conclusions and recommendations

This chapter gives the conclusions and recommendations with respect to this graduation research.

9.1 Conclusions

The following research question was central for this thesis:

> *Find the best match between generalized schematic links and the more detailed road and railroad network and give an indication of reliability and accuracy of the results.*

In order to be able to answer this question, three focus questions have been formulated, which will be answered separately:

1. *Which data sets are available for the implementation of the method?*
   In section 4.1 the available data sets have been described. These data sets are:

   - The *schematic links* – these links indicate the road or railroad segment which has been used by the travelers on the road or railroad network. Only the starting and endpoints have been stored.
   - The *road and railroad network data* – containing the true locations of all roads and railroads in the Netherlands. This network data doesn't contain information about one-way streets or fly-over junctions, but it does contain information about the kind of roads.

2. *Which methods can be defined to match the schematic links with the true road and railroad data?*
   In order to determine a method for matching the schematic links with the road and railroad data set, the following matching measures have been defined and imple-

mented:

$$EV_1 = \#\text{points of generalized route} \tag{9.1}$$

$$EV_2 = \frac{\#\text{intermediate points of generalized route}}{\#\text{intermediate points of original route}} \tag{9.2}$$

$$EV_3 = \frac{2 \cdot \text{Area}}{\text{Perimeter} \cdot \textsc{RoadClass}} \tag{9.3}$$

$$EV_4 = \frac{(\text{Length route} \cdot \textsc{RoadClass})^2}{\text{Area}} \tag{9.4}$$

$$EV_5 = \text{Length route} \tag{9.5}$$

$$EV_6 = \Sigma_{i=0}^{\#\text{polylines}}(\text{Length edge}_i \cdot \textsc{RoadClass}_i) \tag{9.6}$$

$$EV_7 = \frac{\Sigma D}{\#\text{vertices of route}} \tag{9.7}$$

$$EV_8 = \max D \tag{9.8}$$

From these measures, a total evaluation value can be constructed by e.g. calculating a weighted sum of multiple of the above evaluation values (described in section 5.3). Another way of deriving the total method is by defining threshold values and in an iterative process all but one candidate route can be discarded as described in section 5.4.

Both these total methods have not been examined in this thesis. In the other researches on matching multiple representations of street data, only one (Volz, 2006) gives values for the weights of the sum. The determination of the best combination of evaluation values and their weights need more research.

It has turned out that the implementation of a raster based method is redundant, for the following reasons:

- Raster based data is usually derived from vector based data sets, and thus the original (vector based) data should be preferably used.
- The use of raster based data is complex, because the cells have no indication of their relationships to each other, other than their attribute value. In order to retrieve these relationships, the raster data could be *vectorized*.

  For this process a license for ArcGIS could be bought, but the costs of both the raster based data set and this license are higher then the costs for buying the vector data set.

3. *What are the reliability, accuracy, and other characteristics of the implemented variants?*

   The *reliability* is the probability of performing a specified function without failure under given conditions. The reliability is dependent on the parameters of MAX#POLYLINES and MAXDETOUR (see for further explanation of these parameters chapter 7). When these parameters are set too small, no routes will be found which could represent the schematic link. In that case, the travelers on the road or railroad will not be relocated and the input data for TIGER will not be improved.

   A way of ensuring that different candidate routes will be found, is by using an iterative function for setting these values. In this case, the initial values can be chosen low and then increased. This way, matches can be found for each schematic link.

   The reliability of the implemented variants cannot be expressed by numbers. The evaluation of the variants have pointed out that the variant defined by equation (9.1)

is the best measure for matching the data sets. This variant generalizes the route with the Douglas-Peucker line simplification algorithm. The number of vertices of the route are considered, as the generalized route will have few vertices when the original route also has few vertices. The Douglas-Peucker line simplification algorithm gives an indication of the smoothness of a route, because only a few vertices will remain after generalization of a smooth route.

In science, engineering, industry and statistics, the *accuracy* is the degree of conformity of a measured or calculated quantity to its actual, nominal, absolute, or some other reference, value. The accuracy cannot be defined exactly, because no true situation is available for comparing the results of the variants.

The determination accuracy of the results of the matching process has instead be performed by comparing the results with a *ideal* route, determined by human interpretation of the characteristics of the schematic links which are described in section 8.1. For the data set of the whole Netherlands, a manual comparison isn't desirable. In that case, the results can be evaluated by using a weighted sum of multiple evaluation values.

The *computer time* is roughly the same for all of the implemented variants, and is greatly dependent on the values of the parameters MAX#POLYLINES and MAXDETOUR. These values determine the extend of the area searched for candidate routes.

The computer time can be reduced by using a geometric network instead of a topology layer in the feature data set. The geometric network stores the topological relationships of the network, while the topology layer only defines the topological rules for the features. When using a topology layer, the topological relationships have to be determined during the execution of the programme. The geometric network already has the relationships available in advance, therefore, the time consuming procedure of building the relationships can be skipped.

The expenses for executing the programme are roughly €35.000, for both buying the vector based data set and a single-user license of ArcInfo. In these expenses the charge for labor has not be included. These expenses will be a considerable amount, because the evaluation of the output of the programme still has to be performed by a human operator.

The results of the variants indicate that the Douglas-Peucker line simplification algorithm can also be used for other purposes than only for generalizing data sets. The algorithm can be well used as an indicator for the smoothness of lines and thus of the match between the schematic links and the road network data set. The Douglas-Peucker algorithm hasn't been used in previous researches for matching multiple representations of street data sets and the use of this algorithm gives a new evaluation value which could be used for matching data sets.

9.2 Recommendations

When choosing the *Evaluation value* as a quality measure (as described by Féchir and Waele, 2006) a global indicator can be defined, based on the satisfaction degree of some of the parameters. This can be executed by using the *best* functions evaluation values, and normalizing them. The global indicator can then be determined by the weighted sum of the normalized evaluation values, and their weights (e.g. dependent on the quality of the evaluation values according to this research).

By using the evaluation value from variant 1 together with some other evaluation values by calculating a weighted sum of these parameters, probably a better result can be achieved. Further research can for example concentrate on determining weight factors for these parameters in order to improve the determination of the distribution of potential mobile telephone users. The research of Volz, 2006 already determines some of the weight factors, which could be used as indicators of the weights.

The values MAX#POLYLINES and MAXDETOUR can be iteratively enlarged, when no routes have been found which could represent the schematic link. An iterative method prevents unnecessary searches for routes of obvious schematic links, and for the longer and/or more complicated schematic links, the values can be adjusted in order to find some routes.

The use of a better road and railroad network, which include one-way traffic and fly-overs, could improve the matching results. How this information could be used in the programme of this thesis could be further examined.

