Geographical Information Management and Applications

Driving your Way – The Building of a Speed-influenced Navigation System

MSc Thesis – Final Report
Sybren Frederico de Jesus Madureira Porfírio BSc
Student Number: 4174682
Email Address: sfporfirio@hotmail.com
Supervisor: L.C. Vreugdenhil MSc
Responsible Professor: dr. ir. R.J.A. van Lammeren
February 2018

Universiteit Utrecht
TU Delft
UNIVERSITY OF TWENTE.
WAGENINGEN UR

Abstract

Contemporary navigational systems have become better over the years, but have kept a focus on calculating optimal paths, i.e. the fastest, shortest, most fuel efficient, most touristic, etc. path. Little to no attention is given to offering users multiple options, where the final decision would depend on their own priorities. Doing so could raise awareness as to the argumentations behind such choices, which is a goal of the current research, focusing on how adhering to different maximum speeds could influence time and fuel consumption.

The main question to be answered was: *"To what extent does driving at different maximum speeds influence the time and fuel consumption for fastest car routes?"* In order to answer this question, the following mission statement was set: *"To create a website that allows for calculating and comparing the car routes for different custom maximum speeds."*

Before building this website, a literature study into fuel consumption of cars was performed. By doing this, requirements for the website and the data to be used were determined, after which a blueprint for the system was drawn. After building the actual application, it was used to calculate 135.000 paths between 750 pairs of randomly selected departure and destination points within the Dutch province of Zuid-Holland, along with the time and fuel consumed for every path. Using a Monte Carlo analysis, these results were analysed statistically.

It can be concluded that maximum speeds do indeed have a moderately strong influence on time consumption, and a weaker one on fuel. When comparing time and fuel over speed, it can be seen that time savings per litre extra use diminish over time. Varying car models in turn has a strong influence on fuel and none on time, whereas traffic situations only affect both consumptions slightly and dynamic speeds have no notable influence on either.

Though by far the most important factor in determining time and fuel used is the location of the point of departure and the destination, maximum speeds' influence is still large enough for it to be relevant for drivers wishing to make an informed choice on which route to take to wherever they need to get to. As such, the proposed model is deemed useful for this purpose.

Acknowledgements

To all those who supported me directly in writing this master thesis, including my supervisor Corné Vreugdenhil, who was always available to discuss any thought – both mine and his – and whose enthusiasm for this topic at times managed to surpass even my own,

To the *Dreiflüssestadt* and its great people *en marche*, for the incredible friendship and hospitality that was bestowed upon me during my time here, and for the inspiration, motivation and creativity that it brought me at times when it was needed most,

To all those other people, both family and friends, who made arriving at this result possible:

Bedankt, *danke*, *obrigado* and thank you.

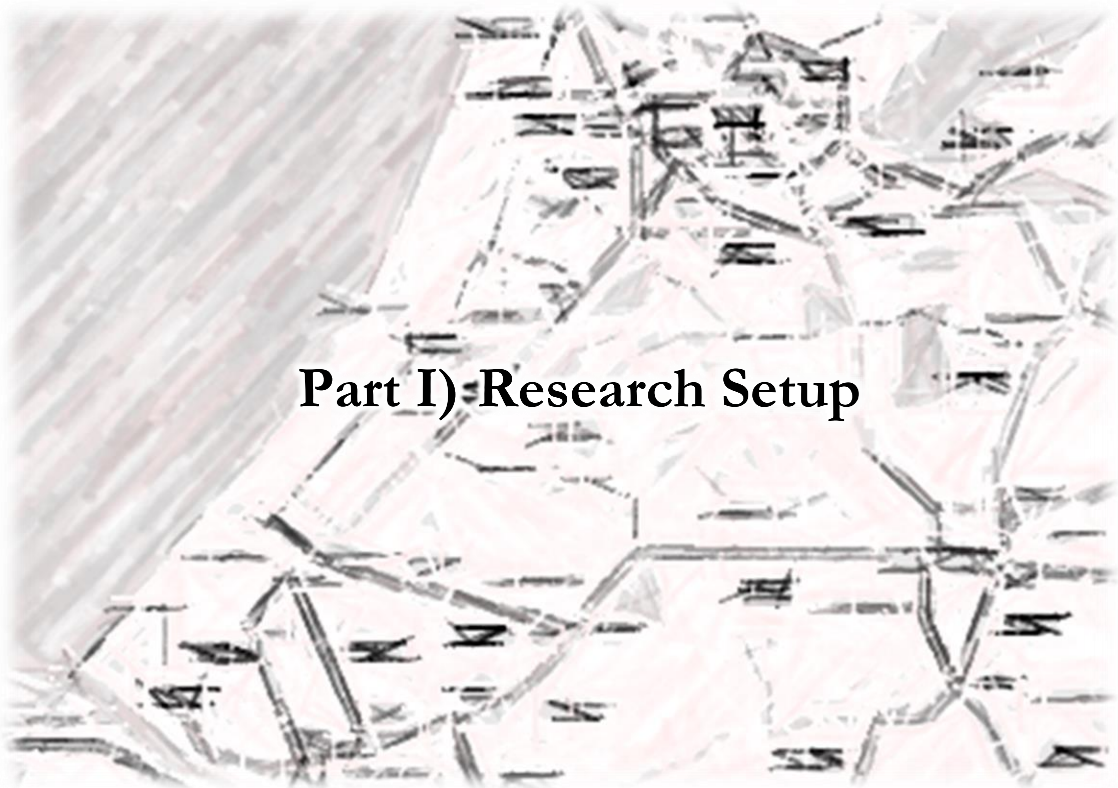
Sybren Frederico de Jesus Madureira Porfírio,
Passau, 20/02/2018

Reading Guide

Every reader has his own goals when reading, or might be looking for different information. To facilitate this process, this short guide has been written on what to expect where. The numbers within parentheses indicate the chapter numbers.

- The first two chapters, **'Problem Context (1)'** and **'Research Objectives (2)'**, discuss the context in which the current study was conducted, the reasons for conducting it and what exactly was to be done.
- The next chapter, **'Methodology (3)'**, then, focuses on the steps of the research and the scientific methods that were used. The **'System Requirements & Blueprint (6)'** chapter in turn discusses the more practical techniques that were used in the actual development of the application.
- The **'Car Characteristics Influencing Fuel Consumption (4)'** and **'Car Model Development (5)'** chapters delve deeper into theories behind fuel consumption, building a framework for this consumption to be used in the remainder of the study and using this framework to create five examples of car models.
- The **'Database Creation (7)'** and **'Website Creation (8)'** chapters can be regarded as technical reports on the creation of respectively the database and website. This information should be useful to anyone wanting to replicate or improve the work done here. It can be seen as a process report on both creations as well.
- The **'Use Case (9)'** chapter, then, is a demonstration in written form of the built application. It sketches a scenario of a possible user of the website and goes through his use of it. Doing this, the results as returned by the website are interpreted in a way in which this fictional user could have interpreted them, leading to a choice for a specific path, based on his situation.
- The final chapters, **'Results (10)'**, **'Conclusion (11)'** and **'Discussion (12)'**, take a look at the results as generated by the application by analysing these statistically. This allows for answering the questions set initially and for drawing conclusions, before discussing what all of this means in practice.
- The **'Glossary (Appendix II)'** provides information on abbreviations and other terms used throughout this study.

Table of Contents



Part I) Research Setup

1) Problem Context

Nowadays, it is nearly impossible to imagine a world without navigation systems. Whereas for decades, motorists had to find their way using maps which may or may not have been correct and up-to-date, anyone can now easily get to where they need to be by simply following directions. The first attempt at creating such a system dates from 1966, when the research department of car manufacturer General Motors developed 'Driver Aid, Information & Routing', or 'DAIR'. Using punch cards, radio communication and magnetic objects buried beneath roads, this device would tell drivers whether to head straight on, or take a left or right turn. While it was never introduced commercially, it was a starting point for others wishing to build such systems. The real breakthrough came in 1994, though, when the American Global Positioning System (GPS) became fully functional. It allowed for navigation by satellite, mostly using dedicated devices. Nowadays, such systems are also available on smartphones, online and built into several car models, such as those manufactured by Citroën and BMW (The New York Times, 2013; Lammertsma, 2005; Citroën, 2013; BMW, 2013).

Finding a path between A and B can be done in many different ways, though. In many cases, the fastest one will be the one sought after, but the shortest one can often be found as well. In fact, the factors that paths can be optimised for are only limited by imagination: Michelin (2017) for example provides a 'discovery' route which tries to incorporate as many scenic locations as possible. Increasingly, though, drivers are starting to care more about their fuel consumption, be it for financial reasons or worries as to emissions. As Zhu et al. (2017) describe, such 'economic' route planners try to find the most fuel-efficient path by looking at factors such as trip length, grades of roads, current traffic and average possible speeds. Norouzi et al. (2017) add the physical condition of roads and air drag to that list. Finally, Zhou et al. (2016) mention the importance of keeping driving speed as constant as possible: these are all factors that, when possible, should be taken into account.

Even though speed can be a factor in finding certain paths, several useful applications do not seem to have been investigated yet. In fact, existing efforts, such as the ones mentioned in the previous paragraph, tend to focus on the influence of the paths on the driving speed. They do not however consider the exact opposite: the influence of speeds upon paths. It is very well possible for drivers to not always want to drive as fast as legally possible – e.g. driving 100km/h on a 100km/h road – but to drive slower, say 80km/h on the aforementioned road: they have total control over the speed with which they travel. Whether or not they will choose to go as fast as legally possible, will depend entirely on their own priorities at that moment. In any case: they are the ones deciding on the speed, and thus have influence on fuel consumption. Adhering to a custom maximum speed will change some optimal paths: a detour via a highway might not be worth it anymore for example when keeping to 80km/h, as other roads with lower limits might be a shorter and thus faster alternative.

In addition to these possibly shorter paths, it is to be expected that lower speeds will lead to longer travel times, but what would it mean for fuel consumption? The influence of these custom maximum speeds on time and especially fuel consumption is the focal point of this research project. Some habits, such as driving at a constant speed, are known to have positive effects on fuel consumption. Having a path with as little traffic lights and of course traffic as possible, will thus be preferred. It is likely for other factors to influence this too, though. Therefore, the initial phase of this research will focus especially on listing these relevant factors (Het Nieuwe Rijden, 2017).

As can be understood from the above, saving fuel will require driving differently than normal – 'normal' being always driving as fast as possible. Drivers will need to adopt a certain behaviour, which might not be what everyone wants. It is however pivotal to mention that all of this comes down to personal preferences and circumstances: those who attribute more value to speed rather than fuel savings or those who simply need to arrive on time, will likely stick to the fastest way of driving. The aim is not, however, to have everyone adhere to one single, 'optimal' way of driving. It is, rather, to provide information on the trade-off between time and fuel for different custom maximum speeds. As with the traditional fastest/shortest route option, it is the driver who ultimately chooses which path to follow, a decision which is purely based on his own circumstances. Of course, these might change from day to day: trying to get home from work on a Monday evening, an employee will most likely want the fastest way possible, whereas he may prefer a slower but more fuel-saving ride when going for a Sunday drive. The ultimate decision lies solely with the user: the system will only present him with the alternatives.

Staying with the topic of practical relevance, it is not only individual users who could benefit from such a system's use. One line of business that could especially do so, are logistics companies. A study by ING (2013) found that roughly a quarter of such companies' costs are fuel costs. The same report highly recommends them to make sure that their drivers adhere to a more fuel-friendly way of driving. Another study by TNO (2013) shows that such savings could amount to up to 10% of fuel costs. In addition, logistics companies themselves have repeatedly expressed that they intend to have their drivers do so and that they are already putting effort into making them more fuel-aware (DuurzaamBedrijfsleven.nl, 2016).

As such, the practical relevance is clear: knowing custom speeds' influence on driving could save fuel, and thus money, which could potentially be interesting for every single driver. Knowing what the alternatives are for getting to your destination leads to better informed choices. A scientific relevance is present too, though. As mentioned before, speeds' influences have not yet been explored to their full potential, as research only focussed upon habits such as keeping a constant speed. Implementing speed from the beginning into pathfinding will allow for the further development of this line of study: if proven to have a notable impact, it could be factored in for future research. It should be stressed that the current research will only look at the impact for cars. The same could be done for other vehicles, such as trucks or buses, too though by repeating this study for those modes of transport.

Another area which could benefit is the development of self-driving, i.e. autonomous, cars. Relying no more upon human drivers, such cars will have to find their way themselves using navigation systems, as pointed out by Tao et al. (2017). As such, autonomous cars can only ever be as good as their pathfinding algorithms. It will still be the passenger who will decide on how to get to a destination, though, making presenting them with alternative routes essential. TomTom, a Dutch producer of navigation devices, has already expressed their wish to shift their effort from developing devices to developing software, in order to provide autonomous car makers with the required navigation systems (NOS, 2017).

Finally, government organisations could benefit from the knowledge as many countries have obligated themselves to reduce carbon dioxide emissions by signing the Paris Agreement, in the hopes of reducing climate change. As less fuel consumption leads to lower emissions of CO_2, governments might want to stimulate fuel-efficient driving. In fact, Dutch courts have ruled that their government is legally obligated to reduce such emissions in the short term and that no measures may be taken that increase this. This might mean that several highways might not be upgraded to higher maximum speeds, and that some might actually have to adopt lower speeds. For both cases, it can be helpful to see the effect on both fuel and time consumption of drivers. The current research could offer more insights into the practical effects of this (Financieel Dagblad, 2017).

In summary, both practical applications and possibilities for studies into future applications exist, creating a clear relevance and applicability for researchers, companies and individual users for the results of this study.

2) Research Objectives

In order to tackle the problems identified in the previous chapter and to come up with an attempted solution, the current research should have clear objectives. These have been split in three parts: a **mission statement**, **research question and sub-questions** and a **scope discussion**. The following sections focus on these three respective elements.

2.1) Mission Statement

The mission statement of this research is as follows:

> *"To create a **website** that allows for **calculating and comparing** the **car routes** for different **custom maximum speeds**."*

Some of the words in this mission statement have been emphasised, as they are crucial choices that impacted the scope that this research adheres to. As such, these choices are explained here. Any other things that are explicitly not part of the scope are discussed in Chapter 2.3.

- **Website**: The instrument chosen for the development of this project is a website, i.e. an internet languages-based system. Two main reasons exist for this choice, as opposed to a stand-alone application. As was mentioned in Chapter 1, a major drive for performing this research is to increase awareness as to the different possibilities that exist in driving options. If awareness is to be raised in as many people as possible, the system should be easily accessible to as many people as possible too. Applications are usually platform-dependent, only working on Windows, Android, Linux or whichever operating system that they have been designed for. Websites, however, are strongly platform-independent, making them accessible to anyone with an internet browser – even though different browsers can interpret the same website slightly different. In addition, it must be kept in mind that this research' central challenge is a GIS one, and not a computer science one. As the author has more experience with website than application development, this left more time to focus on the GIS side, rather than on the technical one. The website and its components are explained in more detail in Chapter 6.3.
- **Calculating and comparing**: In order to get to know the influence of custom maximum speeds on time and fuel consumption, paths with such different speeds had of course to be calculated. However, it was crucial that these different outcomes were compared too. The system is explicitly not to recommend any alternative itself, but to provide a factual comparison of the alternatives, by showing the time and fuel needed for each of them. This allows users to make their own choices, based on their personal preferences and needs.
- **Car routes**: This research does explicitly focus on cars, as opposed to pedestrian, rail or any other non-motorised, non-road modes of transport. The conclusions of this research might be applicable to other motorised road vehicles such as buses, motorcycles or lorries too. Whether this is true or not, has not been investigated here though, but could be the subject of further studies. Possible differences between types of cars, such as SUVs and compacts, are looked at though. In summary, any vehicle that may be driven with a European driving license 'B' was eligible for inclusion: cars with a maximum authorised mass not exceeding 3.500kg with room for not more than eight passengers, in addition to the driver (European Parliament, 2006).
- **Custom maximum speeds**: As opposed to always driving at the speeds allowed on certain road segments – when traffic allows for this – lower speeds are used to analyse their effects on time and fuel consumption. As this research is limited to roads in the Netherlands, where no roads with higher speed limits than 130km/h exist, this is the highest speed included.

2.2) Research Questions

Now that the objective of this project has been explained in the previous chapter, questions that need to be answered in order to successfully achieve this objective, are presented. Using the system that was to be built, the following main question can be answered:

> *"To what extent does driving at different maximum speeds influence the time and fuel consumption for fastest car routes?"*

This question needed to be answered step-by-step though, which is why additional questions, i.e. sub-questions, have been introduced. Each of these questions represents roughly a step or a collection of steps in this research, which consists of three main phases: a 'Theoretical Research' part, a 'System Construction' and a 'Practical Research' part. Whereas the first phase is a literature study serving as preparatory work, the third one consists of using the system to analyse the possible influence of custom maximum speeds on time and fuel consumption. The second stage focuses on building the system, which has no sub-questions tied to it.

2.2.1) Theoretical Research

1) *"Which car characteristics influence fuel consumption and how?"*: To check whether or not differences in max speeds' influence on fuel consumption hold for a wide range of car types, cars with different characteristics are looked at. In order to be able to have an as diverse as possible group of models, though, it had to be known which characteristics, such as – possibly – weight, are relevant for fuel consumption. The answer to this sub-question is a list of characteristics with, whenever possible, mathematical functions detailing their relations to fuel consumption.
2) *"For a chosen group of car models, how does driving at different speeds impact fuel consumption?"*: After having found an answer to the first question, it became possible to create several fictional car models, differing for the characteristics which were found to have an influence on fuel consumption. The goal here was to have a group with as many extreme cases as possible, to be able to test whether or not significant differences could exist between different cars. The answer to this question consists of a list of models along with a mathematical function detailing the relation between driving at certain speeds and fuel consumption.
3) *"Which requirements exist for data and methods for building the desired navigation system?"*: In the Methodology section of this document, Chapter 3, some expected data requirements had already been mentioned. These requirements had however to be researched in more detail. In addition, a suitable method for data manipulation – editing the data to make it suitable for the current purpose – was to be found. The same holds for a routing algorithm. Furthermore, the first two sub-questions could reveal any other requirements which had not been thought of in earlier stages. Also, factors found to influence fuel consumption in these questions are looked at, to see whether or not it is technically possible to implement them. Finally, a blueprint for the construction of the website was drawn up. After answering this sub-question, the system could be built.

2.2.2) Practical Research

4) *"How much fuel and time can be saved or lost by driving at certain maximum speeds in practice?"*: After having built the desired system, it became possible to analyse the actual influence of custom maximum speeds on fuel and time consumption. Here, a Monte Carlo simulation is used with an array of possible paths.
5) *"To what extent do these time and fuel differences for different speeds hold when traffic, dynamic maximum speeds and car models are introduced?"*: Whereas traffic, dynamic maximum speeds and car models were not taken into account for the previous question, they are here in order to see whether or not the possible influences of maximum speeds would still hold when introducing these factors. The results of this analysis are compared to those of the scenario in the previous sub-question, in order to check for any differences that might exist.

2.3) Further Scope Limitations

The two previous sections have specified what is done in the scope of this research project. In order to preserve its focus, though, it is also beneficial to explicitly mention what is not part of the scope.

It is important to remember that this is a proof-of-concept, i.e. the project wants to analyse whether or not custom maximum speeds would be a useful addition to navigation systems. As such, even though the goal is to build a working system, it is in no way implied that it was built in the most (technically) efficient way: there may and probably will be better ways to build this.

In addition, the navigation is only ex ante: no real-time routing – i.e. a system that follows the user while driving and updates directions when needed – was implemented. Factors such as traffic and dynamic speeds limits have been based upon the moment in time at which the paths were calculated, i.e. they are static. As with the previous limitation as to efficiency, it would not add to the proof-of-concept to make these more dynamic.

3) Methodology

This chapter will focus on the 'how' of the current research. It acts as a framework, or as guidelines, for the study itself. A set of steps is to be followed, which should help to answer the sub-questions and main question as stated in Chapter 2. These steps are described in Chapter 3.1. The data which is known to be needed, is described here in Chapter 3.2. In addition, software used is described in Chapter 3.3.

3.1) Research Steps

This section will focus on the different steps that this research went through, in order to achieve its objectives. Some of these steps directly lead to an answer to sub-questions: when this is the case, it is mentioned. As part of this study, a proof-of-concept navigation system was designed, built and used to analyse custom max speeds' influence on time and fuel consumption. These three actions are the main phases of this research. The first stage, 'Theoretical Research', focuses on theory behind fuel consumption and the requirements for building such a system. The second one, 'System Construction', as the name suggests, leads to the actual building of the system. Finally, during the third stage, 'Practical Research', that system was used to analyse the possible influence of custom max speeds.

These steps have been performed according to the Waterfall model. This means that as soon as one step has been concluded, the next step started, all in a sequential way. As mentioned by Stoica et al. (2013), this method is best used when dealing with a small team and when there is no customer who can influence and change the project during its execution. As both are true for this study – only one person works on it without a customer – this method is applicable to the current research.

3.1.1) Theoretical Research

This stage of the research explores the theory needed on fuel consumption and the requirements for building the navigation system as envisioned.

1) **Find information on car characteristics' influence on fuel consumption**: The very first step involves a literature study to directly answer the first sub-question. This establishes a list of technical characteristics of cars that influence their fuel efficiencies. Based on the sizes of these influences, characteristics are chosen to be used in the next steps.
2) **Create a group of cars differing on the characteristics identified in the first step**: Different cars will have different fuel behaviours, and this study aims to analyse whether any possible influence of custom maximum speeds holds for different car types. As such, a small group of five fictional cars is created, based on the characteristics found in the first step. These cars should differ as much as possible.
3) **For every car, create a graph showing the fuel consumption at different speeds**: This step answers the second sub-question. First, a literature study into the relation between speed and fuel efficiency is carried out. Secondly, this is to be related to the cars as created in the previous step. The outcome of this should be a graph showing fuel consumption when driving at different speeds under perfect circumstances, i.e. while being able to keep driving constantly at these speeds. An example of how these graphs should look, can be seen in Figure 3.1.

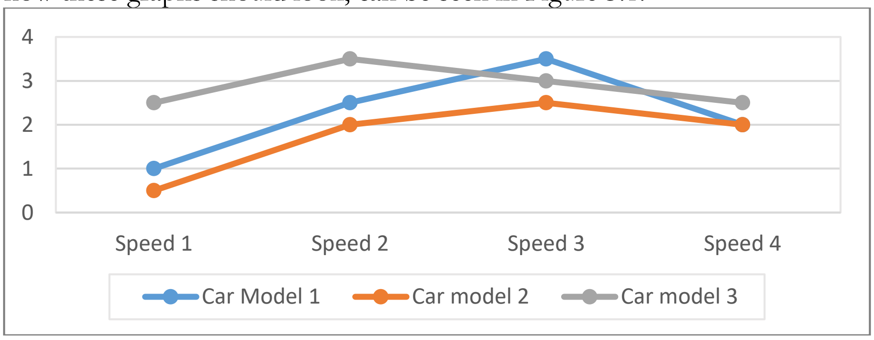


Figure 3.1: Example graph for the speed/fuel efficiency relation for different cars

4) **Find requirements for the development of the system and create a blueprint**: This step answers the third sub-question. With the information about fuel consumption in theory available, the requirements as to data and methods can be listed. All of this can then be translated into a design

blueprint for the navigation system itself. This blueprint shows how certain elements, such as data, interact to form the system. An example of how such a blueprint should look like is shown in Figure 3.2.

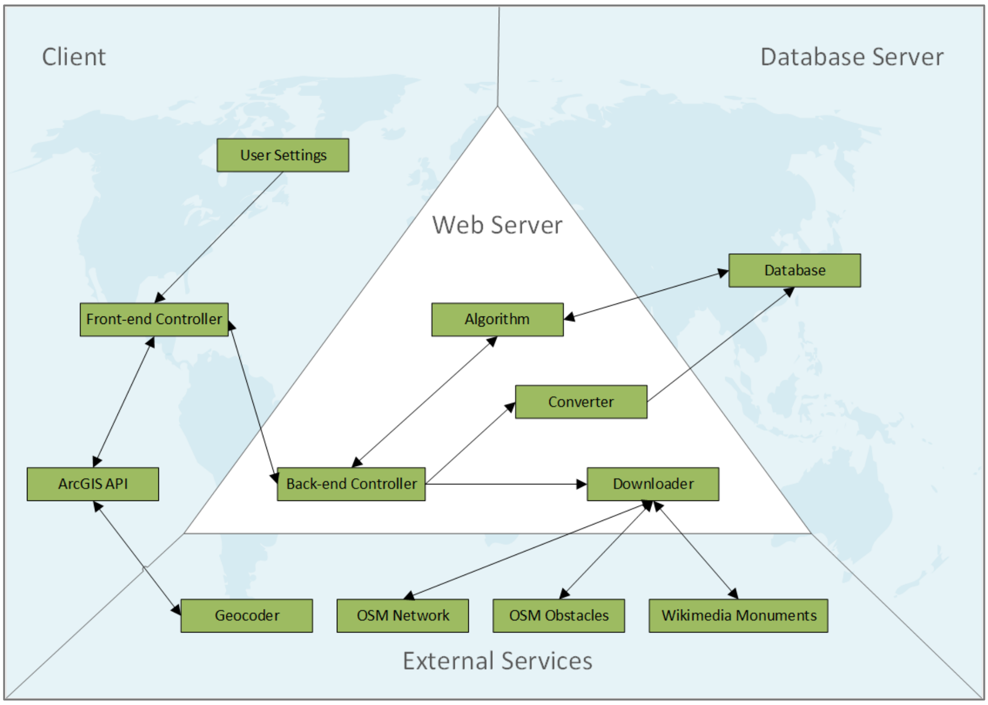


Figure 3.2: An example blueprint for the design of the navigation system (De Jesus Madureira Porfírio et al., 2017)

3.1.2) System Construction

This stage includes the actual building of the navigation system, based on the information acquired and design created during the first research stage. Though none of these steps answer any sub-questions directly, they are reported on as with all other steps of this research: the output of this stage is technical documentation which should allow other parties to build a replica of this system, should they wish to do so.

5) **Create a database with a road network and maximum speeds**: The building starts by setting up a database using PostgreSQL (see Chapter 3.3.1 for details) and populating it with road segments in such a way that they can be used to calculate the paths needed, i.e. the fastest path when adhering to a certain maximum speed. The data used for this is the 'Nationaal Wegen Bestand', a dataset with all of the roads in the Netherlands. Of this dataset, the part involving the province of Zuid-Holland is used. The set is described in more detail in Chapter 3.2.1. Using ArcMap, it is joined with another dataset specifying the legal speed limits for these roads (see Chapter 3.2.2).
6) **Add attribute columns to the road network with altered maximum speeds**: Since this research aims to look at the influence of different maximum speeds, a way to compare paths with these different speeds should exist. Since the network used is always the same, instead of limiting the actual speed of the driver, the maximum speeds in the network can be altered. For this to work, additional columns are added to the network database with different maximum speed limits in steps of 10km/h. Since no road in the Netherlands and thus Zuid-Holland, the study area chosen, allows for driving faster than 130km/h, the column which includes 130km/h is automatically identical to the original network. For the 120km/h network, roads allowing for 130km/h will be lowered to 120, and so on. What needs to be taken into account though, is that minimum speeds do also exist: it would not be secure to drive 30km/h on a 130km/h road, for example. Such minimal safe speeds need to be studied here in order to know which roads to exclude altogether for certain maximum speeds.
7) **Build the navigation system website based on the design blueprint**: In the final step of this stage, the actual website hosting the navigation system is built. This is done based on the blueprint as designed in step 4. The website itself is hosted locally using XAMPP, a software package explained in more detail in Chapter 3.3.3. It connects to the database as hosted on PostgreSQL, as mentioned before. This DBMS uses the PostGIS plugin in order for it to work with geometrical data too.

This website needs to be able to, for a chosen car model (as designed in step 2), calculate the fastest paths between a user-specified start and end point. This is to be done for all of the custom maximum speeds. The output should be a graph showing both the fuel and time consumption for each speed, in order to allow for the user to compare the influence of these speeds. An example of such a graph is given in Figure 3.3. These results should also be stored for later usage during the analyses of the practical research.