The influence of the output of this programme on the planning process has not been evaluated. TIGER should be executed once with the original schematic links and once with the output of this programme. Then both traffic grids from TIGER can be used in a planning tool. The differences of the results have to be determined. With this comparison an indication can be given of the degree of improvement of the traffic grid.

Bibliography

Arc4You. *Poly Clean 1.0.0.3*. WLM Klosterhuber & Partner OEG, 2003.
Visited at May 1st, 2006. www.wlm.at/Arc4You_Eng/A4clean/clean_main.htm#thinness

ArtWork. *Sliver removal*. ArtWork conversion software, inc., USA, 2006.
Visited at May 1st, 2006. www.artwork.com/gerber/gbrcomp/sliver.htm

Bi, Q., Zysman, G. I., Menkes, H., and Technologies, L. *Wireless mobile communications at the start of the 21st century*. IEEE Communications Magazine, 2001, 39(1):110–116.

ESRI, *GIS Topology, an ESRI White Paper*. ESRI Press, Redlands, USA, 2005a.

ESRI, *Help-file ArcGIS Network Analyst 9.1*. ESRI, USA, 2005b.

ESRI, *Help-file ArcGIS Spatial Analyst 9.1*. ESRI, USA, 2005c.

ESRI, *Help-file ArcScan: ArcGIS 9.1*. ESRI, USA, 2005d.

ESRI Developer Network. *Create Shortest Path Feature Task*, 2006a.
Visited at March 24th, 2006. http://edn.esri.com/index.cfm?fa=codeExch.sampleDetail&pg=/arcobjects/9.1/Samples/Editing/Edit_Tasks/Create_Shortest_Path_Feature/CreateShortestPathFeature.htm

ESRI Developer Network. *IPolycurve.Generalize Method*, 2006b.
Visited at April 10th, 2006. http://edndoc.esri.com/arcobjects/9.1/default.asp?url=/arcobjects/9.1/ComponentHelp/esriGeometry/IPolycurve_Generalize.htm

ESRI Developer Network. *Pathfinder*, 2006c.
Visited at March 15th, 2006. http://edndoc.esri.com/arcobjects/8.3/default.asp?url=/arcobjects/8.3/Samples/Network/Path%20Finder/Path%20Finder.htm

Farley, T. *Mobile telephone history*. Telektronikk, 2005, 3/4:22–34, Norway.

Féchir, A. and Waele, J. D. *Databases integration for supporting the future production of ign belgium generalised maps*. Institut Géographique National, Cartography Department. Proceedings of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data, 2006, Vol. XXXVI Part 2/W40, Hannover, Germany.

Harlow, M., Pfaff, R., Minami, M., Hatakeyama, A., and Mitchell, A., *Using ArcMap: ArcGIS 9*. ESRI Press, Redlands, USA. ISBN: 1589480988, 2004a.

Harlow, M., Vienneau, A., Bailey, J., Banning, J., and Woo, S., *Using ArcCatalog: ArcGIS 9*. ESRI Press, Redlands, USA. ISBN: 1589480996, 2004b.

Lee, W. C. Y., *Mobile cellular telecommunications: analog and digital systems*. McGraw-Hill Professional, New York, USA, 2nd edition. ISBN: 0070380899, 1995.

Mantel, D. and Lipeck, U. *Matching cartographic objects in spatial databases*. Database Group, Information Systems Institute, University of Hannover. Proceedings of the XXth ISPRS Congress on Geo-Imagery Bridging Continents, 2004, Istanbul, Turkey.

McMaster, R. B. *Automated line generalization*. Cartographica, 1987, 24(2):74–111.

Mustière, S. *Results of experiments on automated matching of networks at different scales*. COGIT Laboratory, Institut Géographique National. Proceedings of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data, 2006, Vol. XXXVI Part 2/W40, Hannover, Germany.

Pfaff, R., Booth, B., Shaner, J., Crosier, S., Sanchez, P., and MacDonald, A., *Editing in ArcMap: ArcGIS 9*. ESRI Press, Redlands, USA. ISBN: 1589481003, 2004.

Vaughan, J., Whyatt, D., and Brookes, G. *A parallel implementation of the douglas-peucker line simplification algorithm*. Software - Practice and experience, 1991, 21(3):331–336.

Volz, S. *An iterative approach for matching multiple representations of street data*. University of Stuttgart, Institute for Photogrammetry. Proceedings of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data, 2006, Vol. XXXVI Part 2/W40, Hannover, Germany.

White, D. *The Polygon Overlay Operation*. Environmental Protection Agency, 1997. Visited at May 1st, 2006. www.ncgia.ucsb.edu/giscc/units/u186/u186.html

Whyatt, J. and Wade, P. *The douglas-peucker line simplification algorithm*. The Bulletin of the Society of University Cartographers, 1988, 22(1):17–27.

Worboys, M. F., *GIS: a computing perspective*. Taylor & Francis, London, Great Britain. ISBN: 0748400656, 1995.

Zeiler, M., *Exploring ArcObjects*, volume 1 – Applications and Cartography. Esri Press, USA. ISBN: 1589480007, 2001.

Zhang, M., Shi, W., and Meng, L. *A generic matching algorithm for line networks of different resolutions*. Computing Faculty of A Coruña University – Campus de Elviña. Proceedings of the Workshop of ICA Commission on Generalization and Multiple Representation, 2005, A Coruña, Spain.

APPENDIX A

TNO

This graduation project has been carried out under the authority of the Netherlands Organization for Applied Scientific Research (TNO), at the business unit Information and Communication Technology (ICT). Therefore, a short description of this company will be given in this chapter. Figure A.1 gives an overview of the structure of TNO. The mission statement of TNO is:

> *TNO makes scientific knowledge applicable in order to strengthen the innovative capacity of business and government.*

TNO is a knowledge organization for companies, government bodies and public organizations. The daily work of some 5,000 employees is to develop and apply knowledge. The company provides contract research and specialist consultancy as well as grant licenses for patents and specialist software. TNO tests and certifies products and services, and issue an independent evaluation of quality. TNO also sets up new companies to market innovations.

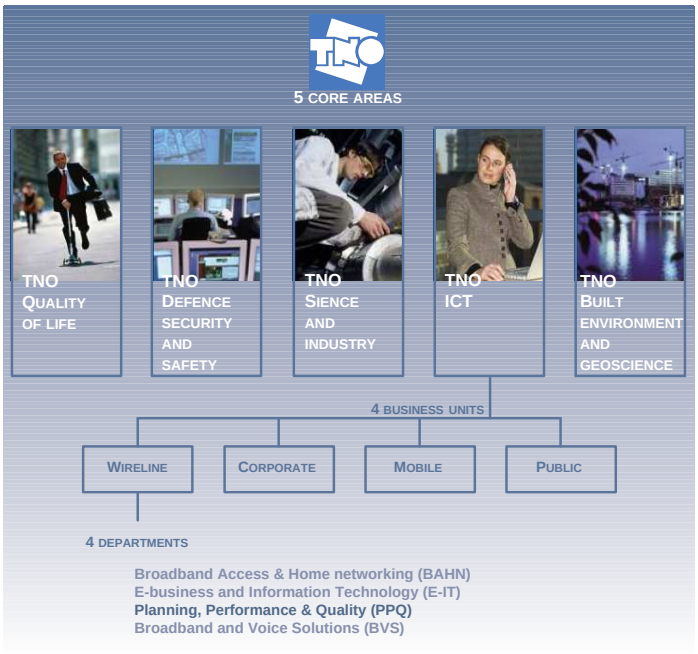


Figure A.1: TNO organization scheme

A.1 TNO Information and Communication Technology

TNO Information and Communication Technology (TNO ICT) is an unique center of innovation in the Netherlands that unites the ICT and Telecom disciplines of TNO. TNO ICT helps to realize successful innovations in ICT. Value creation for clients is the priority. The added value lies in the combination of innovative strength and in-depth knowledge. Research involves more than the technologies themselves. Where necessary, TNO ICT also focuses on user-friendliness, financial aspects, and business processes. The implementation process is supported by carrying out technical and market trials. They are also specialists in innovation strategy and policy, and the extensive ICT expertise is a valuable resource that can be used to address issues in the wider community.

A.2 Department Planning, Performance & Quality

The focus of Planning, Performance & Quality is to optimize the balance between costs and quality of ICT networks and services. In this way customers are able to guarantee quality requirements in a cost effective way. The expertise includes the development and application of models centered on performance analysis and network planning. It is based on an unique combination of both theoretical and practical knowledge of operation research and ICT networks. Developed products enable the customer to visualize decisions concerning design, parameters tuning and operational strategies. As a result decision-making is optimally supported. The department builds on a broad background of expertise, which makes it possible to flexibly adapt to new technological developments within the market.

APPENDIX B

Algorithms for matching spatial data from different sources

This appendix describes the initial algorithms from chapter 6 in further detail. For each algorithm, the steps are visually supported with an imaginary representation of a roadnetwork and schematic links which have to be matched to each other. In appendix B.1 the road network information is available as raster data; in appendix B.2 this information is stored as vector data.

B.1 Algorithms based on raster data

The following subsections will describe algorithms to match the schematic links to the road and railroad data from raster data sets. For these algorithms it is avoided to vectorize the road and railroad data, because the results of vectorization will be less accurate than the original vector data set.

The following four methods based on raster data all use the raster road and railroad data as it is, without vectorizing the data. Basically, the methods all use the vicinity of the pixels to the schematic links as a condition for matching both to each other.

1. The first method converts the (vector) schematic link to raster with a width. This process is also called *rasterization* and is the reverse process of *vectorization*. Each pixel representing a part of a schematic link is given the value of the COUNT (the number of travelers between the vertices of the schematic link) corresponding to the schematic link. Next the nearest road-pixel from the raster data is determined for each of these pixels (or e.g. one of every ten pixels). Optionally, an interpolation can be performed between the pixels with the same COUNT. A further explanation about this method, textual and visual, can be found in appendix B.1.1.

2. The second raster-based method, as also can be seen in appendix B.1.2, determines for every road-pixel the nearest schematic link. Thus, with this method *all* available road pixels will get an COUNT-value, in contrast with the previous method which only gives the COUNT-value to some of the road-pixels.

   In this case, when all road-pixels get a COUNT-value, also pixels with a large distance to the nearest schematic link will get a COUNT-value. To avoid this, a buffer can be used which represents the maximum distance between a road-pixel and a schematic

link. This way the roads, which are distant from any schematic link, will not get a COUNT-value.

3. Another method uses the theory of the Voronoi diagrams (or *Thiessen polygons*). A *Voronoi diagram* is the partitioning of a plane with n points into convex polygons, such that:

   (a) each polygon contains exactly one generating point, and

   (b) every point in a given polygon is closer to its generating point than to any other.

   The areas of closest proximity are polygons, and constitute a Voronoi diagram. When matching the schematic links to the pixels of the raster data, these areas of closest proximity can be used. In appendix B.1.3 a further explanation can be found. Basically, for each road-pixel the nearest schematic link is determined and its COUNT is assigned to the pixels. The results of this method will probably be the same as method 1, but the procedure for approaching the data is different.

4. The last method using raster data determines the areas of closest proximity by using the theory of bisectors. In each junction in the network of the schematic links all angles between the schematic links are divided in two and these bisectors are connected until they constitute the whole area. All closest proximity areas indicate which schematic link is closest by, thus the COUNT of this schematic link can be assigned to each road-pixel in this area. This process is further described in appendix B.1.4.

B.1.1 Rasterize schematic link

The fundamentals of this algorithm is rasterizing the schematic links. The steps to be taken are explained below and illustrated in figure B.1, the numeration of both explanations correspond:

1. The initial stages of both data sets.
2. Rasterize the schematic links and assign the number of vehicles (COUNT) as the new pixel value and store these pixels in a new layer.
3. The layer with the vector schematic links is removed.
4. Determine for each pixel of the schematic links the nearest road pixel.
5. Assign the COUNT belonging to the schematic link as a value to the road pixels, in a new layer.
6. The new layer with only a few pixels with number of vehicles. The layer with the road network data set is removed.
7. Interpolate between the pixels, that are closest to each other and have the same pixel value, by determining a straight line between the two pixels and rasterize this line. Then assign the COUNT to the interpolated pixels.