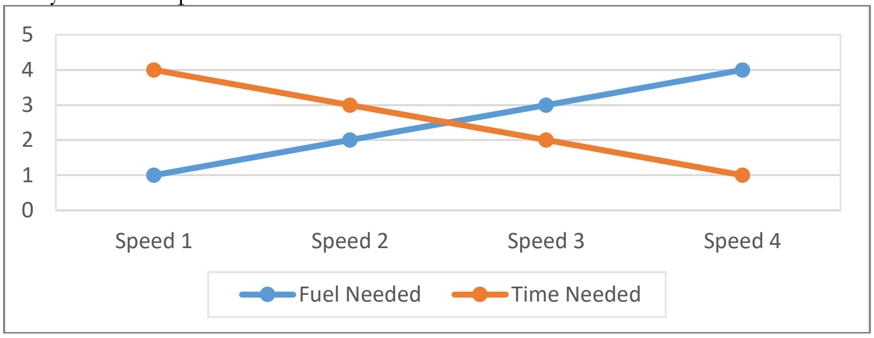


Figure 3.3: Example graph for fuel and time consumption for different maximum speeds

In addition, a map showing the paths for the different maximum speeds is shown, a sketch of which can be seen in Figure 3.4.

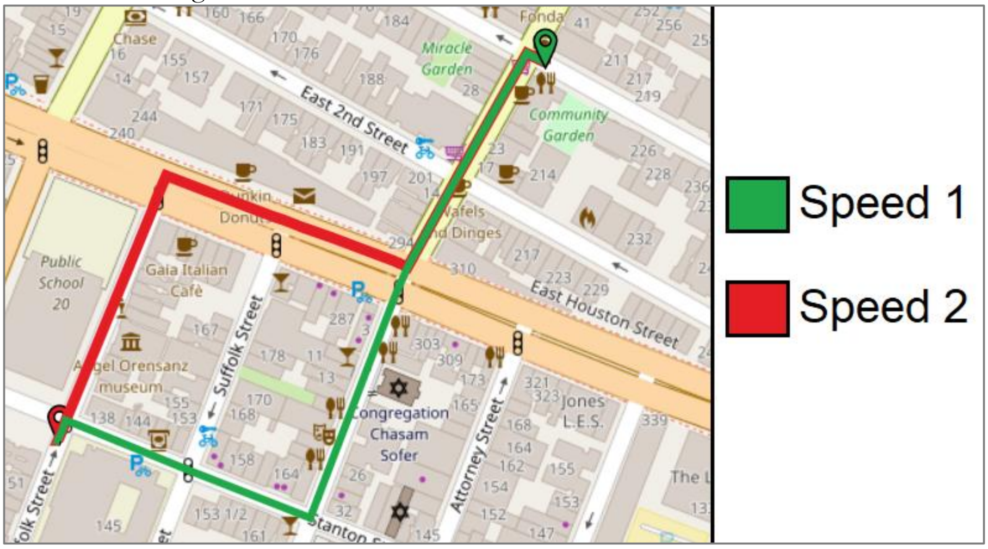


Figure 3.4: Example map for the fastest paths at different maximum speeds

3.1.3) Practical Research

The last stage is where the actual analysis takes place. Using the system built in the previous stage, calculations are performed that should provide the numbers to answer the remaining sub-questions and main question.

8) **Perform calculations to check for the fuel and time consumption significance**: This step provides the answer to the fourth and fifth sub-questions. For a large number of randomly chosen start-end point combinations (more on this near the end of this step description), the fuel and time consumption for the different maximum speeds is calculated using a Monte Carlo simulation. This method involves using random input values to simulate a large range of possible outcomes. In doing so, three steps exist. The first one is the *"Sampling of random variables"*, which involves creating a sample of random values for the input variable used – in this case the pairs of random start and end points. In the second step, *"Numerical experimentation"*, these samples are used as input for the actual calculations, which yield samples of outputs: the time and fuel consumption for the route between the two input points. Finally, the third step, *"Statistic analysis on model output"* involves a statistical analysis of characteristics of the outputs. These parameters are compared for the different maximum speeds, to check for significant differences (Missouri University of Science and Technology, 2017).

This comparison is performed using three main statistical tests. The interest here lies in discovering the size of the effect of different variables upon time and fuel consumption. In order to validly do this, however, it always needs to be checked first whether a significant difference of means exist at all, e.g. between the situations with and without traffic. In order to test this, ideally, an ANOVA test would be used, as it has the most power. There are, however, different requirements that need to be fulfilled in order for an ANOVA to be possible. First of all, the variables compared need to be ratio ones. Since time and fuel consumption are both ratio variables, this condition is always true. Next, the samples compared need to be independent. This holds too, as the results of e.g. car model #1 do not in any way influence the results of the other car models. These results also need to be normally distributed. If the samples have identical sizes – which is true here – this condition holds whenever the size of all samples combined equals or is greater than 30. With a sample size of tens of thousands, this is most certainly met. Finally, population variances need to be similar. This is to be tested with a Levene's test, unless sample sizes are roughly equal. Since they are always equal, there is no need for an additional test, and this condition is met too. As such, in every single case, the requirements for running an ANOVA are met, and this test can be used (De Vocht, 2013a).

After performing the ANOVA, the effect size could be measured using different tests, all with their advantages and disadvantages. A test with substantial power is the Pearson correlation, which is able to determine effect sizes of correlations in which ratio variables are involved. The advantage of Pearson, is that it not only gives the effect's size, but its direction too. The indicator, also called 'r', varies between -1 and +1, the first one indicating a negative relation (the more, the less) and the latter a positive one (the more, the more). Finally, values approximating 0 indicate no relation. The squared 'r', 'r^2', indicates the amount of variance that can be explained by the change in factors. Though powerful, relations must be linear in order for Pearson to correctly estimate the effect size. As can be seen in the graph figures of Chapter 10, this is not the case though: as such, alternative tests are to be used (De Vocht, 2013b).

Instead, the ANOVA Eta test is to be used. This test produces results similar to the Pearson correlation, the difference being that numbers are always positive: there is no direction of effects, as this is not possible with categorical variables, which cannot be negative themselves. Other than that, results can be interpreted similarly, with 'Eta' indicating the effect size and 'Eta Squared' the amount of variance explained by the changes in the independent variables. This test is used to estimate the effect size of any variable which, according to ANOVA, is deemed to have a significant influence upon either time or fuel (De Vocht, 2013a).

In addition to maximum speeds, other variables are used and varied here: traffic, dynamic maximum speeds and car models. It is checked whether or not any significance found still holds when varying these factors, or if any significance that did not exist emerges when doing so.

The exact number of simulations performed in the end, was eventually determined largely by the calculation speed: the faster calculations are done, the more simulations can be run. In every simulation, the combination of start and end points constitutes the random variable that is different for every iteration. As mentioned already, the exact amount of iterations in practice depended upon calculation speed. However, EpiX Analytics (2017) believes that 300 is the rough minimum when it comes to these iterations, and that more is always better. Driels & Shin (2004) in turn determined the optimal number of iterations to be between 500 and 1000. As such, a number of around 500 had sufficed in the current study. In the end, however, it was possible to raise this number to 750, which is the number of iterations actually performed. An important caveat here, however, is that every iteration does in fact consist of multiple calculations for the variations of the factors mentioned earlier. This significantly increases the actual amount of path calculations performed. In every iteration, 180 paths have to be calculated: all options for traffic (yes/no) -> 2 * dynamic maximum speeds (yes/no) -> 2 * different car models -> 5 * different maximum speeds -> 9 = 180. In effect, this means that 180 * 750 = 135.000 paths are calculated.

After doing so, SPSS (see Chapter 3.3.4) is used to statistically analyse whether or not significant differences exist between the consumptions for the different speeds.

9) **Conclude if (and if so, how) driving at certain speeds will affect fuel and time**: After having answered all of the sub-questions, the main question can be answered with the knowledge acquired up to this point. A conclusion is given on how, if at all, custom maximum speeds are able to influence

time and fuel consumption, how the factors of step 8 affect this and what that means in practice for drivers. In addition, these conclusions are discussed.

3.2) Required Data

The data used in the final application is presented and discussed here, with a section on every dataset.

3.2.1) Zuid-Holland Road Network

Finding the shortest, fastest, most efficient or whatever kind of route between two points when restricted to roads, railways or any other network is part of network analysis. As the name suggests, a network is required for such analyses to be carried out. De Smith et al. (2015, p. 537) define a network as follows:

> *"A collection of vertices and edges together with associated attribute data that may be represented and analyzed using graph theoretic methods. A network is often defined as a graph that has at least one real-valued attribute or weight (e.g. length) associated with every edge."*

As such, three elements here are essential for the network needed: it needs to be made up of vertices and edges, graph theoretic methods must be applicable to it and some weight needs to be associated to every edge. This is the cost of traversing the edge, which may be expressed in length or time, depending on what makes more sense in the respective context. The graph theoretic methods, i.e. methods used in the analysis of the network, are discussed later in the requirements exploration of Chapter 6. The network itself is discussed here though.

In this case, the required network represents the road network of Zuid-Holland, the most populous province of the Netherlands. Though this study did aim at including every road in the country, high calculation times forced this goal to be scaled down to the province level. This network is what cars in that country would use to get from A to B. Rijkswaterstaat, a Dutch governmental organisation responsible for highways and waterways, publishes a complete dataset of Dutch roads monthly: the 'Nationaal Wegenbestand', or NWB. The fact that this dataset is published at this frequent rate, is due to the open data laws by the Dutch Government, which aims to provide public and private organisations with as much updated data as possible for them to use and reuse as they see fit (Overheid.nl, 2016).

The data is maintained by different organisations: whereas highways are maintained by Rijkswaterstaat, information on local roads is the responsibility of provinces and municipalities. As changes are included automatically in the NWB, it constitutes the most up-to-date dataset of roads in the Netherlands. It is set up in such a way that extra data can be easily attached. Rijkswaterstaat itself calls this the 'integrator' function of the NWB: as long as external datasets use attributes such as the address, identification number or even coordinates of a road, they can be integrated with the network itself. This is crucial for the addition of speed limits, as discussed in the next section (Rijkswaterstaat, 2014).

As opposed to what its name (in translation: 'National Road File') suggests, the NWB does not only offer data on roads: waterways and part of the railway network are included too. In order to avoid working with more data than necessary, pre-editing of the dataset to include only relevant data is done in step five of the current research.

The NWB is available for download at Overheid.nl (2017a). The version used here is the one published on the 1st of October 2017.

3.2.2) Max Speeds per Road Segment

As mentioned in the previous section, one of the aims of the NWB is being a dataset that can be easily extended with additional data. As the focus of this study is on variable maximum speeds, speed limits of roads need to be known. By default, however, such limits are not included in the NWB. As such, they have to be added to the dataset. As with the original dataset, Rijkswaterstaat offers the speeds dataset at Overheid.nl (2017b). It consists of a .csv set which, using the unique identifier for roads ('WVK_ID'), can be merged with the NWB. This is done in the fifth step of this research, using ArcMap (see Chapter 3.3.5). Dynamic maximum speeds, i.e. maximum speeds that change according to the time of day, need to be included as well: more on this in Chapter 3.2.4.

3.2.3) Traffic

The last part of this study, the practical research, includes a comparison between situations with and without traffic, as encountering traffic does have an influence on how fast drivers can go.

As such, it was originally envisioned to include real-time traffic in the application. This would mean, however, that for every path calculation (or group of calculations) this traffic data would have to be downloaded and joined with the road segments. While technically possibly, it would have impacted the calculation time by adding time for downloading and data manipulation. In addition, datasets available, such as the one by the Nationale Databank Wegverkeersgegevens (2017), only have data available on a limited number of roads, i.e. larger ones. As such, using such a dataset would still lead to a situation in which valuable time is used to import an incomplete set.

Therefore, it was decided to replace this real(-time) dataset on traffic by a custom, fictional one. Here, every road, including smaller ones, has a random traffic intensity value attributed to it. This would of course not be a valid way to include traffic in a final, public application: in that case, real-time traffic, with its downsides, should be used. Using a randomised situation does suit the purpose of this study, though, which is to compare situations with and without traffic. Whether these situations are real or fictional, is irrelevant for the purpose of this study.

3.2.4) Dynamic Maximum Speeds

As is the case with traffic, dynamic maximum speeds need to be introduced to compare situations in which these are enabled and in which these are disabled. Currently, the dataset used for speeds limits (see Chapter 3.2.2) does not include any data on dynamic limits. In addition, no readily available datasets exist with this data.

As such, the only option to include this is to enter such information manually using an available .pdf map on variable speeds by Rijkswaterstaat (2017), part of which is shown in Figure 3.5 as an example. While adding such information to the road dataset manually is far from ideal, it is possible to do so due to the relatively small number of roads that use variable speed limits.

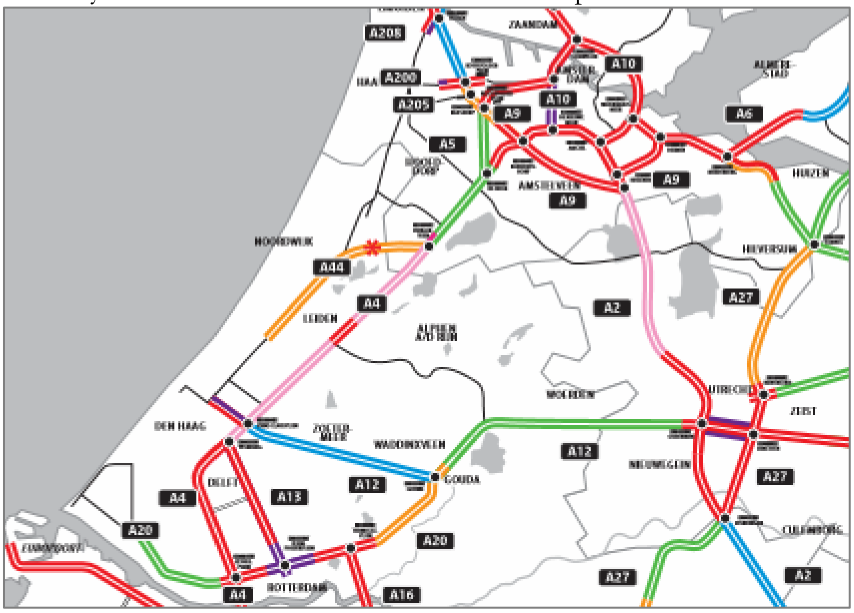


Figure 3.5: Speed limits in the western Netherlands – pink and blue roads indicate dynamic speeds (Rijkswaterstaat, 2017)

3.3) Software Used

This section provides descriptions on the software used and mentioned throughout this research.

3.3.1) PostgreSQL

In order for the website to be able to access the road network, this network needs to be stored in a database. For this to be possible, a database management system (DBMS) is needed. As described by Kedar (2009, pp. 1-4 – 1-5), DBMSs *"consist of a collection of interrelated data and a set of programs to access that data"* and *"are designed to manage large bodies of information"*. As the network dataset (see Chapter 3.2.1) is organised as a database and, with its 0.5GB, can be classified as a 'large body of information', a DBMS is the appropriate system to work with this data.

The exact system chosen here is PostgreSQL. It has all of the functions needed in a DBMS, and has three main advantages that led to the choice in favour of its usage in this study. First of all, it is open source software, meaning that anyone can use it for any goal for free. In addition, a wide range of plugins providing additional features are available for this system: one of those plugins, PostGIS, is used and discussed in the next section. Finally, another strong point of PostgreSQL is its documentation: whereas decent documentation is all but guaranteed with a large share of software packages, PostgreSQL comes with an extensive description of all of its functions, making its use a lot easier. All of this has led to this DBMS being chosen for use here (PostgreSQL, 2017).

3.3.2) PostGIS

PostGIS is an extension to the PostgreSQL DBMS (see previous section). This DBMS does not normally offer support for geometrical data types and methods. PostGIS, however, adds this support. It allows for the network dataset to be loaded as actual geometrical data: without it, this kind of data would just be handled as 'normal' data, i.e. without a spatial component. This allows for methods that would otherwise be impossible or unwieldy to do, such as calculating the distance between two objects. As with the base software itself, the plugin is open source. More importantly, an extensive documentation is provided here too with details on every single feature (PostGIS, 2017).

3.3.3) XAMPP

XAMPP is the software package used to host the website used in this study. The emphasis here is on 'package', as XAMPP itself is actually just an overhead for existing software. It removes the need for installing these different components separately and getting to make them work together. Included are Apache, PHP, MariaDB and Perl, among others. While the latter two are not used here, the first two are essential to the website's functioning. Apache is the actual web server that hosts website files and makes them available for access via the internet.

By default, such files are executed client-side, i.e. in the browser of the user visiting the site. As these are executed by the client, users are, technically, able to read all contents of these files. While this does not pose a problem in many cases, for security reasons so-called server-side languages are also needed. This is especially true when using a database, as the connection to this database needs to be secure. As such, the address, username and password cannot be known by the client. By having the server execute this part, these issues are solved. For this to be possible though, an additional PHP (one of the most common server-side languages) component needs to be activated. XAMPP handles all of this.

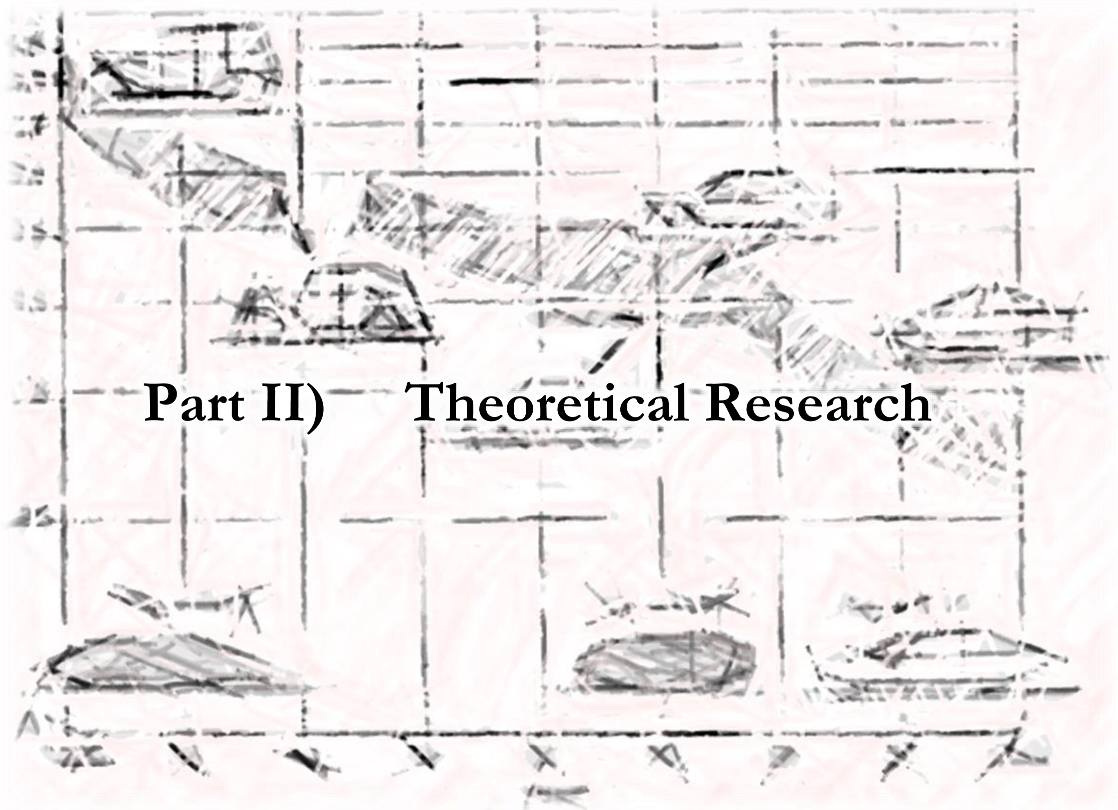
In addition, as with PostgreSQL and PostGIS, XAMPP features an extensive documentation and support community (Dvorski, 2007; Apache, 2017; PHP Documentation Group, 2017).

3.3.4) SPSS

SPSS is a piece of software by IBM offering modules for various statistical analyses. Its main functions are reading, editing and visualising data, but many additional features are available to analyse such data, which is its main advantage. According to a study by Rexer Analytics (2015), it is the second-most used analytics tool. For this study, it is used to analyse the fuel and time consumptions for routes when using different maximum speeds. Whether or not these consumptions differ significantly or not between cases with different maximum speeds, car models, traffic and dynamic speeds, is what SPSS should help to analyse and answer (IBM, 2017).

3.3.5) ArcMap

The final piece of software used is ArcMap, one of the many programs available within the ArcGIS platform. Developed by Esri, it offers a complete suite of GIS functionality. According to Wieczorek & Delmerico (2009), a GIS can be called such if it allows for the collection (and editing), analysis and reporting (or visualisation) of spatial data. ArcMap fulfils all of these requirements, making it a suitable program for GIS functions. In the current research, it is used to prepare spatial data for introduction into the database, whenever necessary. An example of this is the addition of speed limit data to the NWB in step five of this research. The analysis of the spatial data and its reporting is handled by the website, however (Esri, 2017a).



Part II) Theoretical Research

4) Car Characteristics Influencing Fuel Consumption

The first chapter of the theoretical research phase involves a literature study that should answer the question: *"Which car characteristics influence fuel consumption and how?"* As mentioned in Chapter 2.2.1, this answer should consist of a list of characteristics with, whenever possible, mathematical functions describing their relations with fuel consumption.

Luckily for this study, fuel consumption of cars has been studied extensively already, including mathematical relationships. Already in the mid-1990s, Ross (1994) came up with an elaborate model that allowed for the computation of the power that engines should generate in order to allow for driving at certain speeds, under certain circumstances. The final formula that he came up with is as follows:

$$Pb = \frac{Ptires + Pair + Pinertia + Pgrade}{\varepsilon} + Pacc$$

In this equation, every 'P' is a power requirement, and ε is the overall efficiency of the car in generating power, i.e. the engine and transmission efficiency. All of these variables can be equated to functions in their own right, as Ross does too. These are as follows:

$Ptires = CR * M * g * v$ *(power lost to friction between the tires and ground)*

$Pair = \frac{½ * p * CD * A * v3}{1000}$ *(power lost to air drag)*

$Pinertia = ½ * [\frac{\Delta v2}{\Delta t}]$ *(power used while accelerating)*

$Pgrade = M * g * v * sin\theta$ *(power used – or won – by climbing or descending)*

Whereas 'P_{acc}' is a sum of power requirement by non-essential accessories such as air-conditioning.

In 2012, Wu & Liu translated this model into the figure of Figure 4.1, making it easier to understand which and how certain factors impact a car's power requirement.

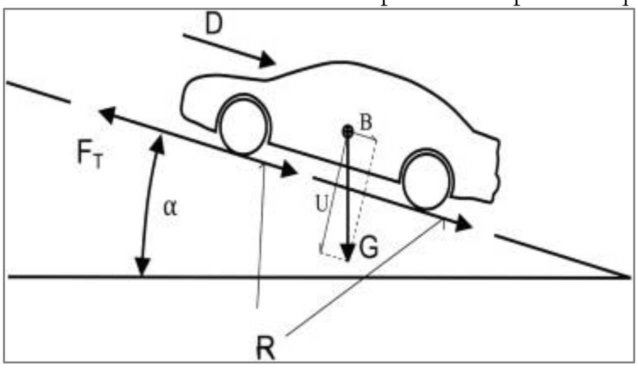


Figure 4.1: Factors impacting cars' power requirements (Wu & Liu, 2012)

Here, 'D' is equal to Ross' 'P_{air}', 'R' to 'P_{tires}', 'F' to '$P_{inertia}$' and 'α' to 'P_{grade}'.

In order to get the actual fuel consumption per unit of distance driven, though, the following formula – also by Ross (1994) – is to be used:

$$MPG = \frac{120600}{\frac{3600 * k * V * < N >}{vav} + \frac{1}{nt} * \frac{1609 * Cr * M * g}{\varepsilon} + \frac{3.6 * p * Cd * A}{\varepsilon} * \frac{< v^3 >}{vav} + \frac{M * g * v * sin\theta}{\varepsilon} + \frac{3600 * Pacc}{vav}}$$

The functions in Ross' model contain three different kinds of variables: constants, circumstantial variables and car characteristics. The constants, 'g' and 'p', will be the same at any given moment. 'g' is the gravitational power pulling the car down, which, according to Hirt et al. (2013), amounts to roughly 9.8 (m/s^2) for any point on earth. The same is true for 'p': this is the air density which is also strongly similar for any point on earth at about 1.2 (kg/m^3). Though this number does vary slightly under different temperatures (1.1 at 50°C and 1.3 at 0°C), this variable will be kept at 1.2 for the sake of simplicity (Shelquist, 2016).

The circumstantial variables change depending on where and how a car is driving. The first one which is relevant here, is 'sinθ', or the grade of the road. As one would expect instinctively, steeper grades lead to a higher power requirement and thus a higher fuel consumption. Finally, 'v', or the speed at which a car is currently travelling, impacts the power usage too: the faster one travels, the stronger effects such as air drag and tire friction are.

Finally, car characteristics affect all of this too, the exploration of which is the goal of this chapter. 'k', 'V' (not to be confused with the aforementioned 'v'), '<N>', 'n_t', 'C_R', 'M', 'ε', 'C_D', 'A' and 'P_{acc}' are all characteristics that impact how much fuel a car uses while driving and even while idling. Of these, only some will be varied during the course of this study, as too many variables would simply make the calculations too complex and reduce the model's ability to simplify. 'k' – engine friction at zero power – will be kept constant at 0.225 (a coefficient), as will '<N>' – engine speed – at 31 (revolutions per second), 'ε' – transmission efficiency – at 0.87 (a coefficient) and 'P_{acc}' – power use of non-essential accessories – at 0.5. All of these values are deemed normal by the different authors previously mentioned in this chapter.

This leaves 'V', 'n_t', 'C_R', 'M', 'C_D' and 'A'. These different characteristics are discussed in more detail in the next subsections.

4.1) Mass ('M')

With mass, the total weight of a car in kg, including its load, is meant. This mass firstly has influence on fuel consumption via 'P_{tires}'. The heavier a car is, the more weight will be exercised on its tires. As a result, the friction between tires and the ground becomes higher, leading to a higher power requirement to overcome this and move the car. The same is true for 'P_{grade}': when climbing a hill, heavier vehicles will need more power to get to the top than lighter ones. The opposite is true too: when descending, heavier cars will go down even easier than lighter ones. As can be seen in the previous section, the relation between mass and these requirements is linear: a car twice as heavy will require twice as much power (Ross, 1994; Biggs & Akçelik, 1987).

4.2) Frontal Area ('A')

Even without any wind, air will be hitting an object when this object moves. As any other substance would do, air resists anything trying to get through it: this is the so-called air drag of 'P_{air}'. In general, bigger objects will have to resist more air than smaller ones when moving.

In fact, this depends – in addition to aerodynamics, which will be discussed in the next section – on cars' frontal areas. This frontal area is where air would hit the vehicle directly, or frontally, when driving forward. It is basically the area that one could see when standing right in front of the car, as is shown in Figure 4.2:

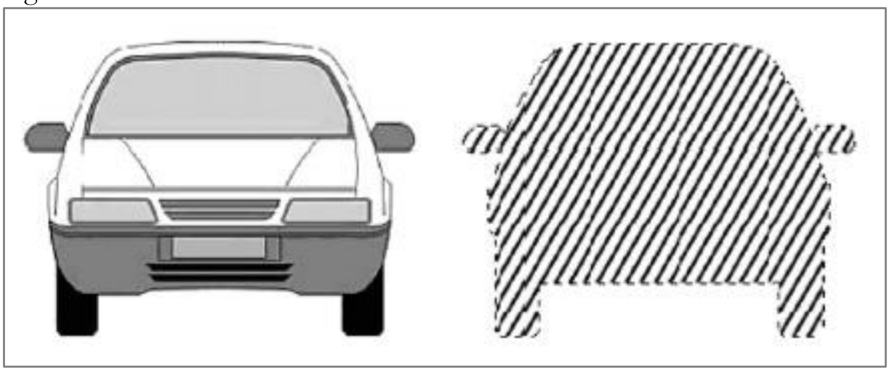


Figure 4.2: The frontal area of a car (left) and its effective size (right) (Wu & Liu, 2012)

Whereas in this figure the left picture shows what you would actually see, the right picture shows the area that counts here. The size of this second area is what is used to calculate the power to combat the total air drag. The relation between the area and drag is linear: cars with a frontal area twice as big, will suffer twice as much air drag (Wu & Liu, 2012).