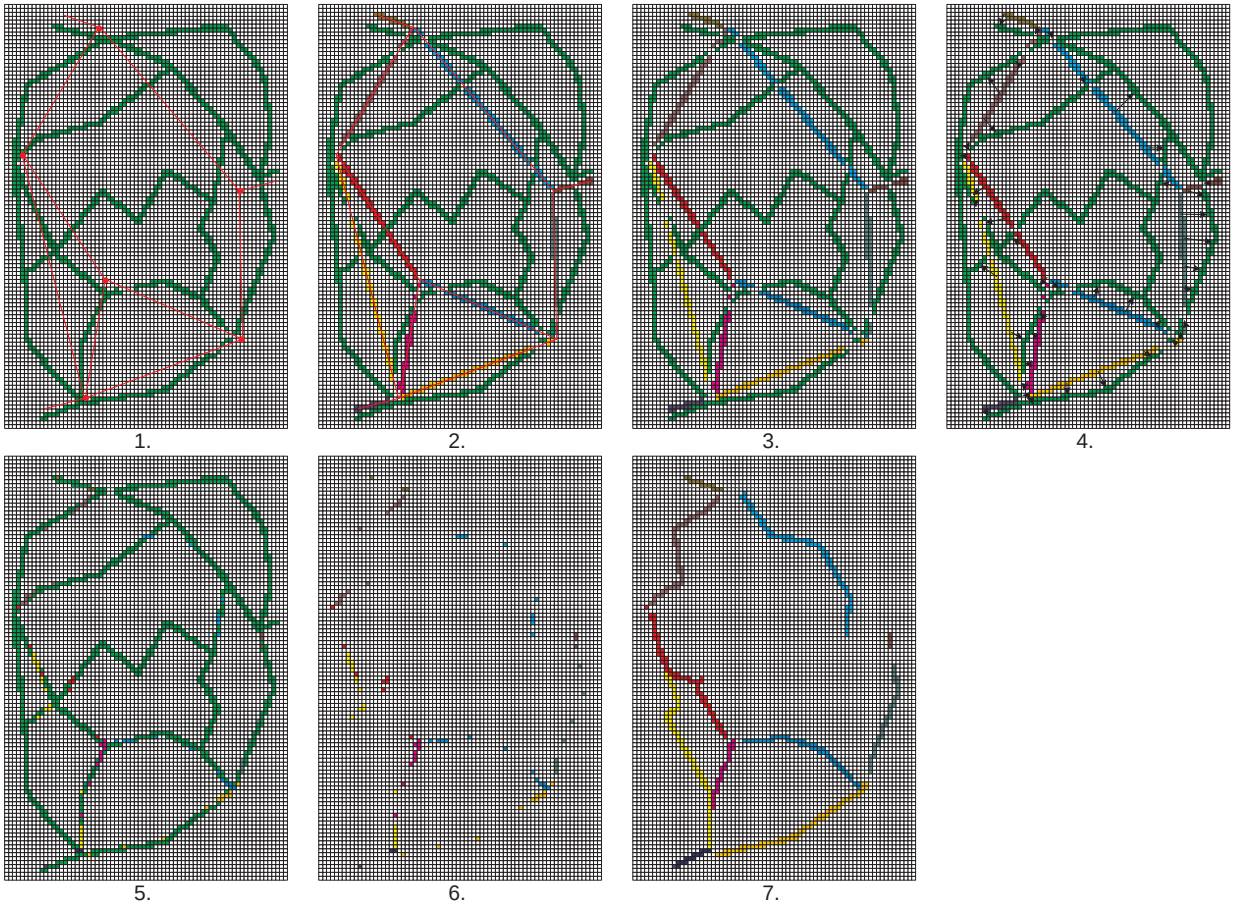


Figure B.1: Rasterize schematic link

B.1.2 Nearest schematic link

This algorithm determines for every road pixel which schematic link is the closest. If desired it is possible to use buffers to eliminate road pixels that have a large distance to the nearest schematic link. In figure B.2 the steps of algorithm 2 are explained:

1. The initial stage, containing the schematic links and the raster-based road data set. The schematic links are colored according to their COUNT.
2. Determine for every road pixel the nearest schematic link and give the number of vehicles of the schematic link to the road pixel in a new layer. Because all the road pixels have been given a value of a link, all original road pixels are still present in the final result.
3. To eliminate pixels that are unlikely to be good road pixels matching with a schematic link, create buffers around the schematic links.
4. Eliminate the pixels that are not contained in one or more buffers. These pixels are too far away from any original schematic link.

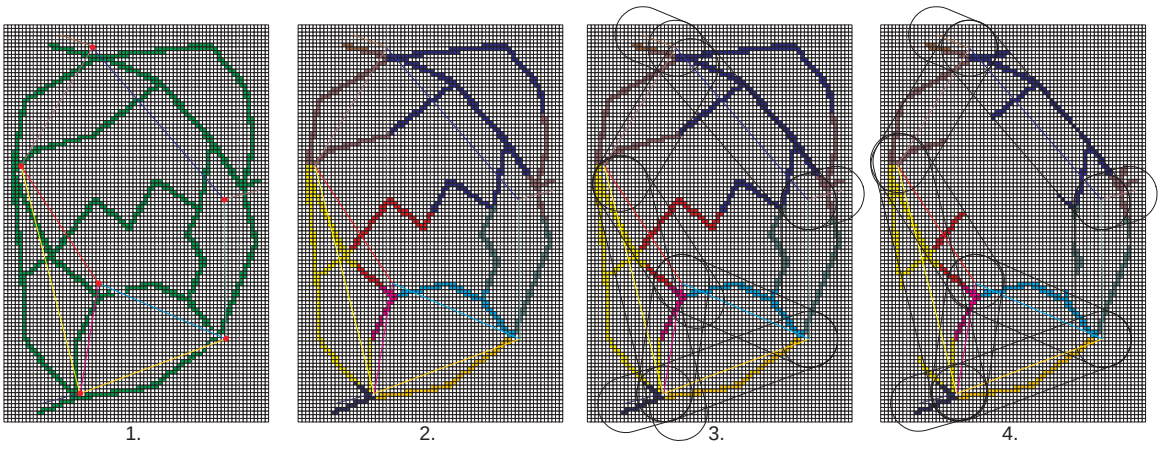


Figure B.2: Nearest schematic link

B.1.3 Voronoi diagram

For matching the schematic links to the road network, the definition of *Voronoi* diagrams does not directly serve the purpose, because the objective is to determine which schematic link is closest to a pixel. Therefore, instead of using the vertices of the schematic links as generating points, the schematic links themselves have to be used as the condition that each polygon of the *Voronoi* diagram contains exactly one generating point. In order to achieve this, the center of the schematic links are used as generating points. How the algorithm will be explained further, is stated below and illustrated in figure B.3:

1. The initial stage of both networks.
2. The centers of the schematic links are determined and the COUNT-value is assigned to these points.
3. The centers are connected by the specifications of a *Delaunay* diagram, the straight line dual of the Voronoi diagram.
4. The Voronoi diagram is derived from the Delaunay diagram.
5. The COUNT-value is assigned to the pixels in the convex polygons of the Voronoi diagram by intersecting the Voronoi diagram with the raster data set.

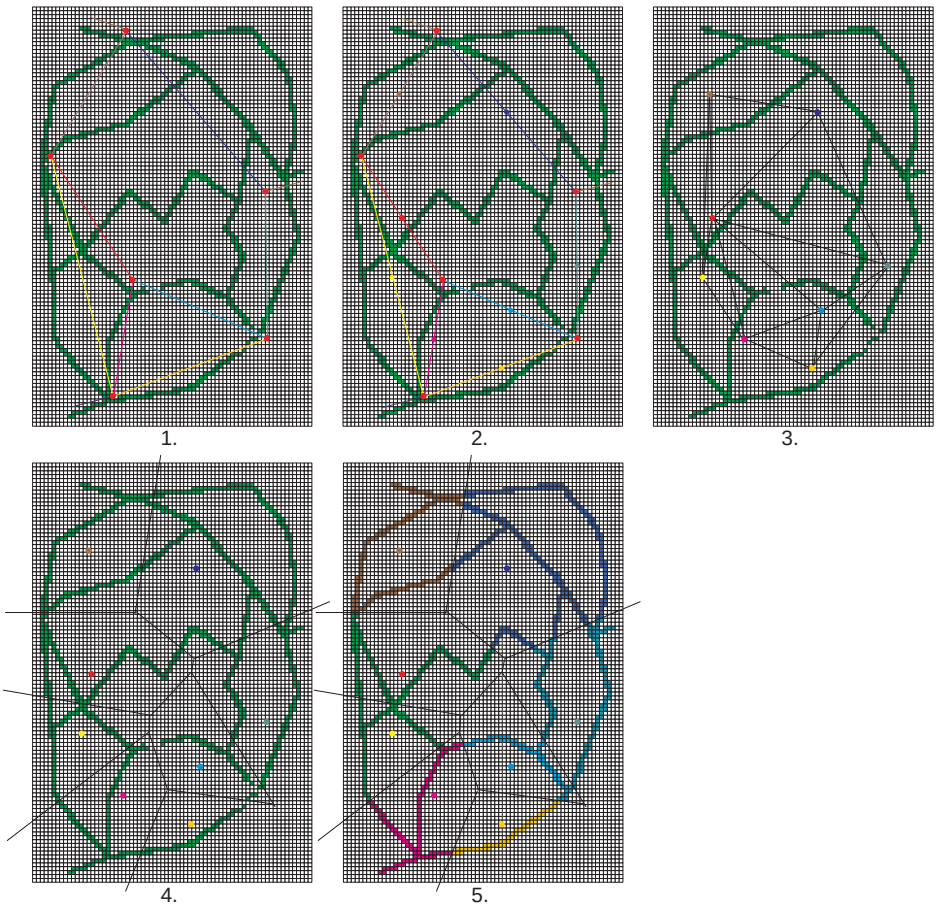


Figure B.3: Voronoi diagram, with the middle of the schematic links as input points

B.1.4 Bisectors

This algorithm uses the theory of bisectors of the triangles in a constrained triangulation of the schematic links for determining the nearest schematic links for each road pixel. Figure B.4 shows the schematic overview of the processing steps of this algorithm and the steps are as follows:

1. Initial drawing containing the road pixels of the raster-based data and the schematic links with the number of vehicles between starting and endpoints.
2. Create a triangulation for the schematic links.
3. Determine the bisectors of each angle in the triangles and intersect the bisectors in each triangle.
4. Remove the layer with the schematic links and the road pixels that lay in the areas where extra segments were added for the triangulation. In each area that is created by the bisectors all the road pixels get the value of the schematic link that was part of this area.

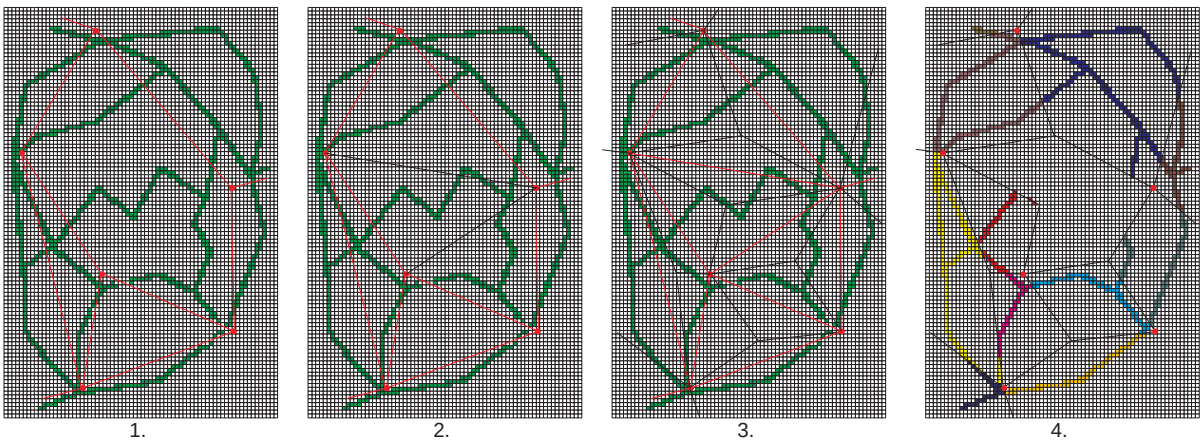


Figure B.4: Bisectors

B.2 Algorithms based on vector data

In the following subsections a vector data set with the road network is used. In this data set the kind of road is indicated by the value of the RoadClass. In the algorithms no account of the RoadClass has been taken as this has no actual effect on the procedure of the algorithm, only on the results when it's implemented.

Further it is assumed that the vertices of the schematic links are already moved to junctions in the road network. An algorithm for executing this is explained in section 6.1. This way the vertices don't have to be matched with junctions, thus matching the schematic links with the road network can directly be executed.

In the beginning of this graduation project, the following algorithms were separately defined. However, during the execution it became clear that implementing most of the algorithms could be easily implemented by making only a few adjustments, because only the selection criterium for selecting the best route is different.

B.2.1 Smallest surface areas

In this algorithm the areas are determined that are created by the schematic links and route that has been found on the road network, see figure B.5:

1. Sketch of the imaginary road network and schematic links, with the vertices of the schematic links moved to junctions in the road network.
2. The areas are calculated by intersecting the schematic link with the route and calculating absolute value of each subarea. In step 2 in figure B.5 only for the routes creating the smallest area between the route and the corresponding schematic link are shown.
3. The final result is indicated by red polylines.