4.3) Coefficient of Aerodynamic Drag ('C_D')

As discussed in the previous section, air drag is the resistance of air towards a moving object. The frontal area is not the only characteristic impacting this, though: the shape of a car is essential. While considerations such as aesthetics are important in designing a car's shape, its aerodynamic capabilities cannot be underestimated either. Whereas in the beginning of the 20th century cars were strongly box-shaped, this has shifted towards more pointy, aerodynamic designs, as is shown well in Figure 4.3.

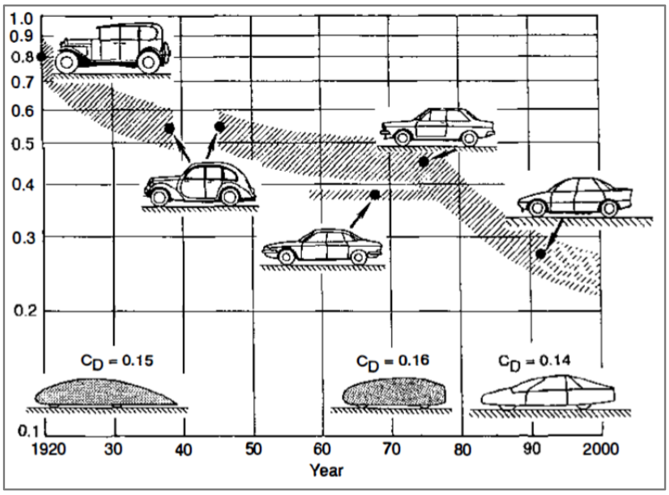


Figure 4.3: The coefficients of aerodynamic drag of different cars throughout time (Hucho & Sovran, 1993)

As can be seen in this figure, the coefficient of aerodynamic drag varies between 0 and 1. In the first, entirely hypothetical, case, the shape would allow air not to hit the object at all. As Hucho & Sovran (1993) point out, though, it would be practically impossible to design a vehicle with a lower coefficient than 0.05. The opposite of this would be a totally non-aerodynamic car with a coefficient of 1, basically meaning a box on wheels. What exactly makes vehicles aerodynamic or not is not of high importance to this study though: it is important to know, though, that 'normal', modern-day cars have coefficients varying between about 0.5 and 0.2.

As can be deducted from the formulae in the first part of this chapter, the relation between the coefficient of aerodynamic drag and the air drag is a linear one.

4.4) Coefficient of Rolling Resistance ('C_R')

In the late '70s, Clark & Dodge (1979, p. 3) defined rolling resistance as the rolling loss caused mainly by *"the friction or scrubbing between tire and roadway"*, the other factor being the state of the tire. This friction depends on the materials of which both the tires and roadway are made of. The higher this coefficient, the more friction – and the more friction, the more power required. A dirt road, for example, is more energy-intensive to traverse than an asphalt road, due to the higher friction between wheels and that road. On the other hand, pneumatic tires (rubber with air) will do better than wooden ones.

Though both tires and roadway are thus important, only the tires are taken into account here when looking at this coefficient. The reasons for this decision lie in the reality of the case study area: in the Netherlands, unpaved public roads barely exist. As such, no meaningful differences would exist here. In addition, no complete dataset exists detailing the materials of roads: while OpenStreetMap (2017a) does offer some information on this, it does by far not offer this data for all roads in Zuid-Holland.

As such, only tires matter when using this coefficient. It can be best seen as the 'quality' of the tires, including its original quality (expensive/cheap) and current condition (pristine/worn-out). As with the coefficient of aerodynamic drag, its possible values can, theoretically, vary between 0 and 1. In practice, however, this coefficient is somewhere between 0.010 and 0.015 for stock tires on asphalt roads (Gillespie, 1992, p. 117).

Again, the relation between this coefficient and 'P_{tires}' is a linear one.

4.5) Thermal Efficiency ('n_t')

Ross (1994) mentions several efficiencies when discussing his model for fuel consumption. Transmission efficiency, mechanical efficiency and thermal efficiency are all mentioned. Of these, the mechanical and thermal ones refer to the engine, i.e. to how much of the energy in the fuel is actually converted effectively into engine power output. Next, transmission efficiency refers to how much of this power is transferred to the wheels, and not lost with factors such as friction of the car components.

Of these, thermal efficiency is of interest to this study. Heat is generated during the combustion process of the engine, which is to be used to drive the engine. Losing such heat to things such as small air leaks means that less of it is converted into power, leading to a lower efficiency. This efficiency has basically been going up throughout time: newer car models use techniques that are better are conserving heat at places where this is needed. It is in effect a variable that says much about the technical quality of the vehicle: newer, more expensive cars will generally have higher quality engines with better thermal efficiencies than older, cheaper cars would. Differences exist between gasoline and diesel vehicles too, the latter having higher efficiencies. This is due to differences in air compression rates: this rate can be higher with diesel than with gasoline, meaning that the same amount of fuel can produce more air pressure and thus more power. Whereas gasoline engines have about 20 to 30 percent efficiency nowadays, diesel ones have 30 to 50 percent here. It must be stressed, though, that non-combustion engine vehicles – such as electric ones – are not included in this study, the reason for this being that such vehicles work fundamentally different from those with combustion engines: an electric car braking, for example, regenerates some of its energy, a technique called 'regenerative braking'. (US Department of Energy, 2003; Watzenig & Brandstätter, 2017, p. 86).

4.6) Engine Displacement ('V')

When talking about the 'power of a car', its engine is often referred to as having a certain number of litres, e.g. a '3L engine'. This is actually the engine displacement, which is a measure for the amount of power that an engine is able to deliver. It refers to the number of litres that its cylinders can hold at a time: the higher this number is, the more power an engine can deliver. The more fuel is injected into an engine at the same time, though, the less fuel efficient it will be (Ross, 1994).

Nowadays, high-duty trucks for the European market, such as the Man TGX, have about 15 litres of engine displacement. In comparison, cars that are branded as being fuel-efficient, such as the Toyota Yaris, hold about ten times less with 1.5 litres engines (Man, 2017; Autocar, 2017).

The relation between the engine displacement and the consumption is linear.

5) Car Model Development

In the previous chapter, many factors influencing fuel consumption have been explored. Of those, several car characteristics have been chosen to work with in the remainder of this study. These are:

- The mass of a car, in tonnes ('M');
- Its frontal area, in m² ('A');
- Its aerodynamic drag, as a coefficient ('C_D');
- Its rolling resistance, as a coefficient ('C_R');
- The thermal efficiency of its engine, as a coefficient ('n_t');
- And its engine displacement, in litres ('V').

In this chapter, five different cars are created, which differ on these exact characteristics only. It must be stressed that the numbers chosen for these variables are not based on any specific existing models, but have been set up as such that a group of strongly different cars would emerge: the extremities are explored here. It is, however, taken into account which values are more or less realistic for cars, which was determined in the previous chapter. Since formulae can only take into account so many factors, it is in no way implied that this would be the exact efficiencies for such cars: the values given are as with any model, at best, approximations.

The overall goal of this chapter is to answer the question *"For a chosen group of car models, how does driving at different speeds impact fuel consumption?"* To that end, graphs are shown for each car model, showing the fuel consumption for different speeds. It should be noted that this consumption is always relative to the speed, i.e. how much fuel is needed to drive at certain speeds.

The formula used to calculate the fuel efficiency is the one given by Ross (1994), as mentioned in the previous chapter:

$$\text{MPG} = \frac{120600}{\frac{3600 * k * V * < N >}{\text{vav}} + \frac{1}{\text{nt}} * \frac{1609 * \text{Cr} * M * g}{\varepsilon} + \frac{3.6 * p * \text{Cd} * A}{\varepsilon} * \frac{< v^3 >}{\text{vav}} + \frac{M * g * v * \sin\theta}{\varepsilon} + \frac{3600 * \text{Pacc}}{\text{vav}}}$$

As it has been determined previously which of the variables in this formula are to be constant, these values can already be entered. This leads to the following equation:

$$\text{MPG} = \frac{120600}{\frac{3600 * 0.225 * V * 31}{v} + \frac{1}{\text{nt}} * \frac{1609 * \text{Cr} * M * 9.8}{0.87} + \frac{3.6 * 1.2 * \text{Cd} * A}{0.87} * v^2 + \frac{M * 9.8 * v * \sin\theta}{0.87} + \frac{3600 * 0.5}{v}}$$

Which can be simplified to:

$$\text{MPG} = \frac{120600}{\frac{25110 * V}{v} + \frac{18124 * \text{Cr} * M}{\text{nt}} + 4.97 * \text{Cd} * A * v^2 + 11.26 * M * v * \sin\theta + \frac{1800}{v}}$$

For the remainder of this study, a km/l instead of a MPG notation is used. Since this means that the result is to be multiplied by 0.43, this results in:

$$\text{km/l} = \frac{51858}{\frac{25110 * V}{v} + \frac{18124 * \text{Cr} * M}{\text{nt}} + 4.97 * \text{Cd} * A * v^2 + 11.26 * M * v * \sin\theta + \frac{1800}{v}}$$

This is the base formula that is used throughout this study when calculating fuel efficiency and consumption. In the next sections, though, an even slightly simpler version is used, as the assumption in the test scenario of this chapter is that here cars will be driving at a constant speed through flat terrain, with sinθ = 0. As such, the final formula used for plotting the graphs in the next sections is:

$$\text{km/l} = \frac{51858}{\frac{25110 * V}{v} + \frac{18124 * \text{Cr} * M}{\text{nt}} + 4.97 * \text{Cd} * A * v^2 + \frac{1800}{v}}$$

The five car models and their fuel efficiency graphs are discussed next.

5.1) Car Model 1: Efficient Prototype

The first car model designed is an extremely efficient one, where all variables are optimised as much as possible for efficiency. It is light, extremely aerodynamic and has tires and a diesel engine of high quality. Such a car would difficultly come into being in practice – at least for now – which is why it is referred to as a 'prototype' here.

The final values used here are as in Figure 5.1.

Characteristic	*M*	*A*	*C_D*	*C_R*	*V*	*n_t*
Value	0.8	1	0.1	0.0075	1.5	0.5

Figure 5.1: Values for car characteristics of the efficient prototype

This leads to the fuel consumptions for different speeds as shown in Figure 5.2.

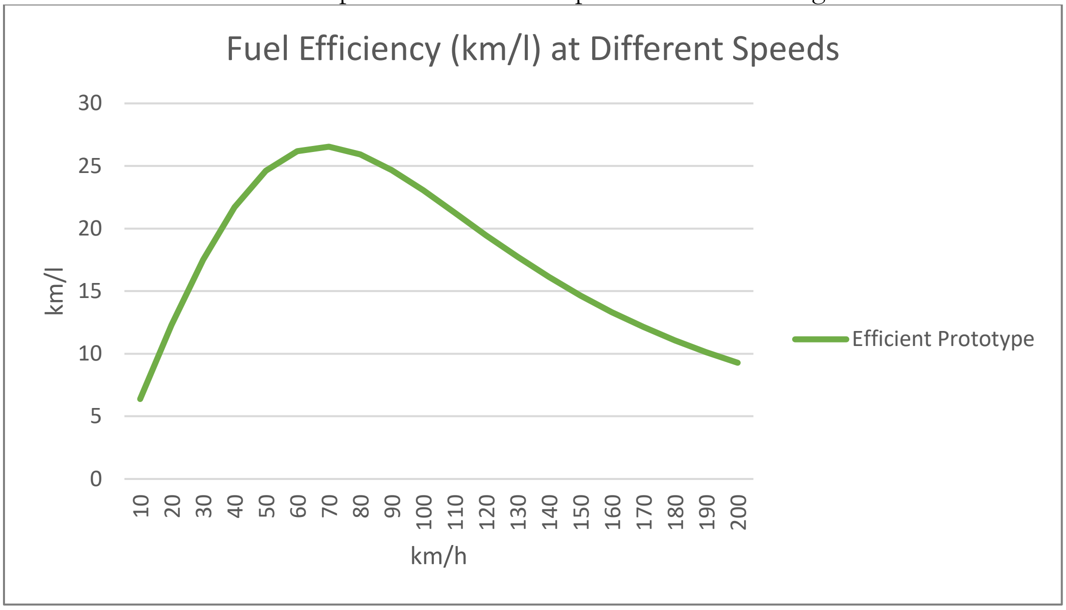


Figure 5.2: Fuel efficiency at different speeds for the efficient prototype

5.2) Car Model 2: Large Old Van

The second model is the exact opposite of the first one. Whereas that one was optimised for efficiency, this one has characteristics leading to significantly lower efficiency. Mass is of course higher, aerodynamic capabilities are poor, the tires worn out and the gasoline engine highly-powered and of relatively low quality.

The final values used here are as in Figure 5.3.

Characteristic	M	A	C_D	C_R	V	n_t
Value	3	3	0.3	0.015	5	0.25

Figure 5.3: Values for car characteristics of the large old van

This leads to the fuel consumptions for different speeds in Figure 5.4.

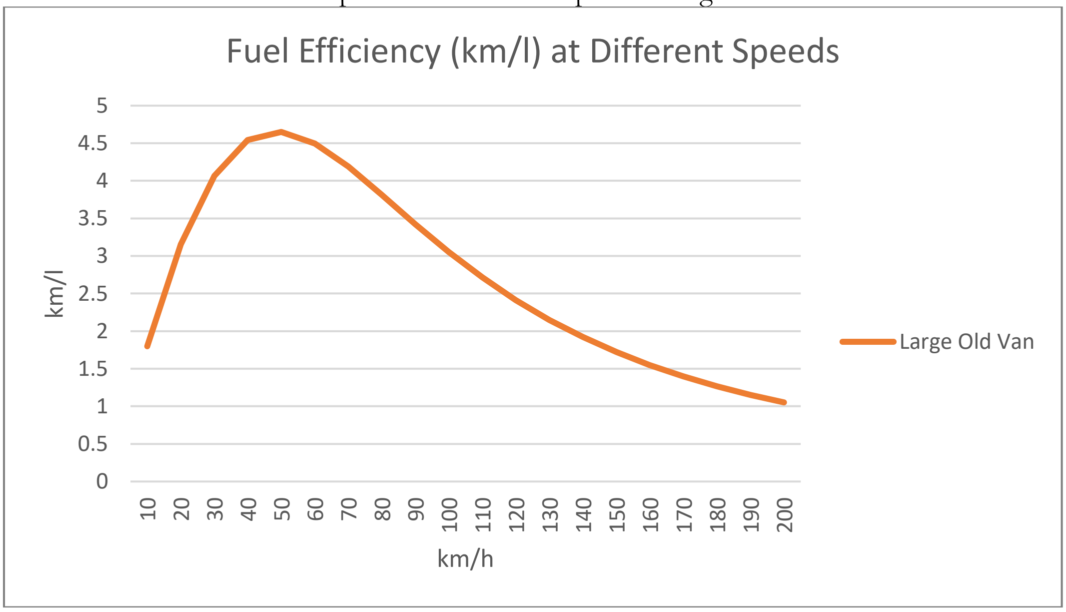


Figure 5.4: Fuel efficiency at different speeds for the large old van

5.3) Car Model 3: Luxury SUV

The third model represent a luxury SUV, which is more of a mix in terms of efficiency. This means that, while it is bulky and sports a powerful, demanding engine, it is built while adhering to high quality standards.

It has a high mass, strong diesel engine and does not possess great aerodynamic capabilities, but high-quality tires and good thermal efficiency.

The final values used here are as in Figure 5.5.

Characteristic	M	A	C_D	C_R	V	n_t
Value	2	2.2	0.25	0.01	4	0.5

Figure 5.5: Values for car characteristics of the luxury SUV

This leads to the fuel consumptions for different speeds in Figure 5.6.

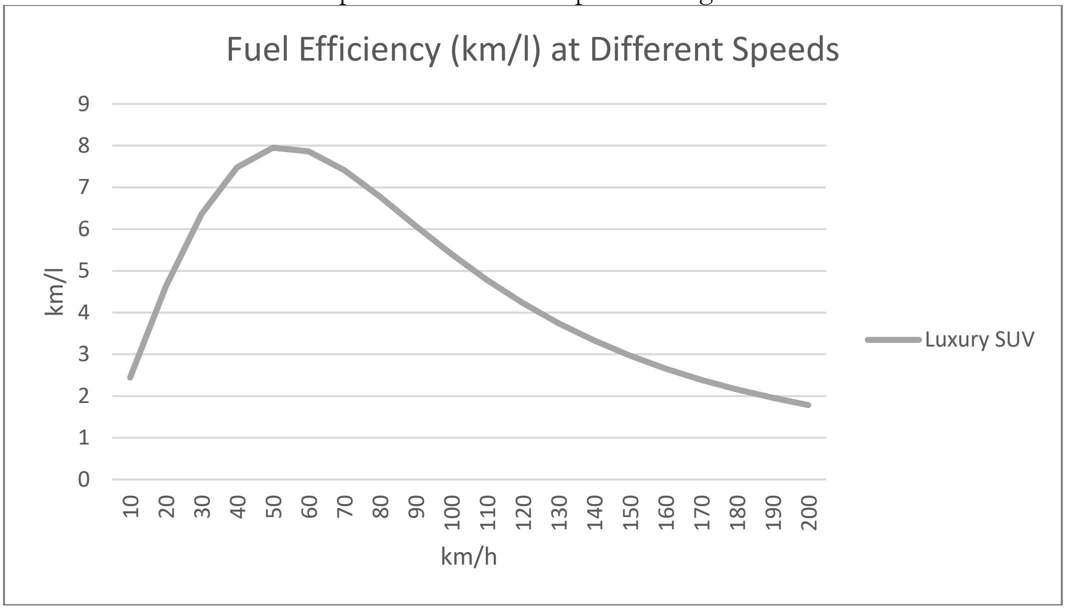


Figure 5.6: Fuel efficiency at different speeds for the luxury SUV

5.4) Car Model 4: Sportscar

This model is similar to the luxury SUV in that it is also of high quality and that it sports a powerful diesel engine. It is, however, lighter and way more aerodynamic. In summary, it has low mass, excellent aerodynamics, a powerful engine, high-quality tires and high thermal efficiency.

The final values used here are as in Figure 5.7.

Characteristic	M	A	C_D	C_R	V	n_t
Value	1	1.3	0.15	0.01	6	0.5

Figure 5.7: Values for car characteristics of the sportscar

This leads to the fuel consumptions for different speeds in Figure 5.8.

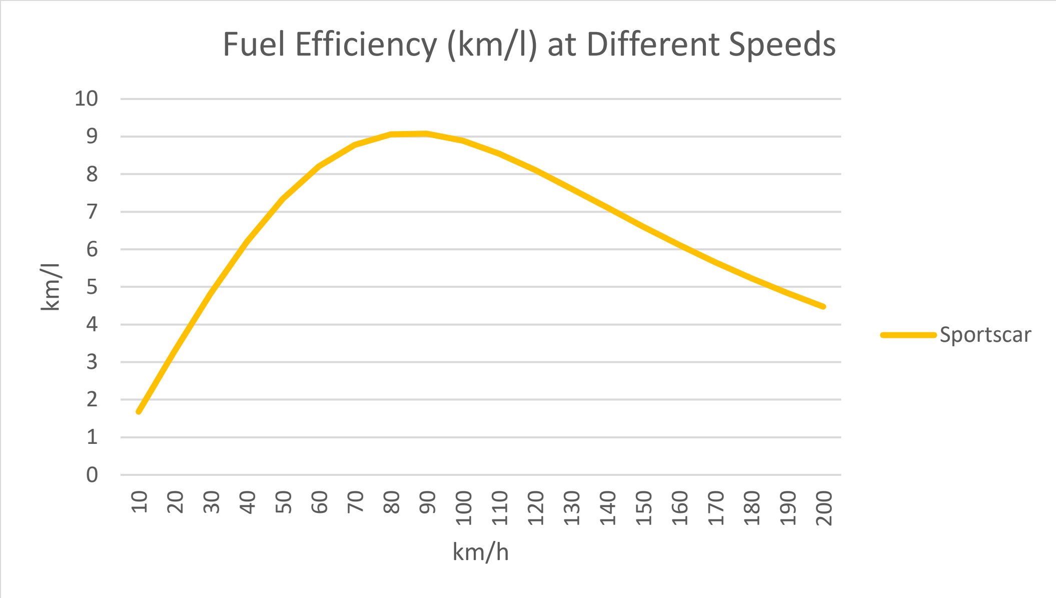


Figure 5.8: Fuel efficiency at different speeds for the sportscar

5.5) Car Model 5: Average Sedan

Finally, a car model is introduced that has average values for all characteristics, to represent a 'normal' family car. Its mass is neither low or high, its aerodynamic capabilities are fine but nothing special, the gasoline engine is not too powerful and the tires and engine are of medium quality.

The final values used here are as in Figure 5.9.

Characteristic	M	A	C_D	C_R	V	n_t
Value	1.5	1.8	0.25	0.01	2	0.3

Figure 5.9: Values for car characteristics of the average sedan

This leads to the fuel consumptions for different speeds in Figure 5.10.

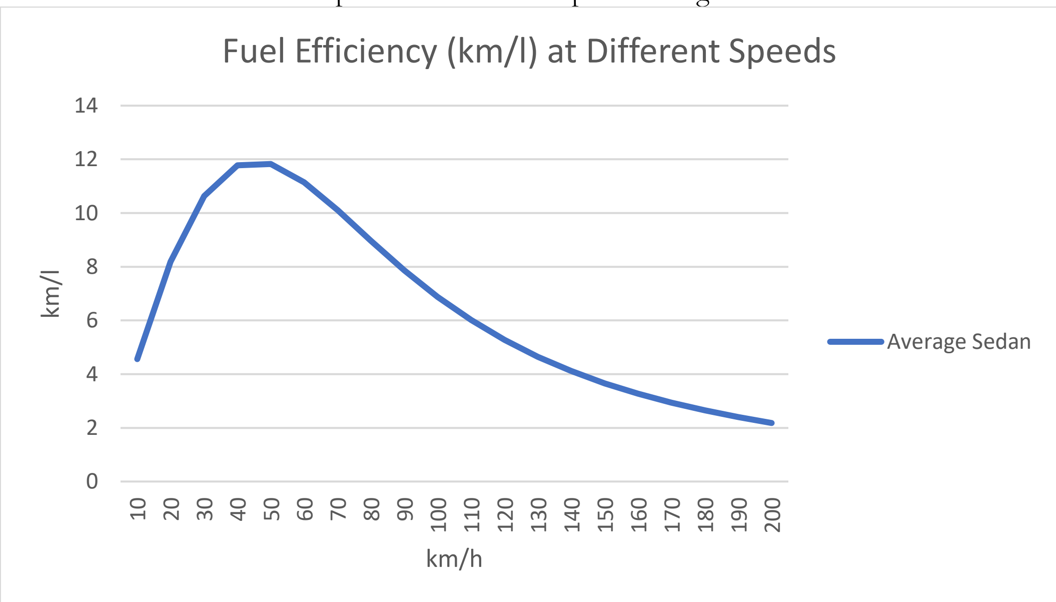


Figure 5.10: Fuel efficiency at different speeds for the average sedan

5.6) Car Models Comparison

From the graphs in the previous sections, it is difficult to see the differences between the five car models directly. As such, all efficiency curves are shown together in Figure 5.11, which leads to some interesting insights:

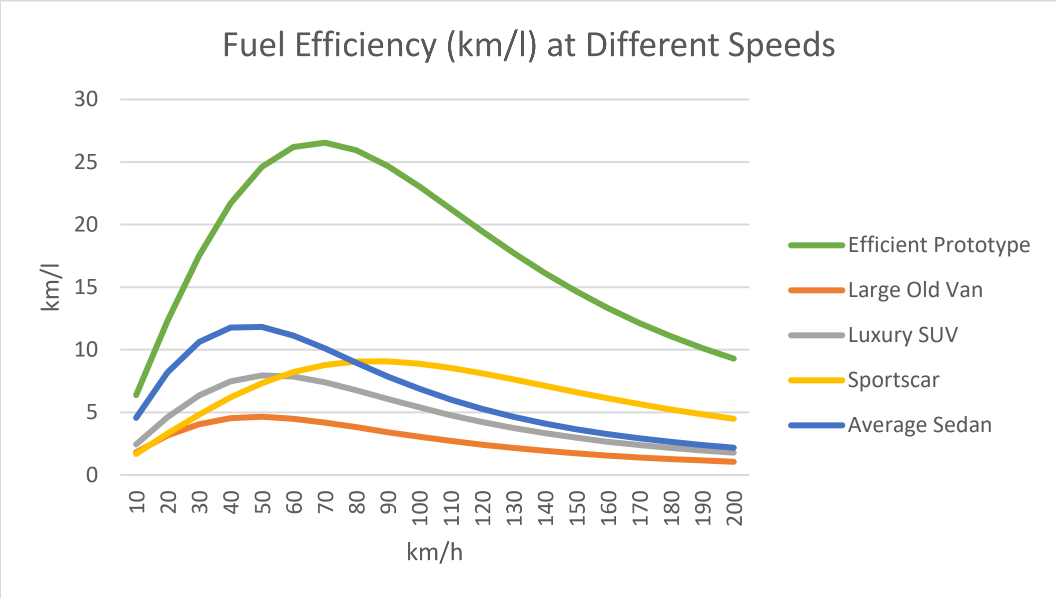


Figure 5.11: Fuel efficiencies at different speeds for all car models

Here, differences become visible. Though all of the curves follow the same logic – an early peak with a rapid descend afterwards – the heights and slopes of the curves, as well as the speeds at which the efficiencies peak, are different. Some observations can be made:

- Though it might be logical to assume that higher speeds are always more inefficient, these graphs show that this is not the case. In fact, a significant boost to fuel efficiency can be achieved by raising cruising speed at first. While it might seem counterintuitive, this is actually quite logical if kept in mind that fuel efficiency is consumption relative to speed. Even when a car is not moving, e.g. at a traffic light, engines consume fuel to continue functioning: they idle. As such, there will always be a base value of fuel consumed to keep the engine running. At a speed of 0, this would lead to an efficiency of 0, as no distance is travelled at all with the fuel used. When increasing speed, this idling value is gradually offset by a higher speed and thus larger distances travelled, leading to better efficiencies. This is not an infinite process though: eventually – at the efficiency peak – increasing speed will cost more fuel to overcome factors such as air resistance than it offsets the base idling value, leading to declining efficiency (Ross, 1994).
- After the optimal efficiency peak, efficiency starts to go down quickly. Though the efficiencies of the different car models vary strongly, this is not the case for the values observed at high speeds: all cars experience significantly reduced efficiencies as such speeds, showing that going past these optimal speeds is generally not a good idea in terms of fuel economy. In the case of the efficient prototype, for example, accelerating from 100 to 120km/h leads to an efficiency drop of about 20%.
- Efficiency peaks early for all car models and does not achieve remarkable values. All models peak in the range from 40 to 80km/h, which in reality is on the low side: most sedans have an optimal speed of about 80 to 90km/h, for example, with lower optima for heavier-duty vehicles. Km/l values are also slightly off, with real values for sedans approximating 15km/l, as opposed to the 12km/l in this study's model. While not optimal, it has logical reasons: a complex machine such as a car can simply not be caught perfectly in mathematical formulae or models. By omitting factors that might also potentially affect efficiency but that are too complex and populous to include, calculation accuracy suffers too. This does however not pose a problem to the remainder of this study, as the real interest lies in – among other – the relative differences between car models – which, as can be derived from Figure 5.11, exist – and not so much in absolute values.

6) System Requirements & Blueprint

In the two previous chapters, a literature study was carried out on what theoretically influences fuel consumption in cars. With these results and the subsequent conclusions, it is now possible to answer this next sub-question: *"Which requirements exist for data and methods for building the desired navigation system?"*

The answer to this question is a collection of different methods and data requirements, based upon the literature study of Chapter 4 and 5. In the first section, the data requirements are summed up. With the data requirements determined, it is possible to move on to the methods to be used in the application. Here, a distinction is to be made between methods for the pre-processing of data and for the actual application. The first relates to the preparation of data for use in the application, which only needs to be done once. As such, it is not part of the application, but done separately before the application itself is built and run. This pre-processing in discussed in the second section. The third section, finally, presents the application blueprint and discusses the application's methods and their implementation.