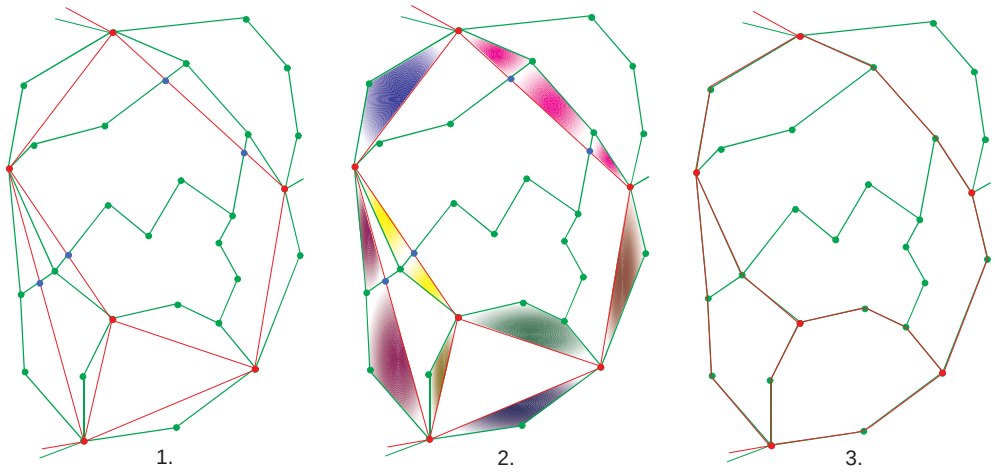


Figure B.5: Smallest surface areas

B.2.2 Shortest path

This algorithm calculates the length of the path, with or without taking the RoadClass into account:

1. The length of all polylines of the road network are shown, in order to give an indication of the length of all possible routes,
2. The routes determined by the shortest path are shown.

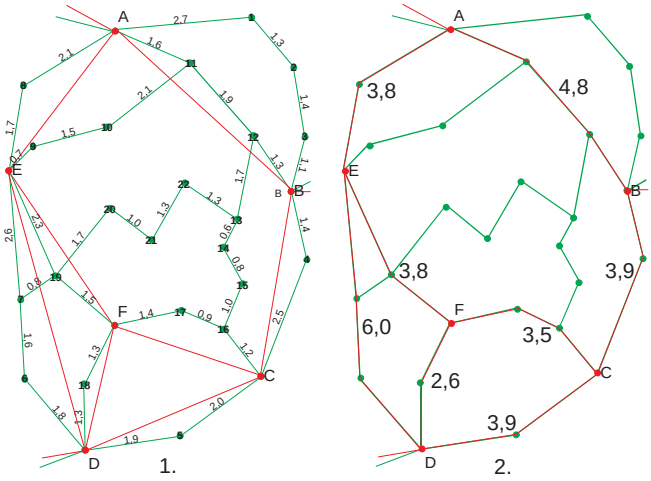


Figure B.6: Shortest path

B.2.3 Smallest angles

Another condition for matching the schematic link to a route could be by selecting the route with the smallest angles. How this algorithm could be executed, is explained in figure B.2.3 and the following steps could be performed:

1. The initial state containing the imaginary road network and the schematic links. The numbers indicate the ObjectID of the schematic links. Further, the smallest angles, between the schematic link and the road network in the starting and endpoints of the schematic links, are indicated by small arcs.
2. The polylines of the road network get the ObjectID of the schematic link.
3. For each schematic link ObjectID the ends are connected. This can result in two cases:
   (a) the total route coincides with the road network. In this case the route is stored.
   (b) the route doesn't coincides with the road network. At the ends, again the angles are determined and the polylines with the smallest angles are selected.
4. The situation after repeating the previous steps until for each schematic link a route has been stored.
5. The final routes are indicated by the red polylines.

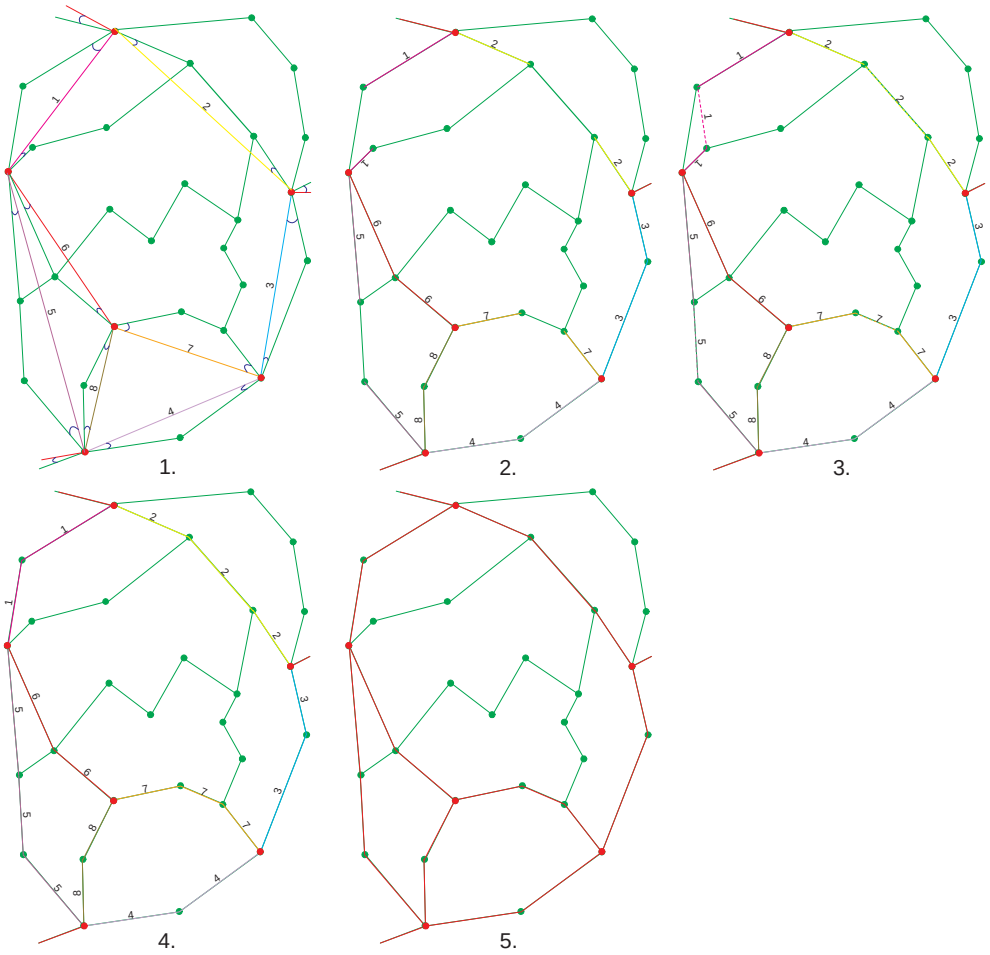


Figure B.7: Smallest angles

B.2.4 Douglas-Peucker

Background to this algorithm is the Douglas-Peucker line simplification algorithm. Another line simplification algorithm could also be used, but as the Douglas-Peucker algorithm has already been implemented as a standard function in ArcGIS this algorithm has been used. The name of the function is *IPolyCurve.generalize* (ESRI Developer Network, 2006b).

The Douglas-Peucker algorithm is recursive and will be repeated until the distance of all junctions of the route to the schematic link is smaller than the tolerance parameter. Below the Douglas-Peucker line simplification algorithm is explained by executing it for the imaginary road network. The algorithm is only performed for one schematic link, but three different candidate routes ($2-5$, $6-7$, and $8-12$):

1. The initial stage with the road network and one schematic link. For this schematic link the Douglas-Peucker line simplification algorithm is executed for different possible routes.
2. The first candidate route is generalized by calculating the distances from the intermediate vertices of the route to the schematic link. In this case multiple distances are larger than the offset parameter (indicated by the buffer), so the vertex with the largest distance is selected.
3. The selected vertex is now part of the line and the distances from the vertices that are left to the line are calculated and compared with the offset parameter. Again the vertex with the largest distance is selected. For the other part of the line the distance of the intermediate vertex is smaller than the offset parameter and is thus ignored.
4. This process is repeated until all distances are smaller than the offset parameter.
5. The generalized route is shown in red. In this example the generalized route only consists of 2 intermediate vertices, instead of 4 in the original route.

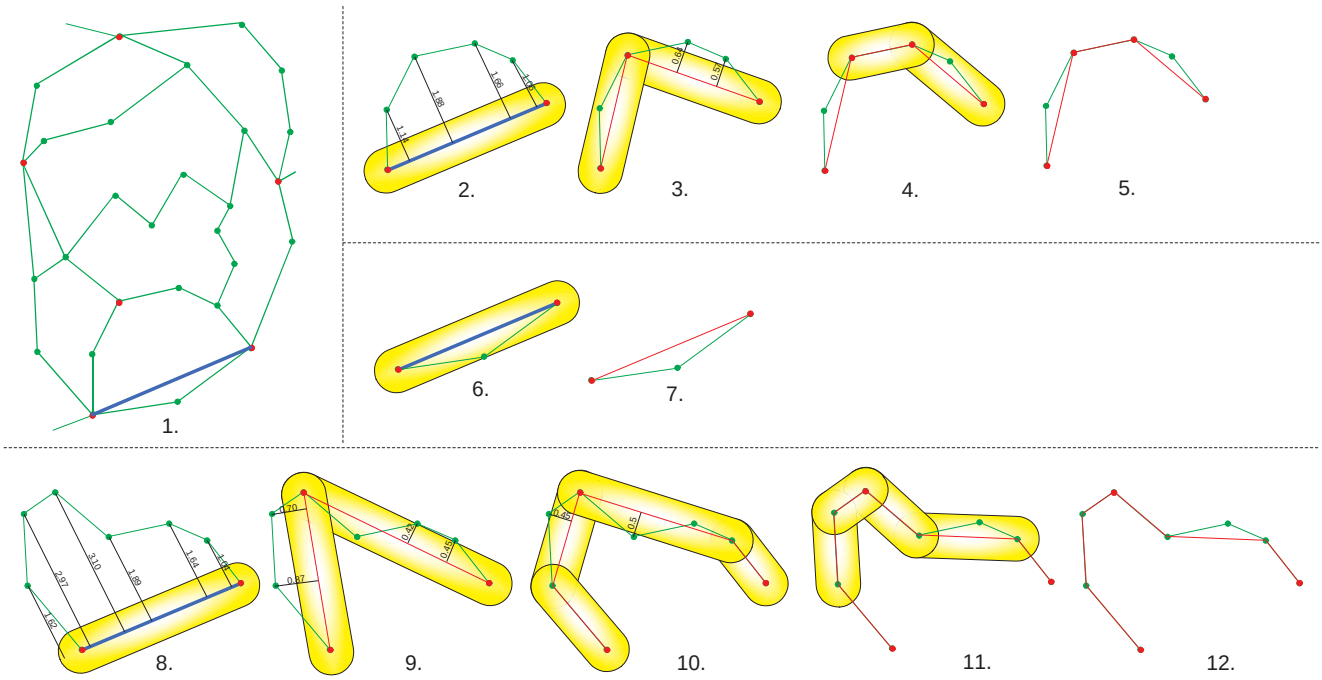


Figure B.8: Douglas-Peucker

6. The second route is selected and the distances of the intermediate vertices are calculated.

7. Immediate all vertices lie inside the buffer of the offset parameter, thus no other vertices are selected and the straight line represents the route as the generalized form.

8. The third route is selected and all distances of the intermediate vertices to the schematic link are calculated.

9. The vertex furthest away has been selected and is now part of the generalized route. The distances of the remaining vertices are calculated and for both parts of the generalized route a vertex is selected.

10. The procedure is repeated until all vertices of the original route satisfy the offset parameter.

11. The final stage of the generalization, all vertices are within the buffer of the offset parameter.

12. The final result containing the generalized route.

By using the predefined function *IPolyCurve.generalize* only the original route and its generalized form can be evaluated. In the function the tolerance parameter can be set and changed if desired. Altering the tolerance parameter can result in more or less generalized routes, and gives an indication of the smoothness of the route.

After the generalization of the route, the number of intermediate points that are left will be compared to the number of intermediate points of the original route. This will give an indication of the smoothness of the original route and thus this will give an indication of the quality of the match.

B.2.5 Reversed Douglas-Peucker

This algorithm will generally do the reverse of the Douglas-Peucker algorithm, described in the previous section (B.2.4). It will evaluate the number of steps that are necessary to construct the route from the schematic link. An offset parameter is used to select junctions from the road network in a recursive procedure:

1. The initial stage, with the buffer constructed from the offset parameter. The junctions from the road network within a buffer are selected.
2. The junctions are connected with straight lines.
3. The process of the previous step is repeated for each straight line that doesn't coincide with the road network.
4. This procedure is repeated until all parts of the route coincide with the road network.
5. The final result of this procedure.

A disadvantage of this algorithm is that it cannot be used when the routes are already determined. Therefore, this method cannot be used as a variant in the implementation. It can however be used to directly detect the best routes.