6.1) Data Requirements

In the methodology section of this study, Chapter 3.2, the data that was used in the end is discussed. The need for some of these sets was already established early on: some methodology requirements, however, only came to light while answering the sub-question of the current chapter. These additional data requirements are discussed here.

Based on the answers to the previous sub-questions, one extra requirement was identified: the need for information on road slopes, i.e. a height map of Zuid-Holland. This lead to this list of datasets deemed to be required for building the current application:

- Zuid-Holland road network (Chapter 3.2.1);
- Maximum speeds per road segment (Chapter 3.2.2);
- Traffic (Chapter 3.2.3);
- Dynamic maximum speeds (Chapter 3.2.4);
- Height map.

In the end, the height map was not used for reasons of considerable calculation times. It was decided still to mention it here though, as it had been previously established that, ideally, it should be included in such an application. It could have improved – and might improve during a future study – the data quality. As such, it is deemed relevant enough to still be discussed in the next section.

6.1.1) Height Map

One of the factors influencing fuel consumption, as found previously, is the power needed to overcome grades while driving. Naturally, this power depends upon the slope of the road. This slope is not given by default in the road network dataset, which is why it would have been needed to add it.

As no readily available dataset has data on road slopes, these have to be calculated. This could have been done by overlaying the network with a height map of Zuid-Holland, after which differences in heights, and thus slope, could be calculated: the exact calculation method is discussed in the next section, Chapter 6.2, though.

Height maps are in fact available for the Netherlands as raster maps with resolutions as high as 50cm, covering the entire country. Multiple height datasets are available, differing in moment of time, resolution and theme. The most recent, complete 50cm-resolution map of the surface – i.e. without taking into account construction heights – was deemed to be the most suitable here: 'AHN2 50cm maaiveld' (PDOK, 2017).

6.2) Pre-processing

The pre-processing part of this study consists of downloading, editing (when necessary) and combining datasets in order to create the datasets needed for the application. Whereas actions performed within the application are performed every time that it is run – e.g. calculating the fastest path from A to B – pre-processing is only done once and outside of the application: its results serve as inputs for it.

As was determined in the previous section, five initial datasets were ideally to be used: the Zuid-Holland road network, maximum speeds for these roads, traffic data, dynamic maximum speeds and a height map – which, as mentioned in the previous section, was not used. The output of the pre-processing phase, or 'database creation', is a database with roads as objects and data deducted from the aforementioned sets as attributes.

To get to this, the road network first needs to be loaded into ArcMap, where it can be viewed and edited. It is then joined with the maximum speeds data, based on 'WVK_ID', which serves as a key attribute in both the roads and speed limits dataset. Next, in steps of 10km/h and for the range between 30km/h and 130km/h (the highest speed limit in existence in the Netherlands) attributes will be added for the custom speed limits: when adhering to a self-imposed limit of 100km/h a 130km/h road effectively becomes a 100km/h, and so on. Minimum speeds will also be taken into account here.

The same is done for dynamic speed limits, adding speed limit attributes for cases with the low variable speeds – the situation during the day – and the high ones – during the night. Finally, this is done too for cases with and without traffic, in which the effective speed limit becomes lower.

After joining these datasets, the heightmap was supposed to be introduced in ArcMap and overlaid with the road network. The goal was to calculate the grade of the road, which is actually not always a single number. Road segments can of course have different grades at different points. As such, an average grade would be used here, which is the steepness of the road when a straight line were drawn between that road's start and end point. The accuracy of this actually being the grade along most of the road depends upon its length: longer roads allow for more variation with actual grades further away from the average one. Two arguments can be made for using the average grade anyway, though. First of all, the actual lengths of roads in the network used are limited. As is shown in Figure 6.1, the average road length is only 151.2 metres. The longest one is 14,063.5 metres long, which does allow for quite some variation in grades along the way. As can be seen from the frequency distribution, however, the vast majority of roads has a small length, leading to only a few cases in which this problem could occur.

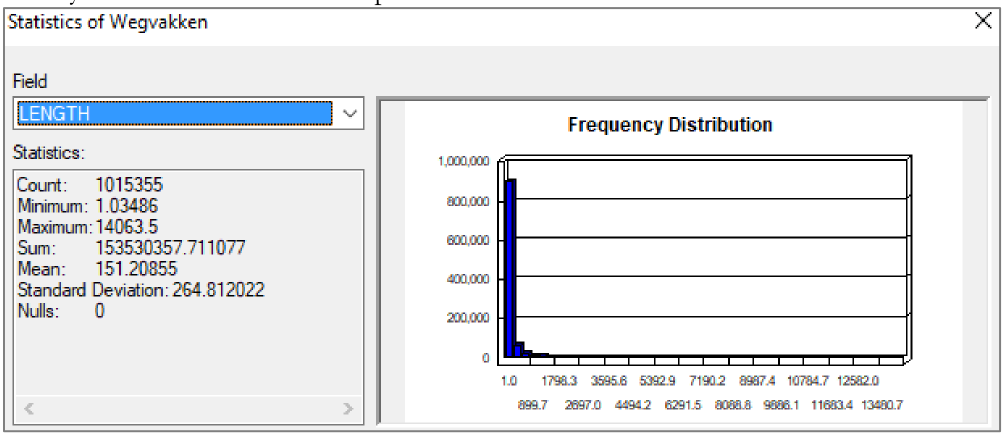


Figure 6.1: Statistics and frequency distribution of road lengths

Furthermore, and perhaps most importantly, grades in the Netherlands are very limited. It is commonly referred to as the flattest country on earth. As can be seen in Figure 6.2, only the east and southeast of the country have some kind of altitude differences that could be called hills. Still, with a difference of only 329 metres between the lowest (7m below sea level) and highest (322m above sea level) points, any real height differences are negligible. More importantly, Figure 6.3 shows that slopes are also non-existent in most of the country. This is an important argument for the case that the results from the built application are still valid, even without using grades: their influence would have been small in any case, due to this large lack of height differences.

After calculating the grade attribute, all that would have been left would be to load the road network with all of its newly created attributes into the PostgreSQL database for later access within the application. This was still done, albeit without data on the slopes. 'shp2pgsql' was used here, a geodata loader for shapefiles (the data format used in ArcMap) included in the PostGIS extension used with PostgreSQL (PostGIS, 2017).

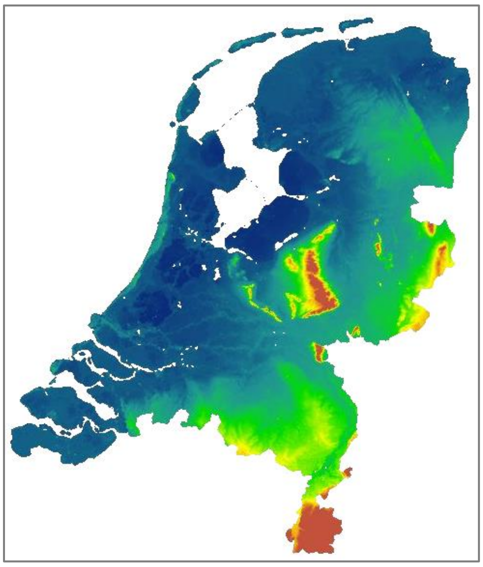


Figure 6.2: Height map of the Netherlands

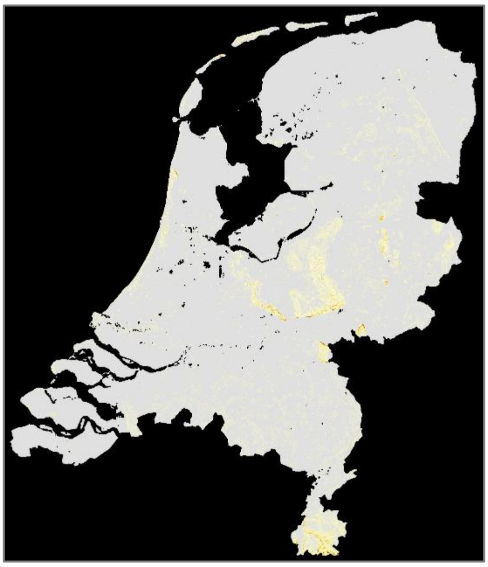


Figure 6.3: Slope map of the Netherlands

6.3) Application Design

Now that all of the requirements for the application and its input data are known, a design for the application itself can be drawn. As mentioned before, the technology used to build this is a website, instead of e.g. a stand-alone application that would have to be installed on a computer.

When building a website, it is imperative to think about the website architecture. Two pivotal concepts here are client-side and server-side operations. Everything happening on a website needs to be executed by either the client (i.e. a web browser) or server (a web server hosting the website). Some operations can only be performed by either, e.g. only the client can show the actual website to the user. However, some operations could be allocated to either. The question thus becomes: what does the client do, and what does the server do? This is highly dependent upon what needs to be done by the website as a whole. A typical client-server setup, with tasks allocated to both the client and server, is shown in Figure 6.4 (Dumke et al, 2003, pp. 22-23).

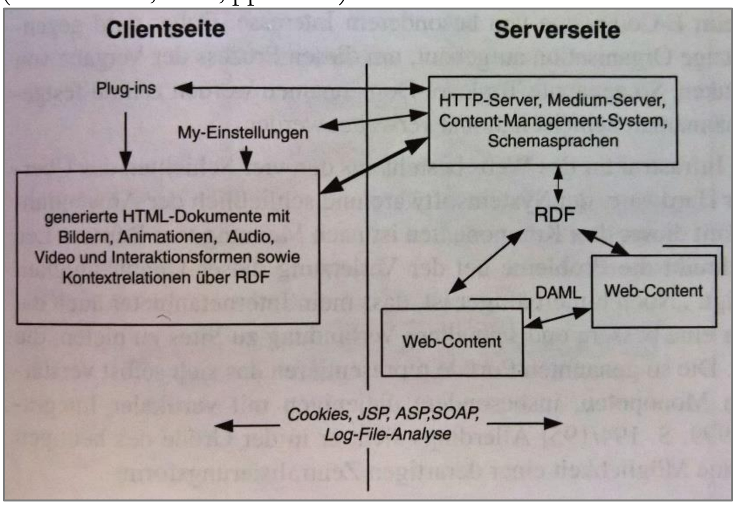


Figure 6.4: Typical client-server architecture (Dumke et al., 2003, p. 33)

In this case, the server-side can be split in two: a web server hosting the actual website and a database server hosting the necessary data. Including a database server in the setup actually makes it easier to determine which tasks should be performed where. For a database server to be accessible, a connection needs to be made to it. For safety reasons – e.g. making sure that its content is not deleted by malevolent users – this connection needs to be made using procedures such as usernames and passwords. These must of course be kept private if security is to be maintained. This makes it an extremely bad idea to have the client establish this connection, as users can always read the source of the client, and thus the secure data. As such, any tasks involving the use of database information must be handled by the web server, instead of the client (Letzel & Gacki, 2001, p. 258).

As all of the actual path calculations rely on the database data, only some tasks are left for the client. This includes setting settings – what does the user want to calculate? – and presenting the results of the calculations to the user. This is often referred to as a 'thin client' setup, in which the web server does the actual heavy-lifting of calculations and the client's main task is visualisation (Held, 2000, p. 80).

During normal, end-user use, the user would tell the website what to do – e.g. calculate a route from this to this point with this car model, including traffic and during the night – but for the current study more is required. As the fourth stage of this research, the practical research, will involve statistical analysis, a large number of path calculations is to be performed (see Chapter 3.1.3). Theoretically, these runs could be calculated one by one using the abovementioned method of setting every single run separately. However, as 135,000 path calculations are necessary, this would be highly impractical. As such, a so-called 'developer mode' is introduced, next to the normal 'user mode'. In both modes, the user (or developer, i.e. the researcher behind this study) configures what needs to be calculated, and leaves the application to do this. What happens after doing so differs between modes, though. In the user mode, only one single run will be done, after which the ArcGIS API and graph plugin (both of which will be explained in the remainder of this chapter) present the results to the user, both as a map and as a graph. For the developer mode, however, the only feedback will be a confirmation that the calculations have been performed. Here after every run, instead of returning and presenting the results to the user, these are stored in the database for later use.

All of these considerations have led to the website blueprint of Figure 6.5. Here, the three main parts of the application – the client, web server and database server – are shown with, as blue rectangles, the components that are handled by each part. These components are discussed in more detail in the next subsections. The arrows linking components describe data flows between them, the arrow direction showing the origin and destination of such data. Finally, these lines also show in which modes certain dataflows exist, as explained in the previous paragraph.

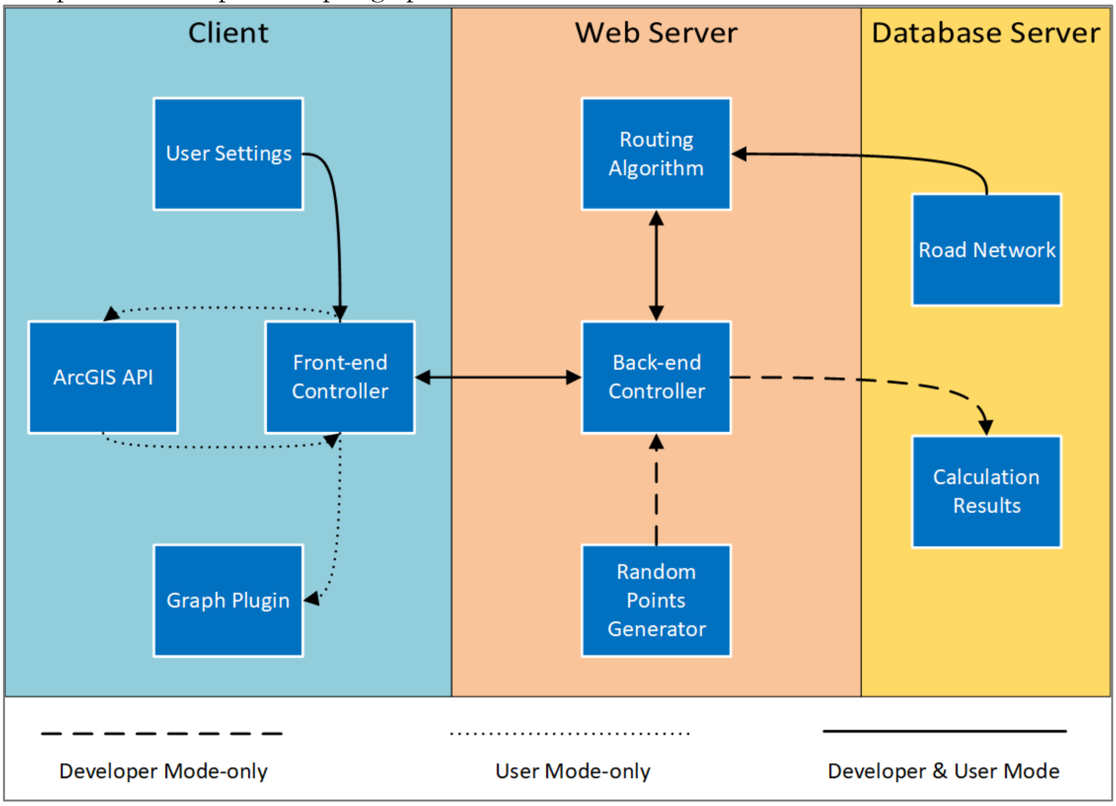


Figure 6.5: Application blueprint

6.3.1) Front-end Controller

As the name implies, the front-end controller is the core of the client-side of the application. It is in fact only a file called 'index.html', the file that web browsers automatically request when connecting to a website. This request is done using HTTP, or the 'Hypertext Transport Protocol'. HTTP allows for clients to 'GET'

– i.e. request – or 'POST' – i.e. transfer – data from or to the server, among others. It is also used to communicate with the back-end controller, which is discussed in Chapter 6.3.5 (Dumke et al., 2003, p. 24).

When received from the server, the web browser will interpret and visualise data. This data can be written in different languages, of which HTML, or 'HyperText Markup Language' is the most common. HTML is the core of any website and tells the client what to visualise: it holds the contents of the website, such as a piece of text. It does, however, not necessarily say anything about how this piece of text would look like: its colour, font, size etc. is not included. While this can also be specified in HTML itself, it is good practice to instead use a separate CSS, or 'Cascading Stylesheets', file. As a rule of thumb, HTML should refer to 'what' is shown, and CSS to 'how' it is shown (Born, 2011, p. 29; Laborenz, 2016, p. 20).

These HTML and CSS parts are what make up the front-end controller. It is the backbone of the website, controls everything that the user sees and handles communication with the server-side of things. The other client components, i.e. the user settings, ArcGIS API and graph plugin, are all built on top of this controller and communicate via it.

6.3.2) User Settings

The user settings are part of the client-side. They are built on top of the front-end controller and are thus part of the 'index.html' file. It includes, as the name suggests, settings for the user to alter. As was mentioned in earlier in this chapter, two different modes will exist: the user and developer modes. It is in the settings that a choice needs to be made between these two modes. Depending on the choice, which should be the first one to be made when using the site, different settings will be presented.

When choosing the user mode, the website is shown as if to the end-user. After choosing a start and end point – which is handled by the ArcGIS API component – the user can set the car model to be used and determine whether or not traffic is to be used. In addition, a choice must be made between using day-time and night-time speeds. After doing so, a 'Go' button can be clicked, which tells the front-end controller to communicate to the back-end controller that a path calculation is to be performed.

When choosing the developer mode, however, other options are presented. This is not what an end-user would see and is only meant to be used for generating results for the statistical analysis of the current study. As such, it would not be part of any final, public version. It has two options, the first of which is the number of runs that the application should execute. Each run consists of calculating the fastest paths between one pair of points for all possible combinations of car models, speeds limits, dynamic speeds and traffic situations. This is included so that, depending on calculation times, more or fewer runs could have been executed later on. It is also a way to be able to continue calculating, should unforeseen events interrupt this. The second option determines whether or not results of previously executed runs are kept – i.e. the results of the new runs are added to the existing results – or that all previous results should be deleted to start anew. While in most cases keeping old results is best (e.g. to calculate 10.000 runs per day over ten days instead of 100.000 in one go), it is sometimes necessary to delete old results, e.g. when changing the routing algorithm. As in the user mode, the calculation is started by clicking the 'Go' button.

While the content of the user settings – i.e. the text, text boxes, sliders, etc. – is written in HTML and CSS, as is the front-end controller, an extra language is needed here. While the aforementioned languages can show content, they are not able to handle any input from the user. Introducing JavaScript makes a website dynamic, i.e. interactive. JavaScript consists of scripts that dictate what to do in certain events, such as when the 'Go' button is clicked. It can also store variables, such as the amount of runs that the user input (Koch, 2011, p. 10).

6.3.3) ArcGIS API

In Chapter 3.3.5, ArcMap was discussed. This stand-alone application is part of the ArcGIS platform by Esri, of which the ArcGIS API is also part. While used for other purposes, it does share some functionality with ArcMap and access to the same Esri resources. This access to tools such as the Esri geocoder has made its use preferable over the use of similar products, such as Leaflet (Leaflet, 2018).

Before discussing the ArcGIS API any further, it is necessary to know what an API is. APIs, or 'Application Programming Interfaces', are interfaces that allow for communication with external applications or services. They can also be seen as website plugins. Important to realise is that, different from 'normal' plugins, APIs rely upon these external applications and are not able to function properly without a connection to them (Association for Computing Machinery, 2017).

This specific API connect to the ArcGIS services. When included in a website, it creates a map box which can be used to show maps, as well as to interact with them. Here, the interaction needed is the ability to specify the start and end points for the calculations, when in user mode. This is done by right-clicking the map. Also in user mode, the paths calculated are drawn on top of the base maps. In both modes, the map can be manipulated by panning and zooming in and out. While versions in different programming languages, such as Java, JavaScript and .Net, exist, the JavaScript one was chosen here. This choice was made from a practical point of view, as other components – e.g. the user settings – are also written in this language (Esri, 2017b).

6.3.4) Graph Plugin

When in user mode, the results of the path calculations are to be shown. While the paths themselves are presented using the ArcGIS API, the differences between the paths in time and fuel consumption are shown in two graphs. In both graphs, a line is shown: one for time, and the other for fuel. Both are plotted for the different custom maximum speeds used. By visualising the results as such, users can see the differences in time and fuel caused by adhering to different speed limits.

As with showing the maps, a plugin is needed to show the graphs. Here, the open source Chart.js plugin is used, 'js' being an abbreviation for JavaScript, the scripting language used throughout the client-side of this application. By downloading and including this plugin, it can be used to draw many different types of charts, of which the line chart will be used here. As opposed to the ArcGIS API, it is not an API, as it does not communicate with or rely upon external applications or services: everything that is needed, is included in the Chart.js file (Chart.js, 2017).

6.3.5) Back-end Controller

Just as the front-end controller was the core of the client, the back-end controller is the web server's core. It is used to communicate with the front-end controller and the other components of the web server, i.e. the algorithm and random points generator. It tells the algorithm what to compute (e.g. the fastest path between A and B without traffic during the night) and either returns the result to the front-end controller – when in user mode – or stores it in the results database – when in developer mode.

It consists of a file called 'routing.php', which also holds the two other web server components. As it is not a client-side component, it is not executed by a web browser, but rather by a server. This server is managed by XAMPP, as described in Chapter 3.3.3. Servers use other languages than clients do, meaning that HTML, CSS and JavaScript cannot be used here. Rather, PHP, or 'PHP Hypertext Preprocessor', is often used. The main difference between client and server languages is that whereas client languages are interpreted (i.e. read by the browser and translated into an object in that browser), server ones are executed and return a certain result (e.g. the outcome of a mathematical calculation). PHP allows for receiving the input sent by the front-end controller (i.e. the user settings), doing something with that and outputting something back to the front-end controller. As the client only receives the final results of the calculations and has no knowledge of how these calculations were performed, it can be used to process sensitive information, such as database credentials (Theis, 2013, p. 18).

In communicating with the results database (and for the communication between the routing algorithm and the road network) an additional language is needed: SQL, or 'Structured Query Language'. This language is used to 'query' a database, i.e. asking for a record or inputting or deleting one, or changing anything else in the database. Such SQL queries can be sent to the database using the PHP function 'pg_query()', which was specifically built to query PostgreSQL ('pg') databases (Letzel & Gacki, 2001, pp. 195-197; PHP Documentation Group, 2017).

6.3.6) Routing Algorithm

The routing algorithm component is what actually calculates the fastest paths, given a pair of geographical locations (the start and end points) and parameters for the speed limit, traffic usage, etc. These parameters are provided by the back-end controller, to which the results of the algorithm calculation are also transferred in the end.

As with any routing algorithm, a road network is needed. This is the network accessible in the road network component. Communication with this component is managed by SQL, as discussed in the previous section.

The algorithm itself is Dijkstra's algorithm. Though dating from 1959, it is still one of the most applicable and used shortest-path algorithms. The logic behind Dijkstra's algorithm is that it basically spreads from the starting points to other points evenly, as in a circle that becomes larger and larger. When a node (road intersection) is selected, all connections to nodes that have not been visited yet, are checked. Here, impedances are used as travel costs. Such impedances can be anything: distance, time, money or whatever else might be relevant. In the current study, time is used as an impedance, which can be calculated using the length and speed limit for a certain road. After having established all connections from a certain node, the current node is marked as 'visited', meaning that it will not be considered anymore by the algorithm. Next, the node that could be reached with the lowest costs up till this point is selected as the new current node, after which the aforementioned process is repeated. All this is done up to the point at which the desired end node is found. When this happens, the shortest path between the two points can be found, as every found node points to its predecessor. By backtracking, i.e. following these predecessors from the end node, the starting point can be found again. An example graph, with the shortest path from A to D, is shown in Figure 6.6, with the shortest path as a thick line (Zhan, 1997).

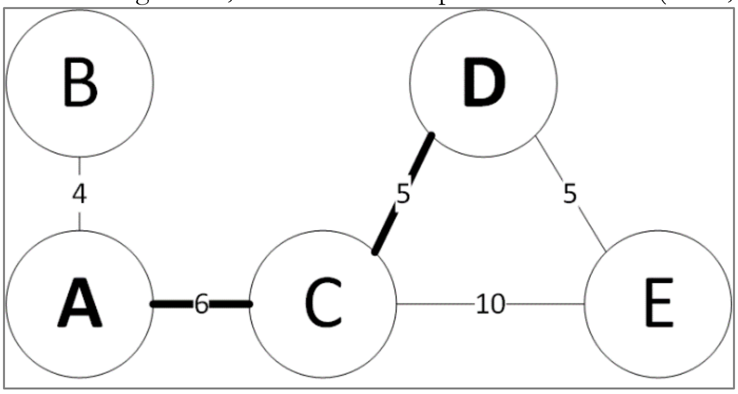


Figure 6.6: An example of a graph for Dijkstra's algorithm

Using this algorithm is the preferred way to go, as the calculation time is linear to the number of nodes in the road network. Dijkstra's algorithm cannot be used in every shortest-path case, however. By always selecting the node that can be reached with minimum costs, it is implied that no better path to this node exists. This is only the case with positive values for costs: when introducing negative costs, it is possible for the shortest path to use a road that may initially be expensive, but that pays itself back after that with negative costs. An example here could be the network in Figure 6.7 in which, again, the shortest path between A and D is wanted. As travel costs between C and E are 10 and between C and D directly 5, Dijkstra assumes that it would make no sense to visit E at all, since the destination can already be reached with less costs. Since the cost of going from E to D is -20, however, it would actually make sense to follow path A-C-E-D, which in this case is cheaper than A-C-D.

As has thus been shown, Dijkstra might return non-optimal paths for networks with negative costs. In such cases, alternative algorithms such as Bellman-Ford are to be used. As time is used as an impedance in this application, however, and time travelled can never be negative, this problem is not present here. As such, Dijkstra can and will be used (Yan, 2017).

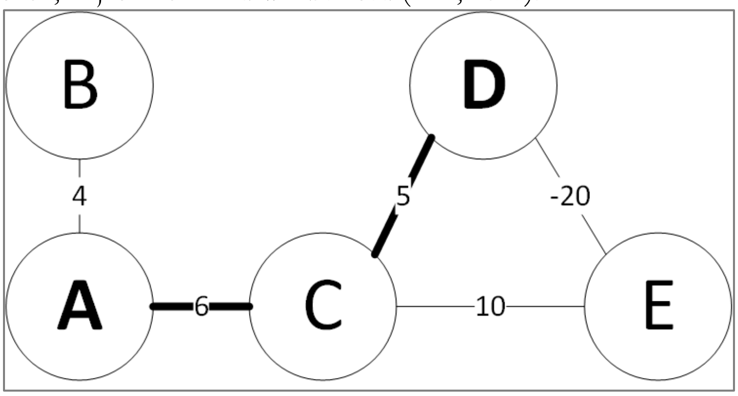


Figure 6.7: An example of a graph for Dijkstra's algorithm with negative impedances

In addition to connections between nodes, nodes themselves can have costs too, e.g. when traffic or traffic lights lead to waiting times at intersections. Unfortunately, no complete dataset exists yet with traffic lights and average time lost there. The 'Basisregistratie Grootschalige Topografie' ('BGT'), a large-scale map of the Netherlands featuring its topography, does include a large variety of objects such as roads, buildings, benches and signs. Traffic lights are included as 'verkeersregelinstallatiepaal' objects here too. Unfortunately, it is part of the optional objects that source providers – such as municipalities and provinces – do not necessarily need to include. Visual inspection of the BGT reveals that such optional objects, including traffic lights, have in most cases been left out. As such, it is not possible to use the BGT as a source for this. Another possible source would be OpenStreetMap (OSM), which also includes a tag for traffic lights, called 'traffic_signals'. Since OSM is entirely built by volunteers, though, it suffers from the same problem as the BGT does: lack of coverage. While many objects are included here, visual inspection shows that traffic lights are barely mapped here either (Ministerie van Infrastructuur en Milieu, 2017; OpenStreetMap, 2017b).

As such, while inclusion would be ideal, with the currently available data it is not possible to use traffic lights for costs at intersections. Another option would be to use a fixed cost at intersections, e.g. counting 10 seconds of waiting at each junction. This would, however, introduce the problem of detecting what 'real' intersections are. Two lines meeting at a node does not necessarily mean that this represents an intersection: on a highway, for example, this could represent a lane for incoming traffic. In this example, traffic already on the highway would not have to wait as they have the right of way, whereas incoming traffic might have to. In addition, on many local roads multiple intersections exist within small distances of one another. While in practice crossing times would be minimal – since local roads do not usually carry large amounts of traffic – adding a fixed cost for every intersection would actually make the use of such roads very costly. This would lead to unrealistic outcomes, which is why this option cannot be used here either. Costs are thus only added for traversing roads themselves, and not for intersections.

Since the version of Dijkstra used here is a common one, pgRouting could be used to calculate the paths. pgRouting is an extension to PostGIS, which in turn extended the functionality of PostgreSQL. It offers some often-used routing algorithms, including Dijkstra's. Though it was originally envisioned to write this algorithm tailor-made in PHP, this turned out to add a substantial overhead cost by making it necessary for the web server to communicate with the database one many times during a single path calculation. By using pgRouting, which is executed entirely by the database and only requires communication of the input and output, calculation times were reduced significantly (pgRouting, 2018).

Finally, after calculating the shortest path in terms of time, the fuel consumption for this shortest path is calculated using the formula in Chapter 4.

6.3.7) Random Points Generator

When in developer mode, no user-specified start and end points are used to calculate paths between. Instead, the Monte Carlo (see Chapter 3.1.3) way using random pairs of points is used.

Though random geographical coordinates could have been used that would then be matched to the nearest intersection, it proved to be much more practical to simply select intersections from the total list of nodes randomly right away. This has the added benefit of preventing any geographical bias towards intersection on the edge of network, as these would be chosen whenever a random point would fall outside of the province. In order to do this, SQL is used to query the road network database and to return two intersections at random, while making sure that the two are never identical.

6.3.8) Road Network

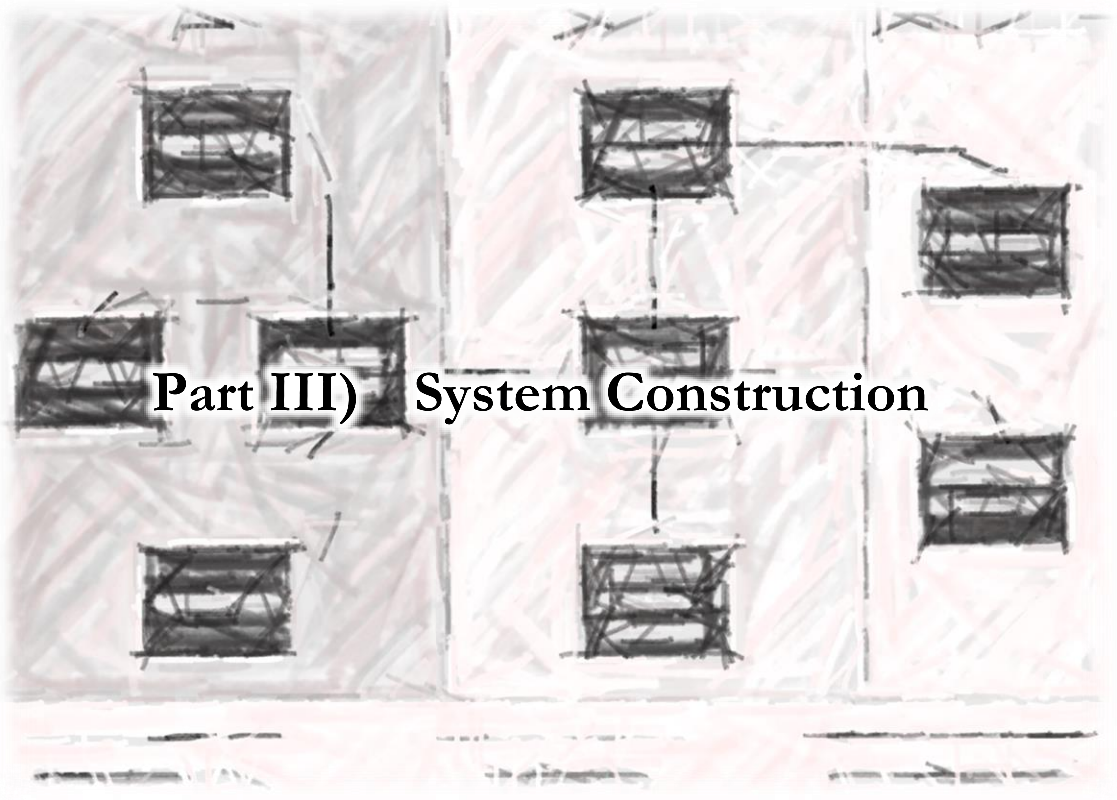
The road network and all of its attributes, as created in the pre-processing phase (see Chapter 6.2), is stored in a database for reading and usage by the algorithm component. This communication is established using SQL, as mentioned before. It is hosted on the database server, which is a PostgreSQL server (see Chapter 3.3.1). No changes are ever made to the data stored there: only retrieval of data is necessary for the proper functioning of the application.

6.3.9) Calculation Results

As with the road network discussed in the previous section, the calculation results are stored on the PostgreSQL database server, from which they can be exported again to files in formats such as .csv – comma separated values. It is separate from that road dataset though, as it is the output – and not the input – of the calculations performed by this application. As opposed to that dataset, it is not always used. Only when in developer mode, the results are stored in this database for later use in the statistical analysis. In user mode, these results are instead returned to the client for visualisation, without being stored permanently.

Every path calculated is a record here. Attributes detail the run number and the parameters used, i.e. the start/end point pairs and car model, traffic option, dynamic speed option, speed limit used, in addition to the time and fuel needed.

Part III) System Construction



7) Database Creation

The previous chapters have detailed the plan for the collection, editing and use of data in this study. Eventually though, every plan is only as good as its execution. In virtually every case, unforeseen circumstances will force initial ideas to be changed, or abandoned entirely. As Meredith et al. (2014, p. 44) said: *"Projects almost never proceed through their life cycles without change."* Such scope changes are not necessarily a bad thing, but should be documented carefully in order to avoid any confusion as to what was, and what was not, implemented in the end.

This chapter serves mainly as such an operational and technical description, detailing the steps taken during the database creation and how these steps differ, in some cases, from the plan as envisioned originally. As such, it can also be seen as a process report of the database creation. The order of the steps that follow, is the order in which these steps were chronologically executed during the database creation process.

7.1) Attributes Clean-up

The basis for the roads network was the 'Wegvakken' dataset, as described in Chapter 3.2.1. This dataset includes, in addition to the geometry of the roads, attributes with more information on these roads. Not all of this information is necessary, however. In order to decrease the size of the set and thus increase the speed with which it can be processed, such unnecessary attributes were dropped from 'Wegvakken' before editing it in any other way.

These discarded attributes are: 'WVK_BEGDAT', 'WEGDEELLTR', 'HECTOLTTR', 'BST_CODE', 'RIJRICHTING', 'GME_ID', 'GME_NAAM', 'HNRSTRLNKS', 'HNRSTRRHTS', 'E_HNR_LNKS', 'E_HNR_RHTS', 'L_HNR_LNKS', 'L_HNR_RHTS', 'BEG_AFSTAND', 'END_AFSTAND', 'BEGINKM' 'EINDKM', 'POS_TV_WOL', 'WEGBEHCODE', 'WEGBEHNAAM', 'DISTRCODE', 'DISTRNAAM', 'DIENSTCODE', 'DIENSTNAAM', 'WEGTYPE', 'WGTYPE_OMS', 'ROUTELTR', 'ROUTENR', 'ROUTELTR2', 'ROUTENR2', 'ROUTELTR3', 'ROUTENR3', 'ROUTELTR4', 'ROUTENR4' and 'WEGNR_AW'.

The attributes that were left and their meanings are as follows:

- 'WVK_ID': A unique identifier for the stretch of road.
- 'JTE_ID_BEG': An identifier for the junction at which the road starts and at which it may connect to other roads.
- 'JTE_ID_END': Same as 'JTE_ID_BEG', but for the end of the road.
- 'WEGBHRSRT': The owner type of the road, used to distinguish between types of roads. Options here are 'G' (municipality), 'P' (province), 'R' (national government), 'W' (water board) and 'T' (other type of owner, e.g. private).
- 'WEGNUMMER': The number of a road used on signs and other road network descriptions, such as 'A12' or 'N219'. Only used for roads administered by the national government (generally highways) and provinces.
- 'RPE_CODE': Only used for multi-lane roads. This attribute describes the position of this lane relative to other lanes of the same road. For highways, this position is either left ('L') or right ('R') of the administrative direction (see 'ADMRICHTING'). For other roads, wind directions are used, with north ('N'), south ('Z'), west ('W') and east ('O').
- 'ADMRICHTING': Only used for highways. Administratively, highways have a direction, e.g. the Dutch A20 leads from the city of Gouda to Maassluis (east to west), even though drivers are able to drive in both directions. The physical direction is the direction used in the dataset between the start point ('JTE_ID_BEG') and end point ('JTE_ID_END'). If these directions match, 'H' is used – otherwise 'T' is written here.
- 'STT_NAAM': The name used to identify the road in the real world. This can be a street name or number, such as used in 'WEGNUMMER'.
- 'WPSNAAMNEN': The name of the town (not necessarily the same as the municipality) in which the road is located.

These attributes are used to calculate new attributes, or in the path calculation process itself.

7.2) Length

After having cleaned up, the first new attribute to be added is the 'LENGTH', attribute, detailing the length of a stretch of road. It is calculated using the ArcMap '*Calculate geometry*' tool, which calculates the total length of the road in metres. This length can then be used to calculate the time and fuel needed to traverse the road.

7.3) Direction

In the physical world, roads can sometimes only be traversed in a single direction, instead of both. In the dataset used, records sometimes depict an entire road with both of its directions (in most cases) or only a road lane, with one direction. For the path calculations, it is necessary to know in which directions roads can be traversed. This 'DIRECTION' attribute specifies that: a '0' represents a road that can be used both ways, '1' a road that can only be used following the physical direction (start junction to end junction) and '2' one that can only be used in the opposite direction (end junction to start junction).

The main attribute used to determine this is 'RPE_CODE': the position of a lane relative to other lanes of the same road. When this is not applicable (as the entire road is modelled as a single feature), '#' is used. In all such cases, the direction can be set to '0', i.e. traversable in both directions.

Determining the direction of one-way roads requires more variables. Unfortunately, roads' allowed directions do not always align with the physical (start to end junction) ones. In addition, no one-size-fits-all solution exists to determine whether this allowed direction is aligned with the physical direction or not. Two different approaches are needed here, due to the differences in coding of 'RPE_CODE': for highways left-right is used, whereas non-highways use wind directions. How exactly this is determined, is best illustrated using figures.

Figure 7.1 shows the logic for highways.

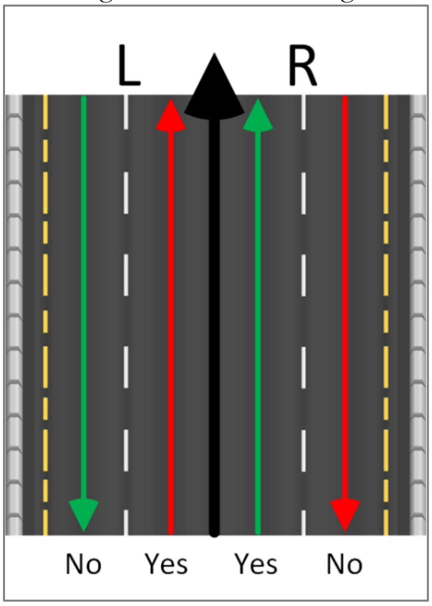


Figure 7.1: Direction logic for highways

In this figure, a road with four lanes is shown, each lane modelled as either a green or red arrow. In addition, the black arrow in the middle shows the administrative direction of the road, in this case from south to north. Two lanes are to the left of this administrative direction, with 'RPE_CODE' 'L', and two others to the right with 'R'. The coloured arrows point to a certain direction too, this being the physical direction of the road (start to end junction). The 'Yes' and 'No', finally, refers to whether or not these physical and administrative directions align.

When on the right, the allowed direction runs parallel to the administrative direction – in countries such as the Netherlands where right-hand traffic exists, that is. As such, if a road runs parallel to the administrative direction, 'DIRECTION' should be set to '1' (green arrows in the figure). If, however, it runs in the opposite direction, 'DIRECTION' should be set to '2' (red arrows). For lanes to the left, the exact opposite is true.

For non-highways, Figure 7.2 shows what logic is used to determine 'DIRECTION'.

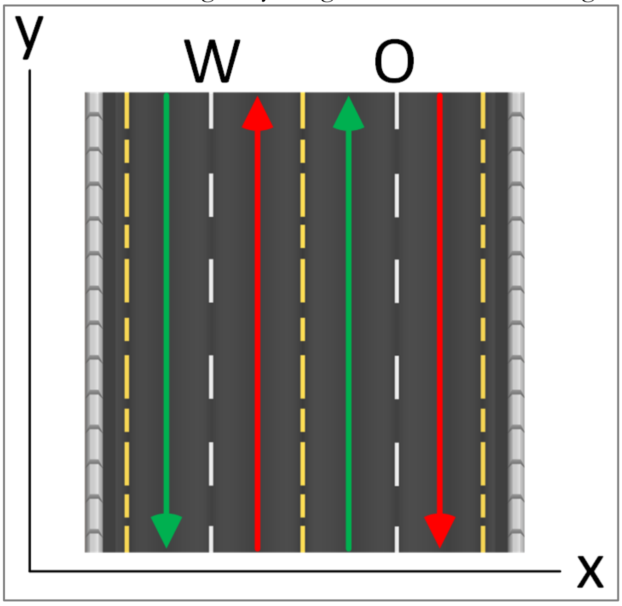


Figure 7.2: Direction logic for non-highways

Here, another road with four lanes – not classified as a highway – is shown, with the arrows showing the physical directions of these lanes. Now, no administrative direction exists. Instead, it is mentioned in 'RPE_CODE' whether a lane is to the west ('W'), east ('O'), north ('N') or south ('Z') of the entire road's centre. As such – as can be seen in the figure – a road with 'W' should run from north to south, and one with 'O' from south to north. Whether or not this is true, can only be checked using coordinates, as an attribute such as 'ADMRICHTING' is not available here. As such, '*Calculate geometry*' is used, this time to get the x and y coordinates of both the start and end junctions, stored respectively in 'JTE_BEGX', 'JTE_BEGY', 'JTE_ENDX' and 'JTE_ENDY'.

In case of a road running from north to south, the y-coordinate of its start junction should be higher than that of its end junction. If true, it means that the physical direction aligns with the real, allowed direction, in which case 'DIRECTION' can be set to '1' (a green arrow). Otherwise, 'DIRECTION' is to be set to '2' (a red arrow). The same logic can be used for all wind directions.

Finally, when drawing the dataset using the 'DIRECTION' attribute, it looks like the network in Figure 7.3. Here, green arrows depict an alignment between physical and allowed directions, red arrows a misalignment and black lines roads that are bidirectional.

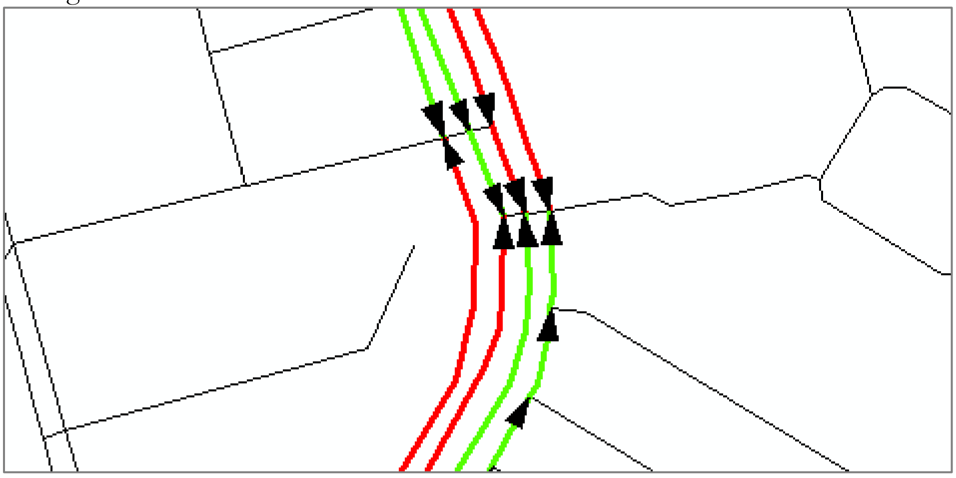


Figure 7.3: 'Wegvakken' symbolised using the 'DIRECTION' attribute

7.4) Speeds

For the different maximum speeds and cases in which dynamic speeds are or are not used, it needs to be known what the effective speed limit of a road would become. This is pre-calculated here, and stored in attributes with the naming logic 'xxx_DAY/NIGHT'. 'xxx' refers to the custom speed limit chosen, ranging from '050' to '130'. The choice for 'DAY' or 'NIGHT' depends on the dynamic speeds. Such speeds differ between night-time (usually 19:00-05:59, in a few cases 23:00-06:59) and day-time (other times). As such, the attribute with the effective speed limit for a road using 70km/h as a custom limit during the night is called '070_NIGHT'.

To get the normal, legal speed limits – i.e. when the custom limit is 130km/h, the highest limit available in the Netherlands – 'Wegvakken' is joined with the speeds dataset as described in Chapter 3.2.2, based on the unique identifier 'WVK_ID', called 'WEGVAK_ID' in the speeds dataset. This is where it first becomes apparent that a huge dataset is being used: joining the 974,686 features – the set of the entire country – with their respective speed limits takes around 42 hours.

After adding the speeds, it also becomes clear that the 'Wegvakken' dataset is not entirely correct. Visual inspection shows that some pedestrian pathways – e.g. in the Prinsenpark in Rotterdam – which do not allow for car traffic at all, are nonetheless labelled as 30km/h roads. This is most likely an error made by the owner of these pathways, in this case the municipality, as separate labels do actually exist for non-car roads. As no other attributes – including the ones deleted in the first stage of this chapter – can be used to identify which roads have mistakenly been labelled as such, it is not possible to filter such pathways from the dataset. Those that were correctly labelled as such were deleted from the set, though, as they would serve no purpose for an application focused on motorised vehicles.

Next, speeds for other custom limits than 130 are added. Not surprisingly, this turns 130km/h roads into 120km/h when choosing 120, and keeps 70km/h roads at 70km/h in that case. This is not true for lower speeds though, as it would not, for example, be safe to drive 50km/h on a highway. In such cases, the limit was set to '-1' to indicate that the road is not to be used with that custom limit. 80km/h is used as the minimum speed for highways, as is 60km/h for roads with legal limits of 80km/h. This leads to effective limits as in Figure 7.4.

		Legal Limit									
		130	**120**	**100**	**90**	**80**	**70**	**60**	**50**	**30**	**12**
Custom Limit	**130**	130	120	100	90	80	70	60	50	30	12
	120	120	120	100	90	80	70	60	50	30	12
	110	110	110	100	90	80	70	60	50	30	12
	100	100	100	100	90	80	70	60	50	30	12
	90	90	90	90	90	80	70	60	50	30	12
	80	80	80	80	80	80	70	60	50	30	12
	70	-1	-1	-1	-1	-1	70	60	50	30	12
	60	-1	-1	-1	-1	-1	70	60	50	30	12
	50	-1	-1	-1	-1	-1	-1	-1	50	30	12

Figure 7.4: Effective speed limits for combinations of legal and custom limits

The legal limits in this figure are the ones actually present in the dataset: this is why 12km/h is included (some small, private roads have this limit) while 110 and 40km/h are not (no road in the Netherlands has this limit).

Finally, the dynamic speed differences have to be taken into account. For the vast majority of roads, limits are set independent of the time of day. For stretches of eleven different highways, however, limits do differ between day- and night-time. This is true for parts of the A1, A2, A4, A6, A9, A12, A15, A28, A29, A58 and A270. As mentioned already in Chapter 3.2.4, no spatial dataset with these differences exists, a .pdf map and text file being the only information available on this topic. As the number of roads that this applies to is limited (eleven), it is possible to integrate this information manually. First, the respective stretches of road are selected manually. This process is aided by the 'WEGNUMMER' attribute, which allows for filtering out all roads not part of the highway needed. Stretches with lower limits – e.g. exit lanes – are also deselected based on the speed limits. This way, the different limits for day- and night-time can be set.

7.5) Traffic

As a choice was made not to use real-time traffic (see Chapter 3.2.3), traffic conditions are simulated randomly. As it would not be realistic for two spatially close roads to have totally different traffic values, though – traffic usually spreads out – this value cannot be filled totally randomly: a spatial relation between values of different roads should exist. As such, bottleneck points are used. These points are locations where some event – e.g. an accident or construction works – took place, leading to strong disturbances to road traffic. Whereas roads close to these points will suffer most, conditions gradually get better when moving away from these locations, which is simulated by decreasing traffic intensities as distances to bottleneck points increase. A maximum of 10km, after which no effect exists anymore, is used.

To achieve this, a minimum bounding box of the Dutch road network is first created using the '*Minimum bounding geometry*' tool on 'Wegvakken'. After doing so, 100 bottleneck points are created at random locations using the '*Create random points*' tool. Then, the '*Near*' one calculates the distance between each road and the closest bottleneck. This is when the final 'TRAFFIC' attribute can be created and populated: for each road where the distance is larger than 10km traffic is set to zero, whereas other roads follow the formula '*TRAFFIC = 1 – 0.0001 * NEAR_DIST*', with 'NEAR_DIST' containing the distance to the nearest bottleneck. Using this formula, every road receives a 'TRAFFIC' value ranging from 0 (no traffic) to 1 (severe traffic). The resulting road network symbolised for traffic intensity for the entire Netherlands (which was later reduced to Zuid-Holland) is shown in Figure 7.5.

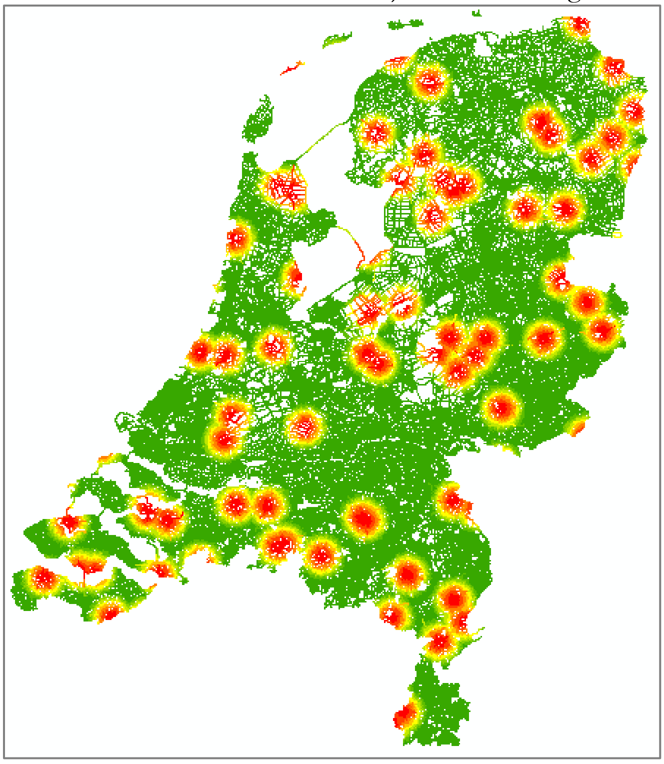


Figure 7.5: 'Wegvakken' symbolised for traffic intensity

7.6) Grade

The envisioned 'GRADE' attribute is where the biggest change had to be made on operational grounds. It was in fact not possible to create this attribute in the end, even though no problems were originally envisioned here. The plan, as described in Chapter 6.2, was to calculate roads' grades by using the differences in heights between their start and end junctions. These heights were to be based off the 'AHN2' dataset (see Chapter 6.1.1).

In order to do this, a Python script was written that, for every junction, would get its coordinates and use these to get its height with the '*Get cell value*' tool. While the script works as intended, calculation times are so high that this attribute had to be abandoned. With approximately 1.5 second of calculation time for every junction, roughly 35 days would be needed to do this for the entire 'Wegvakken' dataset. This is due to the huge size of the height map: around 1TB. Reducing this size was tried using a '*Clip*' operation, but this turned out to be unfeasible due to the dataset size too: after two days of clipping, this operation was stopped too, after which an unknown portion had been processed. Using a lower-resolution height map was deemed too inaccurate, as roads are usually modelled as short stretches.

Though for larger, end-user projects such times could be used – and probably be lower as more calculation capacity would be available – it would simply not be feasible to wait multiple weeks for this calculation to finish, only to add a single attribute and factor. As such, it was decided to drop it.

Though far from ideal, omitting grades from the final algorithm is not disastrous in this specific case. The same arguments already presented in Chapter 6.2 when discussing the precision of this attribute, can be used here. The most important of these is the point that the influence of grades would be very limited for this study anyway. This is due to the characteristics of the Netherlands, a country that is generally flat, save for some smaller areas. As such, the actual significance of grades in fuel consumption is very low for this country, making it acceptable – in this case – to omit them.

7.7) Database Import

Having added all attributes (except for 'GRADE', as mentioned above) to 'Wegvakken', the road network is ready to be imported into the PostgreSQL database for use with the application. For this operation, PostgreSQL conveniently provides the 'PostGIS 2.0 Shapefile and DBF Loader Exporter'. This tool is able to import shapefiles into PostgreSQL databases and export spatial databases to shapefiles. Here, the import function is used (PostGIS, 2017).

This leads to a table (renamed to 'roads') with the original attributes of the 'Wegvakken' shapefile, in addition to an automatically created 'gid' attribute with a unique identifier and a 'geom' one, detailing the geometry of all records. 'gid' is dropped from the table, as 'WVK_ID' already serves as a unique identifier for this dataset. After doing so, 'WVK_ID' is set to be the primary key. No other changes have to be made to this table. The final table with its attributes (not including all speed attributes) is shown in Figure 7.6.

☐	wvk_id [PK] inte...	jte_id_b... double p...	jte_id_e... double p...	stt_naam characte...	wpsnaa... characte...	length double p...	direction integer	traffic double p...	geom geometry	050_day integer	060_day integer	070_day integer	080_day integer
☐	27142004	27142029	27142037	Rondweg	SLUIS	146.042	0	0	0102000...	50	60	60	60
☐	27142010	27142002	27142003	Marktplein	SLUIS	16.9372	0	0	0102000...	30	30	30	30
☐	27142014	27142008	28142009	Grevenin...	SLUIS	95.3008	0	0	0102000...	30	30	30	30
☐	27142015	27142003	28142006	Jonkvrou...	SLUIS	335.362	0	0	0102000...	30	30	30	30
☐	27142016	27142013	28142010	St. Anna...	SLUIS	38.3133	0	0	0102000...	50	50	50	50
☐	27142019	27142002	27142144	Marktplein	SLUIS	57.0423	0	0	0102000...	30	30	30	30
☐	27142021	27142013	27142144	Haven	SLUIS	66.0339	0	0	0102000...	30	30	30	30
☐	27142024	27142037	27142100	Rondweg	SLUIS	35.0468	0	0	0102000...	50	60	60	60
☐	27142026	27142139	31139081	Rondweg	SLUIS	2643.74	0	0	0102000...	-1	60	70	80
☐	27142027	27142100	27142104	St. Anna...	SLUIS	23.8685	0	0	0102000...	-1	60	70	70
☐	27142029	27142080	27142104	Rondweg	SLUIS	185.681	2	0	0102000...	-1	60	70	80
☐	27142030	27142104	27142139	Rondweg	SLUIS	216.315	2	0	0102000...	-1	60	70	80

Figure 7.6: 'roads' table in PostgreSQL

Another table, 'results', is created from scratch afterwards to store the calculation results for the developer mode (see Chapter 6.3.9). Attributes are added to store the parameters used in the path calculations – e.g. traffic or no traffic – and the actual results for time and fuel. Figure 7.7 shows this table with random values for records.