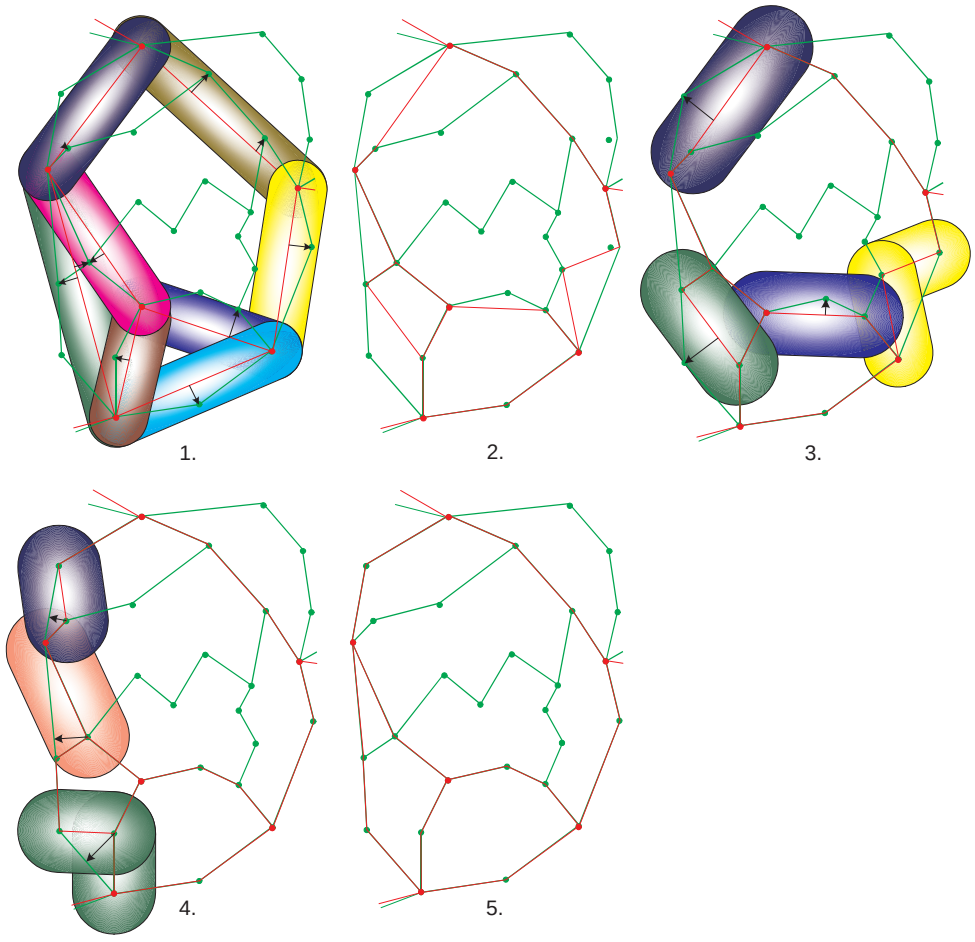


Figure B.9: Reversed Douglas-Peucker

B.2.6 Distances of intermediate route points

The distance of the intermediate points of a route to the schematic link gives an indication of the quality of the match. In figure B.10 the maximum distance is determined:

1. The initial stage containing the road network and the schematic links. For all candidate routes the maximum distance from the intermediate route point to the schematic links are determined. In the figure the buffers indicate this distance as when all vertices of the best route are within the buffer, this buffer indicates the maximum distance.

2. The selected routes are shown in red while the original road network is still shown in green.

A variation of this algorithm is by, instead of determining the maximum distance, determining all distances and calculating the average *route distance*. This value is small for routes which are close to the schematic link and it is large for routes that are far away. In the last case, it isn't likely that the route is a good match, thus this route will be discarded while the route with a small value will be saved.

Another variation is by determining the total of the distances. The resulting value is dependent on the distance of the route to the schematic link, and on the number of intermediate points. One of the properties of the schematic links is that the route should be as smooth as possible and contains as less as possible intermediate points. In this variation, both properties are taken into account.

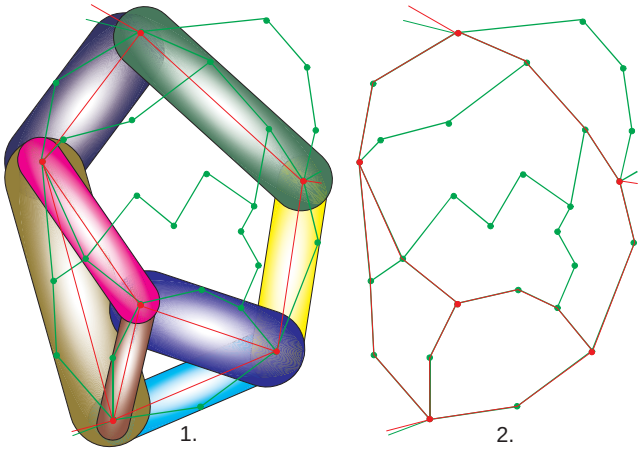


Figure B.10: Maximum distance of intermediate points in the route

APPENDIX C

The programme

This appendix consists of a CD which contains the following files:

- Traffic.mxt – The template containing the programme implemented for this thesis.
- Schematiclinks highways.shp – The shapefile containing the schematic links of the *highways* of the whole Netherlands.
- Schematiclinks minor roads.shp – The shapefile containing the schematic links of the *minor roads* of the whole Netherlands.
- Schematiclinks all roads.shp – The shapefile containing all schematic links representing roads of the whole Netherlands.
- Schematiclinks railroads.shp – The shapefile containing the schematic links of the *railroads* of the whole Netherlands.
- Two files contain the VBA-code from the template. These can be used when ArcGIS is not available and one wants to view the code. ThisDocument.cls contains the code of the commands from the traffic-toolbar. The file named frmSpecifications.frm contains the code of the form of chapter 7 and all functions belonging to the form.
- PathFinder – This COM object demonstrates several Network interfaces and presents a simplified interface for finding the shortest path through a set of points. You can register and use this DLL as is or customize it further for your application (this object has been used in section 7.7).
- CreateShortestPathFeature – The purpose of this sample is to create a new polyline feature that correspond to the Shortest Path between the two input points. The sample is using the Dijkstra algorithm with a weight corresponding to the Euclidian distance between nodes of the graph. The Planar Graph used here is the Topology Graph (this object has been used in section 7.7).
- Street – This folder contains the road and railroad network for executing this thesis.

Before using one or more schematic links data sets, these data sets first have to be imported in a feature data set in a personal geodatabase. Further, the vector data sets containing the road and railroad network have to be imported as a *new Topology* in the feature data set which contains the schematic links. These actions both have to be performed in ArcCatalog.

APPENDIX D

Glossary

Junction A crossing in the true road or railroad network.

Road The true location of roads or railroads. In most cases both roads and railroads are mentioned to decrease the amount of text.

Route The topographical location of the connection between the vertices from the input traffic data. The route connects the vertices of the schematic link through the road and railroad network.

Schematic link The straight line which represent the relation between the vertices from the input traffic data.

TIGER Traffic Intelligence through Geographic Extrapolation. A tool for planning the mobile telephone network.

Traffic The mobile telephone traffic or users of the road and railroad network. In the text the distinction will be made clear.

Vertex The starting or endpoint of a schematic link.