☐	run_id [PK] inte...	start_id integer	end_id integer	carmodel integer	traffic boolean	dynspee... boolean	maxspeed integer	usedtime double p...	usedfuel double p...
☐	1	94098	90433	2	false	true	120	147.5	5.4
☐	2	94098	90433	1	false	true	120	141.6	5.3
☐	3	94098	90433	1	false	true	120	138.6	5
☐	4	94098	90433	6	false	true	120	138.1	5.5

Figure 7.7: 'results' table in PostgreSQL

8) Website Creation

Just as with the database creation process of the previous chapter, application development is bound not to proceed exactly as planned, be it with larger or smaller changes. As such, it is again important to keep track of such changes, as to increase this study's reproducibility. As with the previous chapter, this chapter can be considered as a technical guide and a process report, as well as a presentation of the resulting website.

Each subsection in this chapter represents one of the components as shown in the blueprint of Figure 6.5. As these have already been created at this point, the 'Road Network' and 'Calculation Results' components are not discussed again. The order in which the remaining components are discussed here, is the order in which they are – approximately, as some components are expanded later – developed.

8.1) User Settings

A relatively simple component, the user settings are developed exactly as envisioned in the plan of Chapter 6.3. The final result is a bar at the top of the website, showing three main elements: a mode selector button, specific options for both the user and developer modes – in the user case options for the car model, traffic and dynamic speeds and in the developer case options for the number of runs and overwriting old results – and a 'Go' button. Figure 8.1 shows the bar for the user mode, whereas Figure 8.2 shows it for the developer one.

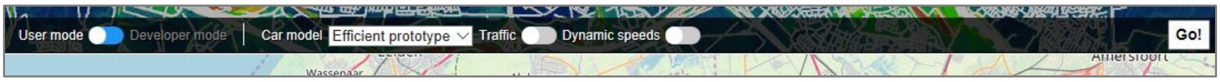


Figure 8.1: The user settings for the user mode

Figure 8.2: The user settings for the developer mode

8.2) ArcGIS API

As with the user settings, the ArcGIS API is implemented exactly as planned. As can be seen in Figure 8.3, from a layout point of view, it occupies most of the website.

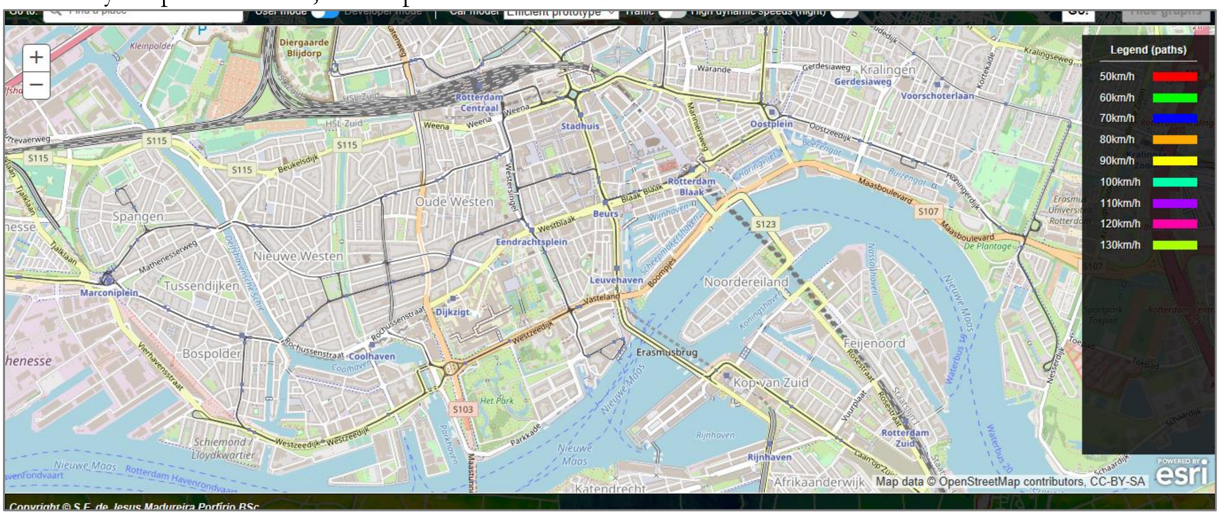


Figure 8.3: The ArcGIS API on the website

As can be seen in this figure, a legend is added to aid in recognising the different paths, which each represent a path for a certain custom speed limit in the user mode. This way, users are able to see how different limits do – and sometimes do not – lead to different paths for the same pair of start and end points. As paths do often at least share some sections, they are drawn with different line thicknesses, as to preserve the possibility of distinguishing between them, even when overlapping. The start and end point can be selected directly within the API by right-clicking the map at the desired locations. An example of paths drawn in the API can be seen in Figure 8.4.

Figure 8.4: An example of different paths for several speed limits

8.3) Front-end Controller

All in all, no significant changes are applied to the front-end of the web application. One of the few later additions is the 'Shown/Hide graphs' button in the top-right corner. This button is discussed further in Chapter 8.7 and can be seen in Figure 8.5, which shows the entire website extent.

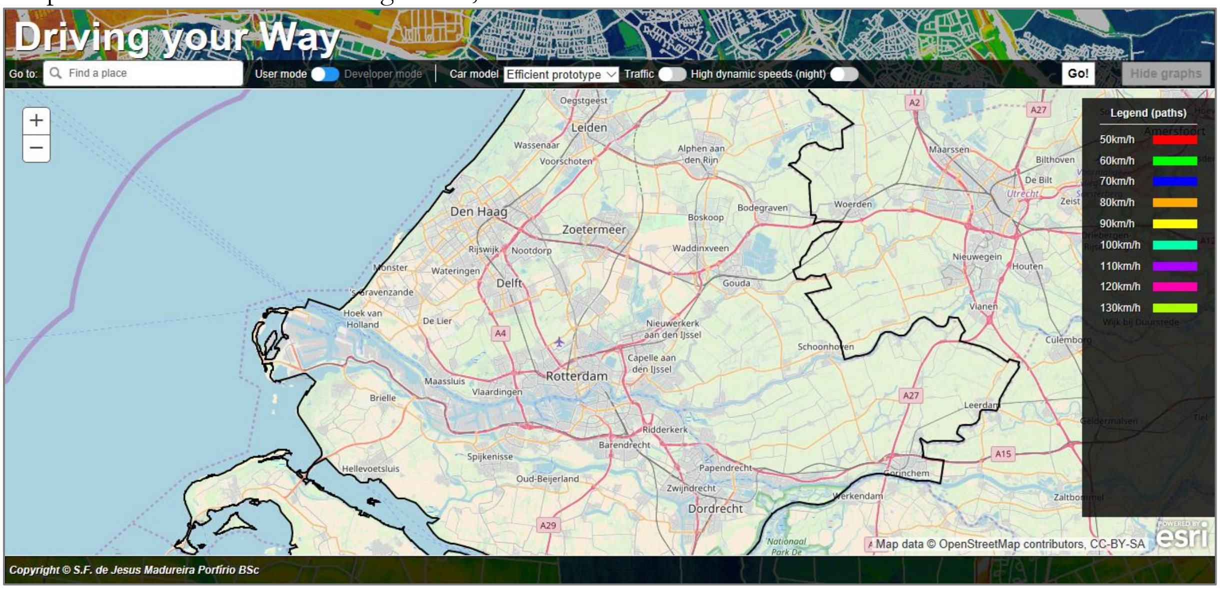


Figure 8.5: The full extent of the website

In this figure, a border, drawn in black, can be seen on the map. This is the other addition that has to be made, as the study area had to be limited to Zuid-Holland. In order to facilitate the user in recognising which locations are still within this area – and can thus be used for paths – the borders of this province are drawn on top of the map.

8.4) Back-end Controller

As opposed to the front-end, the back-end has seen more changes when compared to the original plan. By far the greatest changes are made to the algorithm design, which underwent a total overhaul. The same is true for the random points generator, which is greatly simplified.

While these changes are discussed in their respective sections, one additional change should be mentioned right away. This relates to the coordinate system in which data is stored, transferred and shown.

As the network data used was produced by Dutch authorities, the system used here is Amersfoort / RD_New, or SRID 28992. As such, any paths retrieved by this application would use paths with coordinates in this system. However, as the ArcGIS API uses WGS84, or SRID 4326, this data cannot be used directly. Though the API does offer functionality for projecting data in other systems on the fly, the choice is made to convert the original data to WGS84. As such, the necessity for making on the fly projections every single time is removed. This permanent conversion is done within the database, as PostGIS offers a function, 'ST_Transform', to this end (Kadaster, 2018; PostGIS, 2018).

8.5) Routing Algorithm

As mentioned in the previous section, the algorithm design underwent a total overhaul as compared to the original plan, which was to write a custom algorithm within the back-end in PHP. The original plan was in fact executed, leading to a working algorithm. It soon became clear, however, that it had to be abandoned, due to calculation time issues. Testing of this algorithm shows that, though it does produce correct results, calculation times grow strongly with the distances involved, with one kilometre of distance extra between the start and end points leading to approximately one minute extra in calculation. For every second of calculation time for a single path, the calculation time for the entire collection of required paths amounts to one day. As such, even one minute per path would already constitute two months (60 days) of calculation time – time which is not available within this study.

Most time is found to be wasted on communication between the web server and the database server, as with the custom algorithm – which is executed on the web server and only requests data from the database – for every step SQL requests have to be made. As such, the solution for this is to move as much of the algorithm to the database server, as to minimise time lost in communication. This lead to adopting the use of 'pgRouting', an extension to PostGIS offering functions for path-finding. Though not as flexible or customisable as a self-built algorithm function, it has the advantage of being executed entirely within the database server, cutting calculation time to about 15 to 35 seconds per path when using the entire Netherlands, independent of its length. The inflexible nature of this extension does not prove to be a problem in this case as the algorithm needed is Dijkstra without any customisation – which pgRouting offers (pgRouting, 2018).

As pgRouting functions do require input data to be in a specific format, though, the data structure of the database has to be altered. Whereas in the original plan, different speed limits are included as different attributes, pgRouting does not offer the possibility to alternate between such attributes: it needs to be fixed. The – somewhat redundant – solution to this is to split the network data into different tables, in which each table represents a certain speed limit. Though far from elegant, this does make it possible to use pgRouting.

Even though this use cuts calculation times, they were still too high: in order to meet the deadlines as set for this study, the time had to be reduced to a maximum of ten seconds per path. Testing shows that most time is spent on requesting records from the road network dataset. The network of the entire country consists of almost two million (1,860,419) records, which would need to be searched continuously. In order to cut time to acceptable rates (< 10 seconds per path), it was decided to reduce the size of the dataset by limiting the study area to the province of Zuid-Holland (303,698 records), instead of using the entire Netherlands. Of course, this is only a necessity for this study: using more powerful equipment, the entire country, or any other area for which data is available, could be used. The choice for this province is made as it still has a relatively dense network (even though there are twelve provinces, it holds one-fifth of the roads) and since it has both urban (dense) and rural (non-dense) areas. Zuid-Holland is, in addition, the province with the highest share of highways with dynamic speeds, making it easier to test for the influence of such speeds. It is also a province without any notable height differences (see Figure 6.3), which further minimises errors that could arise due to not using heights.

By doing all of this, the final number of runs for the statistical analysis is even higher than initially set: instead of 500, 750 runs are done, leading to more reliable results (see Chapter 10).

8.6) Random Points Generator

Originally, this element was to choose random points within the study area based on its geographical extent. The downside of this, is that points could fall within water areas or neighbouring countries, which could be corrected by choosing the closest point afterwards. This would, however, lead to a disproportionate number of chosen points being on the edge of the area, which could create a bias in the results.

The solution is to not randomise points based on their geographical location, but on their id in the database. To this end, a separate table, in which every junction appears once and only once – as to create equal chances of records being picked – is created. By randomly choosing two (different) junctions from this table, entirely unbiased pairs of points can be generated.

8.7) Graph Plugin

The graphical plugin, Chart.JS, is the last one to be integrated into the front-end of the application. On the successful calculation of paths in the user mode, a window is opened, showing two graphs: one for time consumption with different custom speed limits, and one for fuel. To allow for switching between the graphs and the actual paths on the map, a 'Show/Hide graphs' button is added, which allows for hiding the graphs window when desired.

Figure 8.6 shows an example of graphs for time and fuel for the several paths that exist between a chosen pair of points.

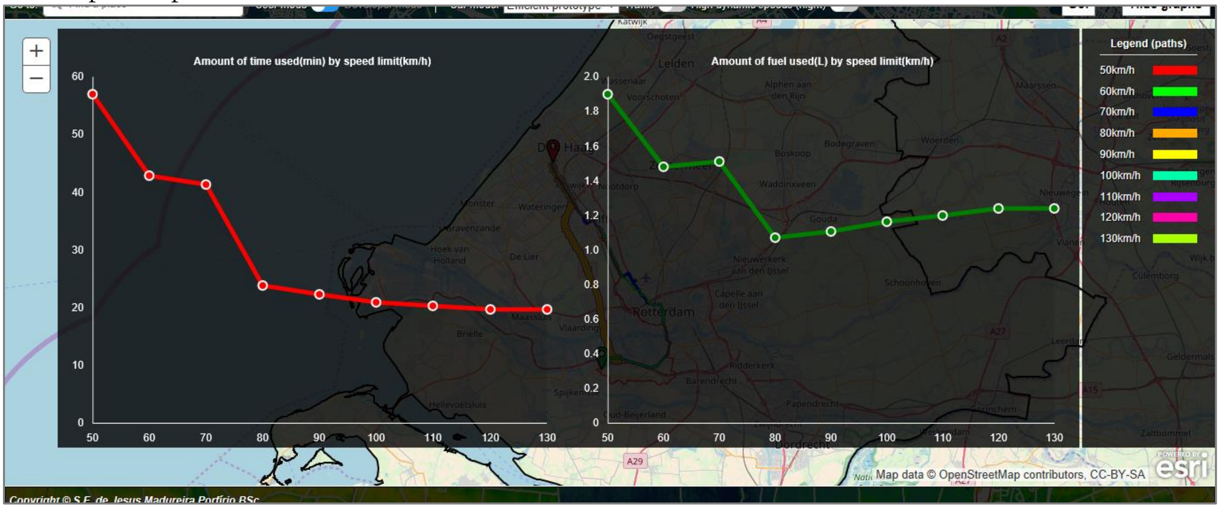


Figure 8.6: An example of graphs showing time and fuel consumption for different speed limits

9) Use Case

In the chapters of the practical research, the results generated using the developer mode are analysed: the user mode is not used there. This mode is – as its name suggests – only meant to be used by end users, and does not generate any results for analysis. As the application would however in the end be built for end users, it is worthwhile to include an example case of how it could be used by such users.

This case is based upon a fictional user who, based on his situation, has certain priorities and makes decisions based on these priorities. Following his requirements and plausible thinking process, this serves as a demonstration of how the application could be used in a very specific case.

The 'user' here is a man called Peter. Peter is a civil engineer and lives in the city of Leiden. Today, he needs to visit a fair in Rotterdam Ahoy, a convention centre in the city of Rotterdam. There, he will have some meetings with potential business partners at 11:30 in the morning. Peter wants to go by car, as he will need to drive to some other locations afterwards that are not easily reachable by public transport. He would be ready to depart at 10:30, meaning that he has one hour to get to his destination.

Peter decides to use the 'Driving your Way' website. When opening it, the site detects his location and uses this as the default point of departure, which in this case is correct:

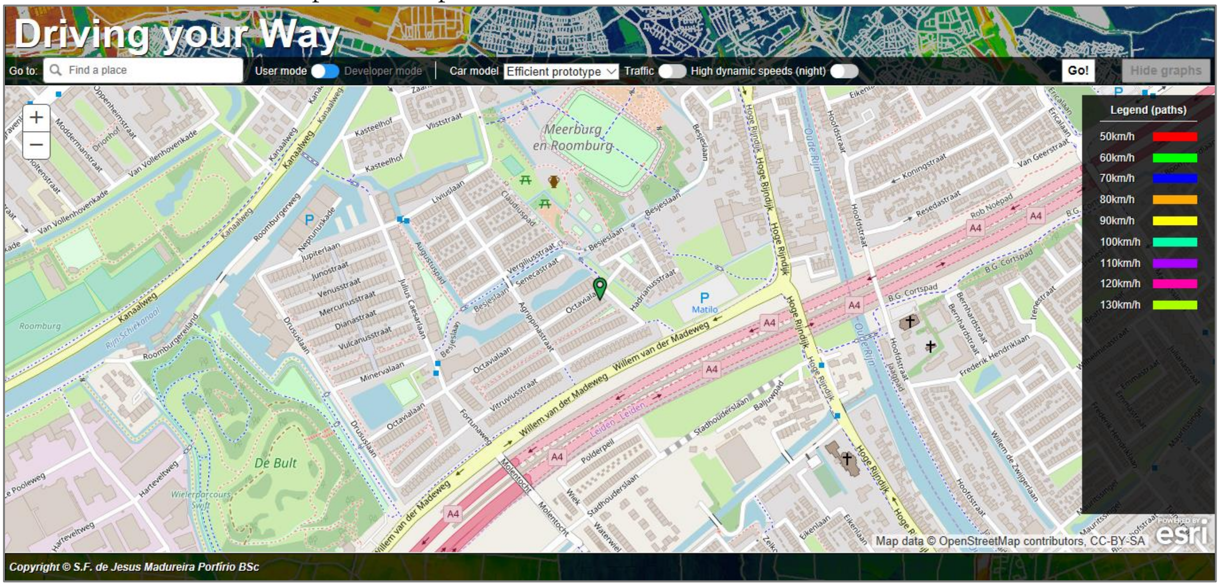


Figure 9.1: The first thing Peter sees

Next, the destination is to be chosen. As Peter does not know exactly where Rotterdam Ahoy is located on the map, he types this name in the 'Find a place' toolbar to navigate there. The map automatically centres at the right spot, after which Peter right-clicks one of the parking lots next to the convention centre and selects 'End here', placing the destination marker there:

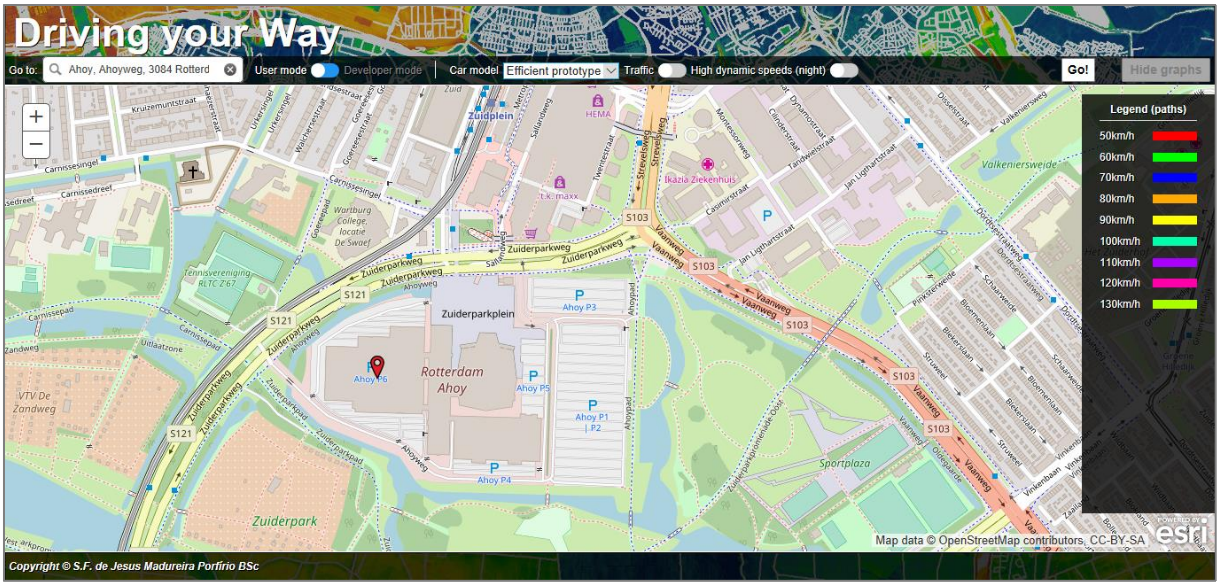


Figure 9.2: The selection of the destination

Having both the point of departure and destination set, some settings can be configured differently in order to make the returned results more precise and valid. The first thing to do is to select a car model. Since Peter drives a luxury SUV, he selects this option. He wants current traffic to be taken into account as he will be leaving soon, and does not select night-time dynamic speeds as he will be travelling during the day. All that is left to do, is to press the 'Go' button:

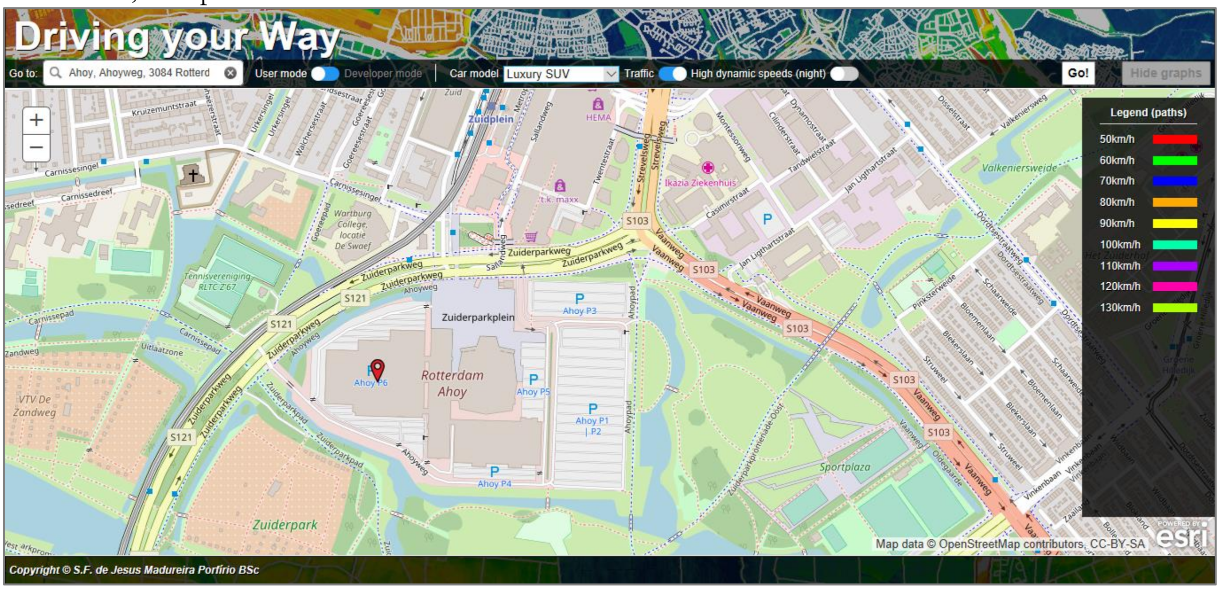


Figure 9.3: Configuring the settings

After a short while, Peter is presented with the results of the calculations. The first thing he sees are two graphs, both showing consumption – the left one for time and the right one for fuel:

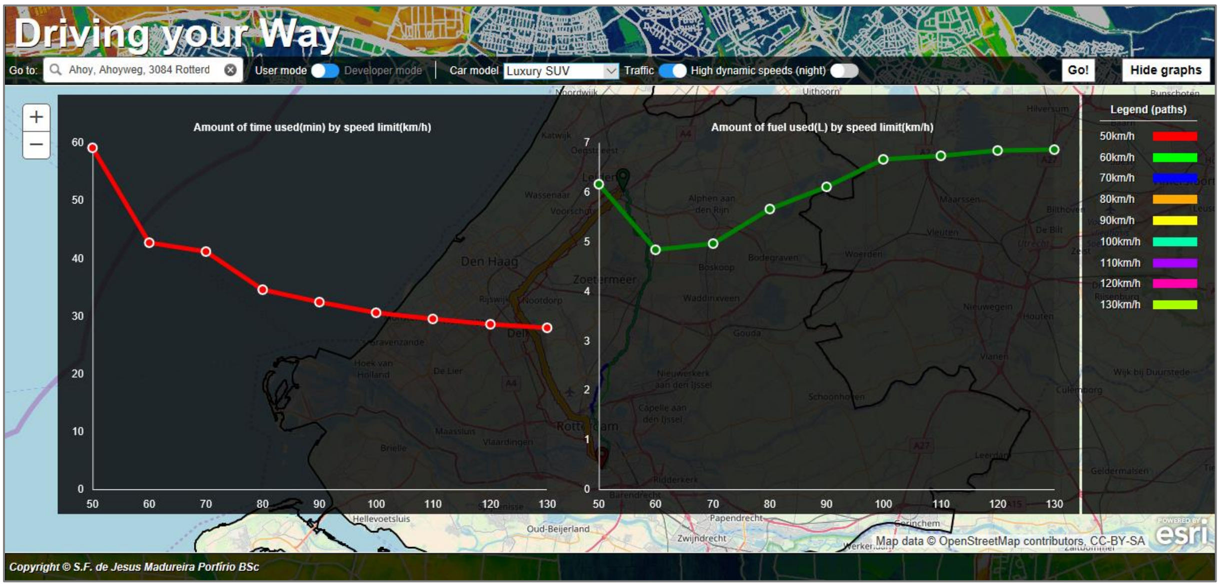


Figure 9.4: The time and fuel consumption for the different speed options

Peter decides to take a look at the actual paths. He clicks the 'Hide graphs' button, after which the graphs are hidden from sight and the paths become better visible. The options turn out to be quite different: for higher speeds, paths stick to highways for the larger part of the trip. For 60 and 70km/h, however, provincial roads are used mainly, as the use of highways is not allowed at such speeds. For 50km/h, finally, provincial roads are out of the question too. This path actually resembles the ones for the highest speeds, using roads close to the highways, but never the highways themselves:

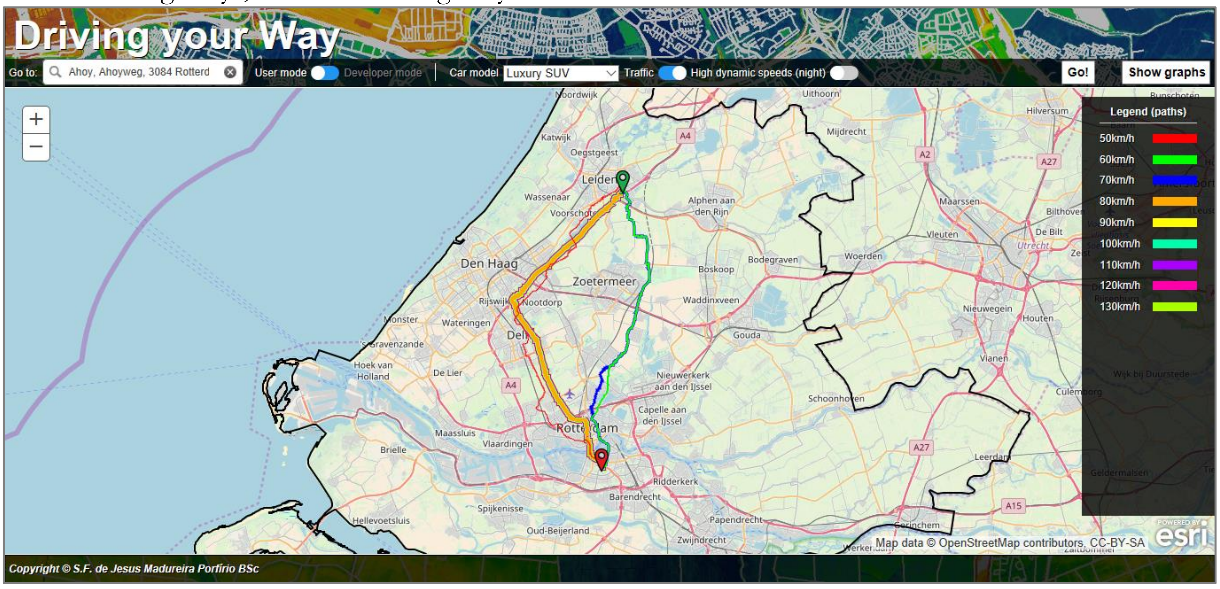


Figure 9.5: The paths for the different speed options

Peter does not really hold a preference for any type of road, so he decides to base his decision on the time and fuel needed to get to Rotterdam. He clicks the 'Show graphs' button, after which the graphs are shown again, as in Figure 9.4. The 50km/h option is quickly discarded as it takes 59 minutes, which would probably cause him to be late for his meeting, and since it uses more fuel than higher speeds anyway. The difference between 60 and 70km/h is marginal: both take about 42 minutes and 4.9 litres. 80km/h, then, leads to 34 minutes and 5.6 litres used. Since higher speeds do only marginally reduce time – 130km/h would take 28 minutes – but do lead to higher fuel consumption, those are discarded too. For Peter, the choice lies between 70 and 80km/h, meaning a difference of 8 minutes and 0.7 litres. He realises that he would not mind being able to do a short round of the fair before his meetings, and decides that the time saved by driving 80km/h is worth the extra fuel used. As such, Peter settles for this option.

Part IV) Practical Research

10) Results Analysis

After having been built, the application ran for five days, in order to create the results that are analysed in this chapter. Though the goal for the number of runs in this Monte Carlo analysis was initially set at 500, the lower than expected calculation times allowed for raising this number to 750. As Driels & Shin (2004) mentioned that the optimal number of runs for Monte Carlo analyses typically lies between 500 and 1000, it should lead to results that are, statistically, as valid as possible.

The first section of this chapter, 10.1, analyses the overall influence of custom maximum speeds on time and fuel consumption. The second one, 10.2, in turn looks at the influence of car models, traffic and dynamic speeds.

10.1) Custom Maximum Speeds

The fourth sub-question of the current study is *"How much fuel and time can be saved or lost by driving at certain maximum speeds in practice?"* and is answered in this section. The – possible – influences upon time and fuel are discussed separately. In the third part of this section, the more practical influence upon concrete savings is looked at.

10.1.1) Custom Speed Influences on Time

First, it is useful to look at the different means for time (in seconds) of paths between the different speed limits. These means can be found in Figure 10.1:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Mean Time	2452	2019	1948	1604	1530	1466	1437	1411	1397

Figure 10.1: Mean values of time per speed limit

It is important to note though, that these values have no absolute, but only relative meanings, as the absolute values depend on the size of the study area. As such, it is only relevant to look at the differences between limits. These can be seen easier by plotting these values, as is done in Figure 10.2:

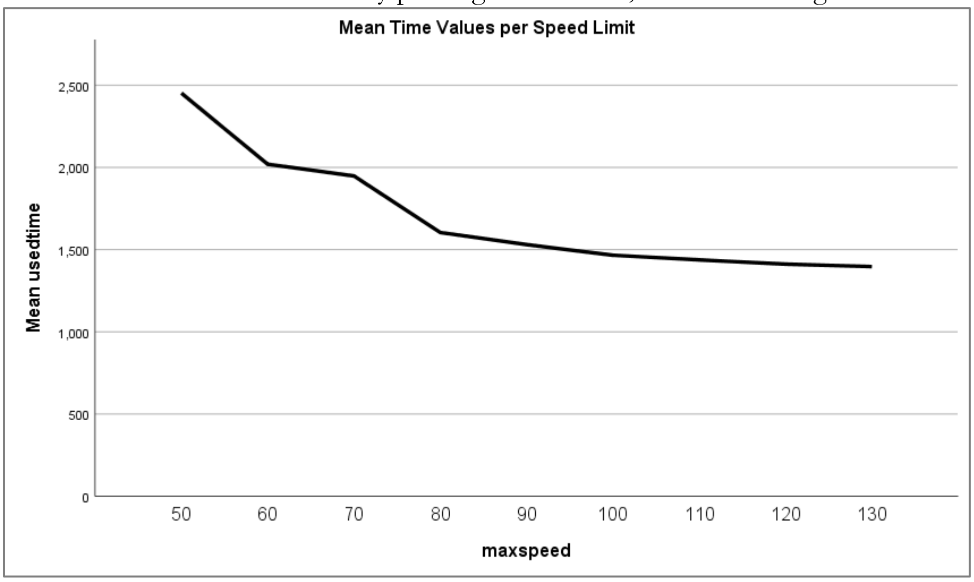


Figure 10.2: Mean values of time plotted per speed limit

In this graph, it can be seen that the decline in time needed is rapid in the beginning, and slows down at higher speeds. In any case, it is not linear – confirmed by a skewness of 0.997, indicating skewness at the left side – which could have been expected following the logic that going x% faster should make the trip x% shorter. This would be true, were it not for the fact that at lower speeds, not all roads can be used and thus detours need to be made. This effect of 'locked roads' can be spotted well in the graph, as the decline between speeds with different ranges of available roads (50 and 60, as well as 70 and 80) is the steepest.

The relative time reductions, as compared to time spent when adhering to 50km/h, are found in Figure 10.3. It confirms that reductions are highest in the beginning, and reveals that driving at 130km/h almost halves time required as compared to 50km/h, with just 56.7% of time needed.

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
% Time Spent	100.0	82.4	79.4	65.4	62.4	59.8	58.6	57.6	56.7

Figure 10.3: Mean time needed per speed limit, compared to 50km/h

Using ANOVA, with a statistical significance of < 0.001, the hypothesis that these differences in means are due to chance, can be rejected. With an Eta of 0.369, a moderately strong relation between maximum speeds and used time exists. The Eta Squared of 0.136 shows that 13.6% of variance in time can be explained by differences in maximum speed.

As such, for the whole dataset, there are significant differences in time averages, which are influenced moderately by maximum speeds.

10.1.2) Custom Speed Influences on Fuel

Now that the influence of speed on time has been proven, the same process can be used to determine whether or not the same holds for fuel. Figure 10.4 shows the mean values for fuel usage (in litres) per speed limit:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Mean Fuel	4.12	3.65	3.73	3.91	4.15	4.43	4.61	4.82	4.98

Figure 10.4: Mean values of fuel per speed limit

Again, only relative changes are relevant here. These can be visualised easiest by plotting them in the line graph of Figure 10.5:

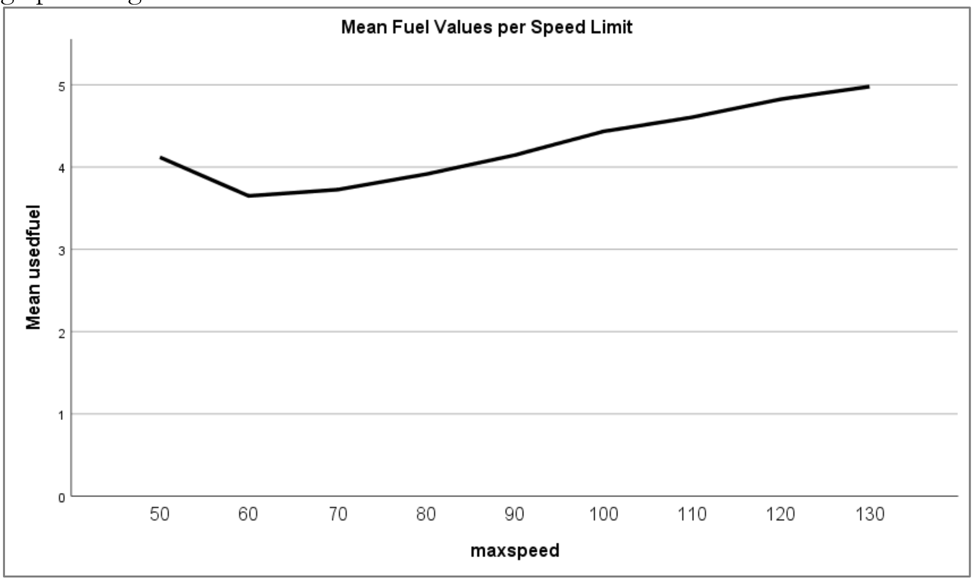


Figure 10.5: Mean values of fuel plotted per speed limit

Here, it can be seen that the mean fuel usage declines in the beginning, but starts rising again after 60km/h. The value of 50km/h is surpassed again at 90km/h, after which usage rises fairly linearly. As with time, the skewness value of 1.660 indicates that fuel usage is strongly skewed to the left side. In fact, this graph resembles the fuel efficiency graphs of Chapter 5, in that 50km/h uses relatively much fuel, the next speeds less and higher speeds more again.

These changes of fuel usage in percentages, compared to 50km/h, are presented in Figure 10.6.

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
% Fuel Spent	100.0	88.6	90.5	94.9	100.7	107.5	111.9	117.0	120.9

Figure 10.6: Mean fuel needed per speed limit, compared to 50km/h

Seeing as the lowest value (at 60km/h) is only 36.4% lower than the highest value (at 130km/h), the observed differences seem to be smaller than the ones in time.

Using ANOVA, with a statistical significance of < 0.001, the hypothesis that these differences in means are due to chance, can be rejected. With an Eta of 0.126, a weak relation between maximum speeds and used fuel exists. The Eta Squared of 0.016 shows that only 1.6% of variance in fuel can be explained by differences in maximum speed.

As such, for the whole dataset, there are significant differences in fuel averages, which are influenced weakly by maximum speeds.

10.1.3) Custom Speed Influences on Time/Fuel Savings

Finally, it is interesting to not only look at time and fuel separately, but to compare both in different maximum speed situations as well. By doing so, more insight into actual savings achieved by choosing to adhere to different speeds can be gathered.

Figure 10.7 shows the average time (in minutes) and fuel (in litres) consumption used at varying speeds, plus the ratio of litres per minute used:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Mean Time	40.900	33.700	32.500	26.700	25.500	24.400	24.000	23.500	23.300
Mean Fuel	4.120	3.650	3.730	3.910	4.150	4.430	4.610	4.820	4.980
Litres/Minute	0.101	0.108	0.115	0.146	0.163	0.182	0.192	0.205	0.214

Figure 10.7: Mean values of time and fuel and their ratios per speed limit

From this table, it could be concluded that the lower the speed, the higher the efficiency, as litres/minute. This can however be deceiving, at it is a mere indication of the efficiency per minute driven. When wanting to go for the most efficient speed, this ratio would suggest that going for 50km/h is best. However, when looking at the time and fuel consumptions for this speed, it can be seen that both are actually higher than for 60, 70 and 80km/h.

Instead, it is much more insightful to look at what savings – or extra expenses – changes in speed cause. To that end, Figure 10.8 compares the changes in time and fuel when compared to the previous speed, e.g. how much is used more or less at 80km/h as compared to 70km/h:

Speed Limit	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Δ Time	-7.20	-1.20	-5.80	-1.20	-1.10	-0.40	-0.50	-0.20
Δ Fuel	-0.47	0.08	0.18	0.24	0.28	0.18	0.21	0.16
Δ Fuel/Δ Time	-0.07	0.07	0.03	0.20	0.25	0.45	0.42	0.80

Figure 10.8: Differences in time and fuel spent compared to 10km/h slower

The third row with data in this table, is the one that provides actual practical insight into the numbers: it shows how much litres of fuel extra are needed to save one minute of time. When accelerating to 60km/h from 50km/h, fuel is actually saved. Going from there, though, speed increases lead to more fuel being consumed. These marginal costs of increasing speed increase at higher speeds: whereas the marginal costs of going from 60 to 70km/h are 0.07 per minute, that amount is 0.8 when accelerating from 120 to 130km/h. This shows that, although increasing speed does continue to cut time usage, fuel usage rises at such a higher rate that the marginal costs of doing so keep increasing for every step.

10.2) Car Models, Traffic & Dynamic Speeds

The fifth sub-question of the current study is *"To what extent do these time and fuel differences for different speeds hold when traffic, dynamic maximum speeds and car models are introduced?"* and is answered in this section. As was established in the previous section, different speeds do indeed lead to significantly different time and fuel usages. As such, it makes sense to analyse whether or not these significances hold when introducing car models, traffic and dynamic speeds. These factors will be discussed separately in the following sections.

10.2.1) Car Model Influences on Fuel

For car models, only the influence on fuel is analysed. After all: car models are only used in the formulae for fuel, and not for time, making any influence on the latter impossible – at least in this model. Car models should however – theoretically, at least, according to Chapter 5 – influence fuel. To see whether or not this holds in practice, Figure 10.9 shows the mean fuel usage per speed limit for the different car models:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Efficient Prototype	1.43	1.23	1.24	1.21	1.25	1.30	1.33	1.38	1.41
Large Old Van	7.18	6.49	6.71	7.36	7.89	8.54	8.92	9.39	9.71
Luxury SUV	4.29	3.82	3.92	4.22	4.52	4.89	5.10	5.38	5.57
Sportscar	4.89	4.14	4.08	3.75	3.76	3.80	3.85	3.92	3.99
Average Sedan	2.80	2.56	2.67	3.04	3.31	3.64	3.82	4.06	4.22

Figure 10.9: Mean values of fuel per speed limit for car models

The same data is visualised as a graph in Figure 10.10:

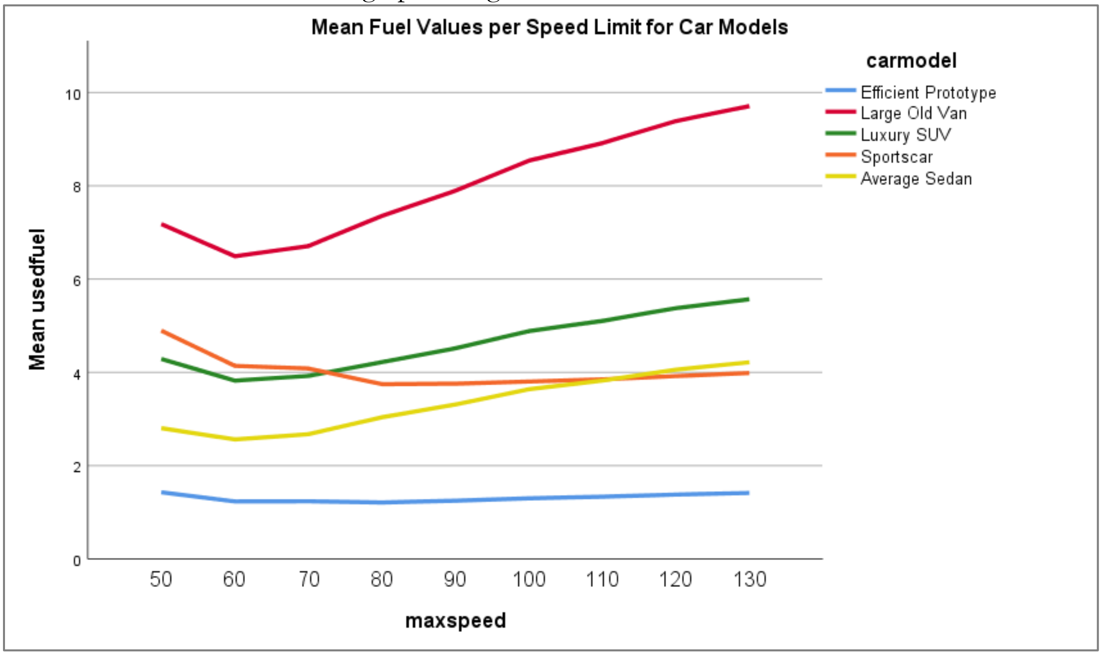


Figure 10.10: Mean values of fuel plotted per speed limit for car models

These figures do seem to suggest strong differences in fuel consumption between car models. At 130km/h, the consumption of the large old van is almost seven times higher than that of the efficient prototype, for example.

This is confirmed by the ANOVA, which shows that for every speed limit the significance is extremely high, being < 0.001 in every case. Even more importantly, the Eta of 0.619 shows that a strong relation between car models and fuel consumption exists. The Eta Squared of 0.383 in turn, shows that 38.3% of variance can be explained by variations in car models.

As such, it can be concluded that car models do have a strong influence on fuel consumption.

In addition, it is checked whether or not for each car model separately, speed limits still have a significant influence on the fuel consumption, as it has for all models taken together. Though, judging from Figure 10.10, some models do seem to sport more variance than others, with significances at < 0.001 for every model, differences are still significant. When checking the Etas for the relation between maximum speeds and fuel consumption by car model, it is seen that the relatively small effect of maximum speeds on fuel found in Chapter 10.1.2 is not representative for all car models. Whereas for the efficient prototype and the sportscar the effect sizes are, respectively, 0.109 and 0.151 and indeed rather small, for the other models the sizes lie between 0.2 and 0.3. This already indicates a stronger, though still small, effect of maximum speeds upon fuel consumption.

10.2.2) Traffic Influences on Time

As opposed to car models, it is checked if traffic has an effect upon time usage in paths. Here, the cases with and without traffic are compared. Figure 10.11 shows the mean time usage for both cases:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
With Traffic	2599	2143	2068	1716	1638	1569	1540	1513	1499
Without Traffic	2305	1895	1828	1492	1423	1362	1335	1310	1295

Figure 10.11: Mean values of time per speed limit with and without traffic

This suggests time usage to be higher with traffic, which would be a logical result. The difference is around 13% at lower speeds, increasing to 15% at higher ones, indicating that, on average, traffic adds about 14% to time consumption.

Plotting these results in a graph, Figure 10.12 suggests the same:

Figure 10.12: Mean values of time plotted per speed limit with and without traffic

Here, it can be seen clearly that paths without traffic systematically take less time, as was established to be the case earlier. Using ANOVA, the differences are shown to be significant (< 0.001). Eta was used to check the effect size, which was found to be 0.122, indicating a weak effect. The explained variance, as revealed by Eta Squared, is 1.5% here.

It can as such be concluded that traffic has a weak influence on time consumption.

When checking whether the influence of speed limits upon time still holds when differentiating for traffic, this is found to be true. Both with and without traffic, this influence is significant (< 0.001). Eta is used to check whether or not the effect size of maximum speeds varies between the traffic situations. The difference is found to be small, with an Eta of 0.391 without traffic and 0.358 with traffic.

10.2.3) Traffic Influences on Fuel

Next, the effect of traffic on fuel is checked for. Figure 10.13 shows the mean fuel consumption per speed limit, for the cases with and without traffic:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
With Traffic	4.49	3.95	4.01	4.15	4.38	4.63	4.79	4.96	5.13
Without Traffic	3.75	3.35	3.44	3.67	3.92	4.24	4.43	4.70	4.83

Figure 10.13: Mean values of fuel per speed limit with and without traffic

As with time, fuel usage seems to be higher for situations with traffic than for the ones without. At lower speeds, scenarios with traffic burn on average 20% more fuel. At higher speeds, though, this percentage is reduced to 6. These numbers do as such suggest that traffic's influence decreases at higher speeds.

This is easily visible when plotting the values in Figure 10.14 as well:

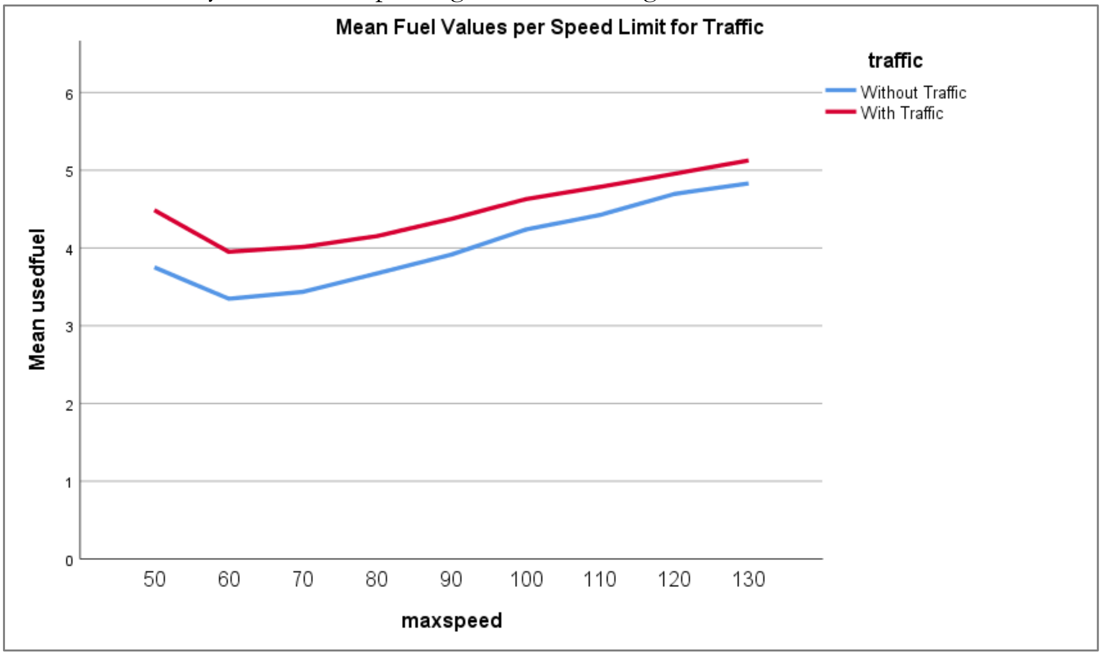


Figure 10.14: Mean values of fuel plotted per speed limit with and without traffic

Thus, both the table and graph suggest that the difference in fuel use becomes somewhat smaller at higher speeds. For the overall picture, using ANOVA, with a significance < 0.001, it can be said that significant differences in mean fuel consumption exists. With an Eta of 0.065, no real effect seems to exist, with only 0.4% of variance in fuel explained by traffic.

As such, traffic has a very weak influence on fuel consumption.

Also, when checking whether or not speed limits still have a significant influence on fuel when differentiating between traffic cases, this was found to be true with a significance of < 0.001. Different effect sizes pop up when looking at the traffic cases separately with Eta: with traffic, Eta equals 0.107 and without traffic 0.150. In both situations the effect size is still small.

10.2.4) Dynamic Speeds Influences on Time

Finally, the influence of dynamic speeds on time and fuel is to be checked, starting with the one on time. Figure 10.15 has the mean values for time consumption for both day-time (lower dynamic speeds) and night-time (higher dynamic speeds).

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Day-time	2452	2019	1948	1604	1530	1466	1438	1413	1400
Night-time	2452	2019	1948	1604	1530	1466	1436	1409	1394

Figure 10.15: Mean values of time per speed limit for day- and night-time

Only for 110, 120 and 130km/h can any differences at all be seen. This is logical, as the higher dynamic speeds are only used for higher speeds. Still, even at such speeds, differences are barely notable: at 130km/h, the difference is a mere 0.4%.

Figure 10.16 shows this as well:

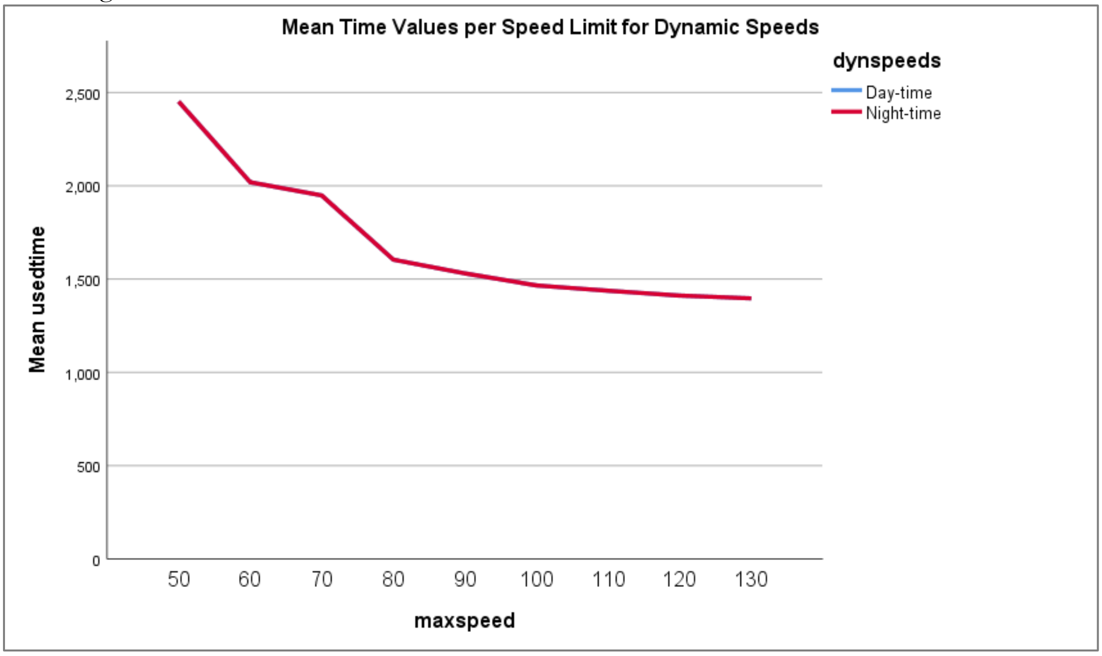


Figure 10.16: Mean values of time plotted per speed limit for day- and night-time

In this graph, no differences whatsoever can even be spotted. This is confirmed by ANOVA, which shows a significance approximating 1, meaning no significant differences exist. It is thus unnecessary to run any further tests.

In conclusion, dynamic speeds have no influence upon time consumption.

10.2.5) Dynamic Speeds Influences on Fuel

Though no influence of dynamic speeds upon time can be proven to exist, it could still have an influence on fuel consumption, as was the case with car models. Figure 10.17 shows the mean fuel values for this factor:

Speed Limit	*50*	*60*	*70*	*80*	*90*	*100*	*110*	*120*	*130*
Day-time	4.12	3.65	3.73	3.91	4.15	4.43	4.57	4.74	4.81
Night-time	4.12	3.65	3.73	3.91	4.15	4.43	4.64	4.91	5.14

Figure 10.17: Mean values of fuel per speed limit for day- and night-time

As with time, differences only exist for the three highest speeds. In percentages, the differences are 2, 4 and 7% for 110, 120 and 130km/h respectively. The graph of Figure 10.18 shows how the differences become larger at higher speeds:

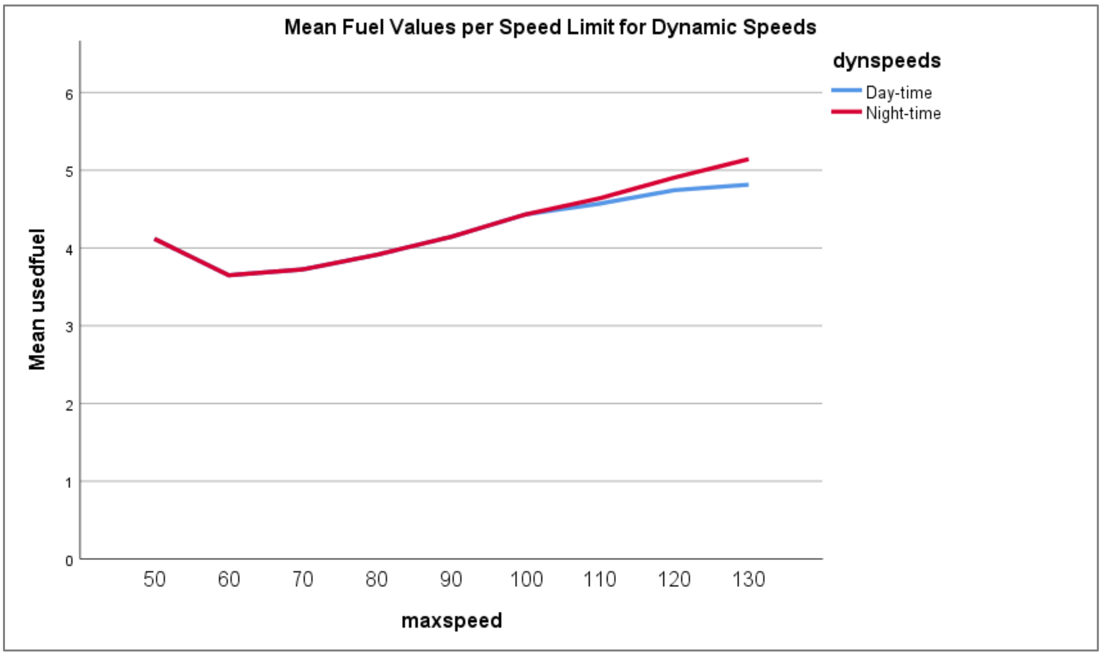


Figure 10.18: Mean values of fuel plotted per speed limit for day- and night-time

As opposed to time, fuel does seem to be influenced by dynamic speeds. ANOVA shows that the differences are significant on the 0.001 level. However, as could be expected from the figures above, the effect size of dynamic speeds is extremely small, with an Eta of 0.009 and an Eta Squared approximating 0.

As such, the effect of dynamic speeds upon fuel, at least when considering all speeds, is virtually non-existent.

11) Conclusion

The current research is nearing its end: the application has been designed, built and used to produce results on paths, which have been analysed statistically. As such, it is now possible to summarise the answers to the sub-questions posed at the start of this study, and to the main question. These sub-questions will be discussed separately first, after which the answers to these questions will be used to answer the main one.

11.1) Sub-question 1 – Car Characteristics & Fuel

The first sub-question is: *"Which car characteristics influence fuel consumption and how?"* To find an answer to this question, a literature study into the fuel consumption of cars and characteristics influencing this was done.

This study showed that six main, relevant characteristics exist here: mass, frontal area, the coefficient of aerodynamic drag, the coefficient of rolling resistance, thermal efficiency and engine displacement. The characteristics influence consumption through different resistances: the frontal area and coefficient of aerodynamic drag increase the fuel consumption linearly through air resistance. The coefficient of rolling resistance and mass do the same linearly through tire friction. Thermal efficiency then, decreases the amount of fuel linearly that is needed to produce certain amounts of power. Engine displacement finally, linearly increases usage through increased fuel usage.

11.2) Sub-question 2 – Car Models & Fuel

The second sub-question is: *"For a chosen group of car models, how does driving at different speeds impact fuel consumption?"* The answer to this question was found by entering the characteristics found in the first sub-question in formulae with which fuel consumption can be approximated. Five car models with varying characteristics were created.

As can be seen in Figure 11.1, all fuel curves follow a similar pattern: fuel efficiency is low at lower speeds, and increases for moderate ones, before declining again at higher speeds. The exact speeds and efficiencies vary between models, but the pattern followed is always the same.

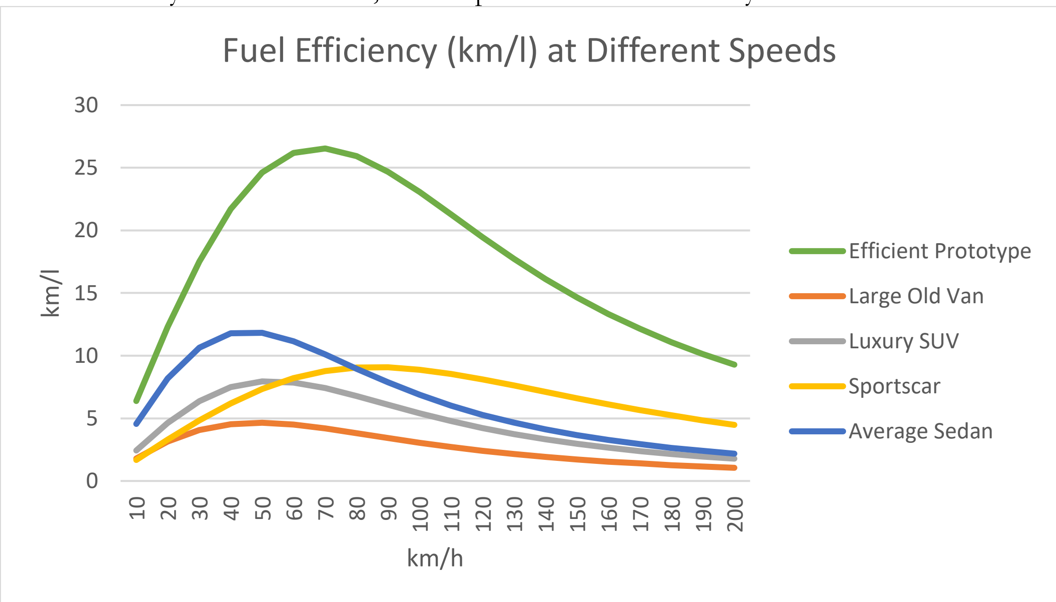


Figure 11.1: Fuel efficiencies at different speeds for all car models

11.3) Sub-question 3 – System Requirements

The third sub-question is: *"Which requirements exist for data and methods for building the desired navigation system?"* While some requirements were already known from the beginning, the setup of the current research is such that additional requirements could be discovered later on, mainly through the answers to the first two sub-questions.

The answer to this question is twofold: it has requirements for both data and methods. The required data for the application were a road network of the Netherlands, data on maximum speeds per road segment, on traffic and dynamic speeds and a height map. During the course of the study the road network was limited to the province of Zuid-Holland and the height map was left out entirely due to the excessive calculation times that including it would have led to.

The methods used in building the application are summarised in the blueprint of Figure 11.2.

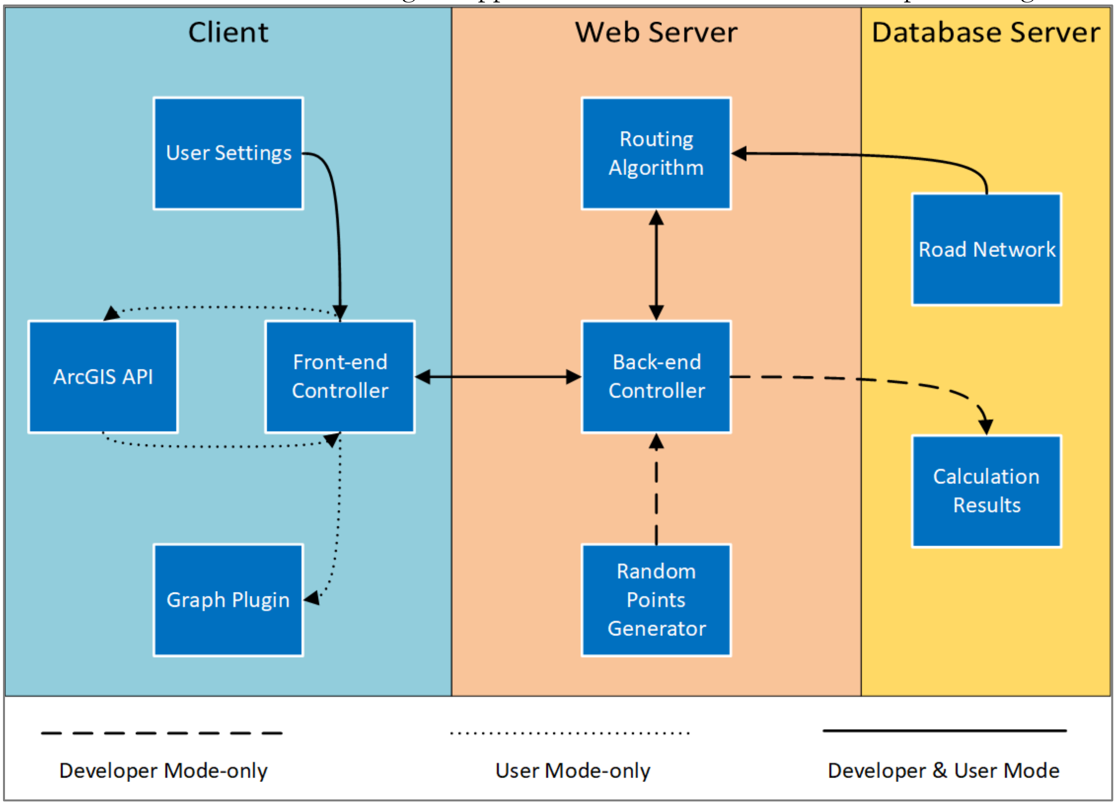


Figure 11.2: Application blueprint

11.4) Sub-question 4 – Time, Fuel & Maximum Speeds

The fourth sub-question is: *"How much fuel and time can be saved or lost by driving at certain maximum speeds in practice?"* This is where the application was used to produce 750 different start and end points, paths between them and data on the time and fuel consumption of these paths. Using statistical methods such as ANOVA and Eta, this data was analysed.

The ANOVA test proves that significant differences exist in mean time consumption for different maximum speeds: it is significant at the 0.001 level. More importantly, the ANOVA's Eta shows that a moderately strong relation exists between speeds and time consumption. The relative differences in mean time used between maximum speeds are substantial: on average, keeping to 130km/h will take only 56% of the time it would take at 50km/h. As such, destinations can be reached almost twice as fast by increasing speed.

For fuel, ANOVA proves that significant differences exist as well, at the 0.001 level. The effect size arrived at using Eta is smaller than for time though: it indicates a weak relation between speeds and fuel. Nonetheless, on average, adhering to different speeds can lead to substantial differences in fuel too: fuel consumption at 60km/h, the most efficient speed, is on average 36% lower than at 130km/h, the most inefficient one. Drivers can, as such, save a third on fuel through limiting their speed.

What is also clear from analysing the results, is that, from an efficiency point of view, it is virtually never worth it to accelerate to the highest speeds. This can be concluded by comparing the wins and losses that accelerating 10km/h brings in terms of time and fuel. For time, savings are significant when working through the lower speeds, while small at higher ones. For fuel, in turn, some savings can be achieved in the beginning, after which higher speeds start to cause higher fuel consumptions. By comparing both savings, it can be concluded that, expressed in litres needed to save one minute of time, the returns diminish rapidly when accelerating, meaning that the faster one goes, the costlier any speed increase becomes.

11.5) Sub-question 5 – Additional Factors

The fifth sub-question is: *"To what extent do these time and fuel differences for different speeds hold when traffic, dynamic maximum speeds and car models are introduced?"* This question could only be relevant in case the fourth one showed that influences of maximum speeds exist on time and fuel. This proves to be the case. Thus, additional ANOVAs and Eta tests were performed here to test for the influence of traffic, dynamic maximum speeds and car models.

Differences in car models do not have any influence at all on time. This is intended, as car models only used different formulae for fuel, and not for time. Fuel however, is, strongly, influenced by car models. The ANOVA shows that significant differences exist, and Eta that a strong effect by models exists. Moreover, the significant differences in time and fuel for different speeds hold for each car model separately: in all cases, these are significant. The effect on fuel is weaker for the efficient prototype and the sportscar and stronger for the remaining ones.

Differing traffic situations does not have a lot of influence on time: only a weak effect is found, as the Eta shows. The effect upon fuel is even smaller, and can, with an Eta < 0.1, hardly be called an effect at all. When looking at the situations with and without traffic, the influence of maximum speeds on both time and fuel is still significant at the 0.001 level in both cases. The effect on time in the scenario with traffic is found to be slightly weaker than without traffic. The effect on fuel too, is somewhat stronger without traffic than with it.

Differentiating between scenarios with lower and higher dynamic speeds, finally, does not have any effect on either time or fuel. The differences for time are not significant, meaning that no effect can be proven. These are significant for fuel, but as the effect itself, measured by Eta, approximates 0, any meaningful effect can be rejected. When differing between lower and higher dynamic speeds, effects of maximum speeds upon time and fuel are still significant, without any notable change in effect size between both situations.

11.6) Main Question

Now that the sub-questions have all been answered satisfactorily, an answer can be given to the main question: *"To what extent does driving at different maximum speeds influence the time and fuel consumption for fastest car routes?"* The first thing to do, is to see whether or not this influence exists for every studied situation, i.e. combinations of car model, traffic and dynamic speeds scenarios. This turns out to be the case: the differences between maximum speeds for time and fuel are always significant on the 0.001 level, in every scenario.

As to the extent of this influence, different conclusions exist for time and fuel. For time, the effect size, indicated by Eta, is always between 0.358 and 0.391, with an average of 0.375. For fuel then, it varies between 0.097 and 0.336. Here, the differences in effect are caused mostly by the differences in car models, as the answer to the fifth sub-question supports.

Finally, though effects may vary between situations, on average, substantial differences in both time and fuel usage can occur when adhering to different speeds: whereas the highest speed leads to, on average, time savings of 56% compared to the lowest one, the most efficient one leads to fuel savings of 36% compared to the most inefficient one.

12) Discussion

With the conclusions drawn up, they can finally be discussed. This is done in three parts: a discussion on what the results mean in practice, the limitations of the study and, finally, recommendations for future research and possible applications.

12.1) Interpretation

Both time and fuel are found to be influenced by the custom maximum speed set by the driver. Higher speeds generally lead to lower time and higher fuel usage, but not as much as one could expect: driving twice as fast does not lead to a 50% time saving, nor does it lead to twice as much fuel used. While this might have seemed logical especially in the time case – the relation between fuel and speed is already not linear – the actual consumptions are largely dependent upon the network: a network without any highways will not yield any time benefits (or fuel losses) when opting for higher custom speed limits, as these can simply not be reached in such a case. The actual values achieved will always depend largely upon this network. In a country such as the Netherlands with a dense road network – a characteristic Zuid-Holland shares – countless alternative routes exist, meaning that not being able to take a highway due to a low custom speed limit does not immediately cause an enormous detour, something which would be possible in areas with less dense networks.

The influence of speeds upon time is found to be larger than the one upon fuel, as the correlation between speeds and the first is higher. A valid question here would be: "Why?" How can this difference be explained? The variance in fuel is known to be smaller than the one in time: whereas the biggest average time difference amounts to 56%, for fuel this is only 36%. That is only half of the answer though: while traffic and dynamic speeds have equally low influence on time and fuel, the effect of car models is totally different: for time it is non-existent, whereas variation in car models does have a strong effect on fuel consumption. As such, it may be fairer not to say that maximum speeds have less influence on fuel, but that other variables – i.e. car models – have more influence in this case.

The other two variables, traffic and dynamic speeds, both seem to have a negligible effect though on both consumptions. This can most probably be attributed to the test area again. In the case of traffic, the high density of the road network yet again offers drivers plenty of alternatives, which means that less time is lost on a small detour than on confronting the traffic.

Dynamic speeds' low influence has a slightly different probable cause: the scarcity of roads with dynamic speeds. Though the area chosen, Zuid-Holland, does not have a smaller proportion of roads with dynamic speeds than the overall country, there are simply not a lot of these roads. This leads to them not being considered in the algorithm that often. In addition, even when considered, the actual change they bring is marginal. The majority of these roads only offer an increase from 120km/h to 130km/h – which is only relevant for the highest custom limit – and only two of them offer an increase from 100km/h to 130km/h. This combination of low occurrence and low impact likely explains the absence of any effect upon time or fuel consumption.

Still, except for car models' influence on fuel, effects might be smaller than could have been expected. This may even not seem logical, seeing as how maximum speeds can on average lead to 56% higher time and 36% higher fuel consumption. The reason for this, is that one factor manages to have an even larger influence upon the final consumptions: the combinations of start and end points. Due to the entirely random nature of these points' selection, the Euclidean distance between points within pairs can differ vastly: in theory, it varies from just a couple of metres to around 90 kilometres (the maximum distance between locations within Zuid-Holland). The locations of these points will in the end be the main factor in determining consumptions: after all, independent of speed, traffic, the time of day and your car, a trip to the local supermarket a couple of streets away will inevitably be less consuming than one to the other side of the province. This is proven when checking for the ANOVA's Eta here: an Eta of 0.875 (explained variance of 76.5%) for time and an Eta of 0.650 (explained variance of 42.2%) for fuel show that these point combinations have a substantially larger influence on consumption than any other variable, though the effect size of car models on fuel does come somewhat close with an Eta of 0.619. As such, it is much more relevant to look at the amounts of variance caused by differences in maximum speeds than at the found effect sizes.

One of the purposes of the current research was to make differences in consumption visible to drivers, as to increase their awareness and stimulate them to think as to how their driving style influences this. As to this, the research seems to be able to in fact offer them relevant information: both time and fuel do often differ significantly for different speed limits. The fact that these differences hold for both variables, and not just one, is crucial here. It requires users to actually make their own decisions, as two different criteria – getting somewhere as fast and cheap as possible – need to be compared by them. The end result of this comparison will depend upon the weights that they themselves attribute to each factor, which may differ between certain moments in time, e.g. when in a hurry and when not. Had this not been the case, it would have been a matter of simply optimising the path for one variable – e.g. time – and just recommending that one, without requiring the driver to think about the consequences of the different options offered. Thus, this goal of the research seems to have been reached.

12.2) Limitations

As is mentioned in the introduction of Chapter 7, it is extremely rare for projects or studies to finish without any scope changes occurring: unforeseen events or conditions will likely make at least some changes necessary. This does not need to cause any problems, as long as there are valid reasons for them and, ideally, ways to minimise any negative effects caused by the changes. In addition, there will always be choices, whether made initially or at a later stage, that might have influenced the validity of the study results.

Throughout the current research as well, some sacrifices had to be made as to the original plans, and methods and data used do not always reflect reality fully – which, as it concerns modelling of reality, is never entirely possible anyhow.

The first simplification of reality had to be applied during the creation of the fuel consumption formulae, the determination of which characteristics to include and how this would culminate into car models. The final list of car characteristics used for varying between car models, consists of six characteristics: in this model, in addition to these characteristics, only speed and – originally (see below) – grades exist. In reality, however, fuel consumption is determined by many more factors than only eight: as mentioned in Chapter 4, this list is endless in reality. As with any model, however, a model is only as good as its ability to capture complex real-world problems in simplified versions of reality that we can understand. That is what the model used here tries to do, using the characteristics with the most impact only. Still, that means that some harder to model though still possibly relevant factors, such as gears and acceleration, have not been included. Still, as supported by literature studied – e.g. Ross (1994) – this model should provide a satisfactory amount of validity.

The biggest change as compared to the original plan, concerns the use of a height map. It was the intention to use a height map to deduct slopes of roads, as grades were found to have an influence on fuel consumption. This did, however, prove not to be possible here, due to operational limitations: it would not have been possible to pre-process the height map into grades for road segments within an acceptable time frame for this study. While not ideal – grades were to be used with fuel calculations – the largely flat study area chosen meant that the effects of leaving out grades are way smaller than they would have been in areas with rougher terrains. In such different areas, though, this could have posed serious problems to the validity of the results, as it would mean that a significant factor in fuel consumption would be left out.

The other change pertains to the study area as well. Whereas it was intended to include the entire country of the Netherlands, high calculation times for paths using the full network meant that the allowed time frame for these calculations would not have sufficed. The choice was made to limit the network to the province of Zuid-Holland, as to limit calculation times to acceptable values. This was achieved, though limiting the area does make it less representative for the original area: though care was taken to preserve representativeness, having had the ability to include the full network would have increased the validity. This limitation did, however, also allow for increasing validity through another measure: increasing the number of runs. Since the calculation time per run was brought down to values lower than aimed at originally, the total number of paths calculated was raised from 90,000 (500 runs) to 135,000 (750 runs), a 50% increase.

As such, though the results of this research are deemed to be valid, it would certainly be possible to repeat it in an even more valid way. While extending the fuel model is not deemed beneficial for reasons of abstraction mentioned already here, including a height map and the entirety of the Netherlands could prove beneficial. Though care has been taken to select a representative province, having a larger area could already lead to more variety in possible routes, possibly leading to new insights. Access to more powerful equipment than was available to the researcher – a mere i3 2.3 GHz processor machine – would remove the technical limitations encountered here.

12.3) Recommendations

Some recommendations for removing limitations that were present during this research were already given in the previous sections. Rather than improving what has already been done here, though, there is also the possibility to build upon and extend this work, which is what is discussed here.

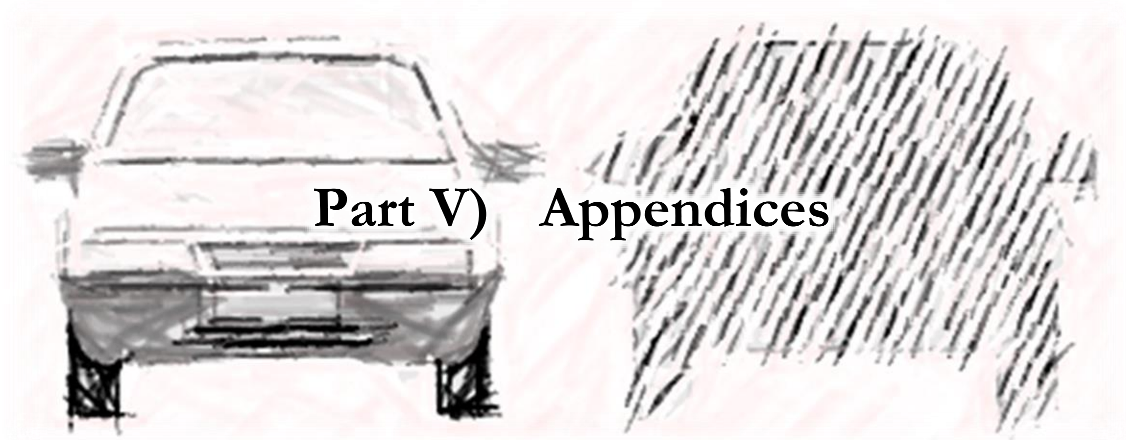
Though it has been recommended here not to make the fuel model itself more complex, i.e. not to add more detailed characteristics, it would be interesting to create a comparison between totally different cars, e.g. based on their engine: as was determined in Chapter 4, vehicles with different engine types, fuelled by different fuel types, can behave significantly differently: electric vehicles, for example, do not only consume fuel – electricity – while driving but are also able to produce some from energy generated when braking. Incorporating these changes would mean having to use fundamentally different formulae for different engine types, which could be done in future studies focusing on this aspect.

In addition to comparing different vehicle types within one study, it would also be possible to instead focus on a very specific vehicle case, and fleshing this out. Examples could be delivery trucks within urban areas, or any other logistical cases. Here, both the vehicle characteristics and study area could prove to be interesting factors to look at.

This study area could actually already be a research focus in itself, though. It has been mentioned earlier in this chapter that the Netherlands (and Zuid-Holland) have a relatively dense road network. This allows drivers to relatively easy pick alternative routes when circumstances such as traffic call for this, without having to make huge detours. This raises the question of how strong the effect of the road density can be on such factors: how much quicker are detours made in dense networks as compared to less dense ones, and what is the effect of factors such as traffic?

Such are mostly suggestions for further research. The current research does however also offer options for continuing in a more practical direction. One such practical application is one that was mentioned already in Chapter 1 while discussing the relevance of doing this research. Governments could use this to study the effects on traffic flows of changing the speed limit on roads, was the suggestion given there. This could, however, be extended way more: since this application studies the paths that are created when adhering to many different speed limits, it could be used to analyse the behaviour of different types of motorists, e.g. between those who prefer to drive as fast as legally possible and those who prefer to take it easy and take backroads with lower speeds. How would a change in speed limit, the (temporary) closure of a road or the introduction of a dynamic speed influence the paths that these different kinds of drivers take, could be interesting questions to organisations involved in road network planning.

Finally, one of the original goals of this application, which is to give drivers a choice and to make them more aware of their choices and their consequences for time and fuel, could fairly easily be extended to any other area of interest. Time can be weighed against fuel usage, but it could be equally interesting to compare time used with aspects such as the touristic value of a route (how many landmarks do I see per hour driven?) or any other value of interest. The options here are endless, and when implemented properly could offer drivers many different alternative routes to choose from – it could finally allow them to drive their way.



Part V) Appendices

Appendix I – References

- Apache (2017), Apache HTTP Server Version 2.4 Documentation. Available at: https://httpd.apache.org/docs/2.4/. Accessed on: 03/10/2017.
- Association for Computing Machinery (2017), Standards, APIs, Interfaces and Bindings. Available at: http://oldwww.acm.org/tsc/apis.html. Accessed on: 14/11/2017.
- Autocar (2017), New Toyota Yaris 1.5-litre Features Advanced Exhaust Cooling System. Available at: https://www.autocar.co.uk/car-news/new-cars/new-toyota-yaris-15-litre-features-advanced-exhaust-cooling-system. Accessed on: 03/11/2017.
- Biggs, D.C. & R. Akçelik (1987), Estimating Effect of Vehicle Characteristics on Fuel Consumption. ASCE Journal of Transportation Engineering 113(1), pp. 101-106.
- Chart.js (2017), Chart.js. Available at: http://www.chartjs.org/docs/latest/. Accessed on: 14/11/2017.
- BMW (2013), Navigation, Entertainment, Communication. Available at: http://www.bmwpkod.com/Media/Default/Documents/BMWNA%202013%20Navigtion%Owners%20Manual.pdf. Accessed on: 20/09/2017.
- Born, G. (2011), *HTML5 – Das umfassende Praxis- und Referenzwerk*. München, Germany: Markt + Technik Verlag.
- Clark, S.K. & R.N. Dodge (1979), *A Handbook for the Rolling Resistance of Pneumatic Tires*. Ann Arbor, MI: University of Michigan.
- Citroën (2013), Citroën eMyWay – Manuel d'Installation. Available at: https://citroen.navigation.com/static/WFS/ShopSite/CitroenEMEA/ShopCitroenEMEA/en_US/Landing%20Pages/Citroen%20User%20Guides%20Page/citroen_emywy_download_guide_multi.pdf. Accessed on: 20/09/2017.
- Driels, M.R. & Y.S. Shin (2004), Determining the Number of Iterations for Monte Carlo Simulations of Weapon Effectiveness. Available at: http://www.dtic.mil/dtic/tr/fulltext/u2/a423541.pdf. Accessed on: 17/10/2017.
- Dumke, R., M. Lother, C. Wille & F. Zbrog (2003), *Web Engineering*. München, Germany: Pearson Studium.
- Dvorski, D.D. (2007), Installing, Configuring, and Developing with XAMPP. Available at: http://dalibor.dvorski.net/downloads/docs/InstallingConfiguringDevelopingWithXAMPP.pdf. Accessed on: 03/10/2017.
- DuurzaamBedrijfsleven.nl (2016), Forse brandstofbesparing Connexxion door zuinige chauffeurs. Available at: https://www.duurzaambedrijfsleven.nl/mobiliteit/12251/zuinige-chauffeurs-leveren-connexxion-forse-brandstofbesparing-op. Accessed on: 30/09/2017.
- EpiX Analytics (2017), The minimum number of iterations one can run in a Monte Carlo Simulation. Available at: http://www.epixanalytics.com/modelassist/AtRisk/Model_Assist.htm#Montecarlo/The_minimum_number_of_iterations_one_can_run_in_a_Monte_Carlo_simulation.htm. Accessed on: 17/10/2017.
- Esri (2017a), ArcMap Overview. Available at: http://desktop.arcgis.com/en/arcmap/. Accessed on: 03/10/2017.
- Esri (2017b), ArcGIS API for JavaScript Overview. Available at: https://developers.arcgis.com/javascript/3/jshelp/. Accessed on: 14/11/2017.
- European Parliament (2006), Directive 2006/126/EC of the European Parliament and of the Council of 20 December 2006 on driving licences. Available at: http://eur-lex.europa.eu/eli/dir/2006/126/oj. Accessed on: 09/10/2017.
- Financieel Dagblad (2017), Dijksma gaat aanpak luchtvervuiling versneld doorvoeren. Available at: https://fd.nl/economie-politiek/1217432/rechter-eist-van-staat-aanpak-luchtvervuiling. Accessed on: 30/09/2017.
- Gillespie, T.D. (1992), *Fundamentals of Vehicle Dynamics*. Warrendale, PA: Society of Automotive Engineers, Inc.
- Held, G. (2000), *Server Management (Best Practices)*. Boca Raton, FL: CRC Press.
- Het Nieuwe Rijden (2017), Rijtips. Available at: http://hetnieuwerijden.nl/rijtips/. Accessed on: 30/09/2017.

- Hirt, C., S. Claessens, T. Fecher, M. Kuhn, R. Pail & M. Rexer (2013), New Ultrahigh-Resolution Picture of Earth's Gravity Field. Geophysical Research Letters 40, pp. 4279-4283.
- Hucho, W.H. & G. Sovran (1993), Aerodynamics of Road Vehicles. Annual Review of Fluid Mechanics 1993(25), pp. 485-537.
- IBM (2017), IBM SPSS Statistics 24 Documentation. Available at: http://www-01.ibm.com/support/docview.wss?uid=swg27047033. Accessed on: 03/10/2017.
- ING (2013), Themabericht Transport en Logistiek – Aanpakken verborgen brandstofkosten. Amsterdam, The Netherlands: ING.
- Jesus Madureira Porfírio, S.F. de, I.I. Klatt, S. van der Klei & V. Straβer (2017), Taking the Scenic Route – Final Report. Available at: https://1drv.ms/b/s!AkC3FNVsViStnakU8lCvqs_zlzffxQ. Accessed on: 02/10/2017.
- Kadaster (2018), Rijksdriehoeksstelsel. Available at: https://www.kadaster.nl/rijksdriehoeksstelsel. Accessed on: 19/01/2018.
- Kedar, S. (2009), *Database Management System*. Pune, India: Technical Publications Pune.
- Koch, S. (2011), *JavaScript – Einführung, Programmierung und Referenz*. Heidelberg, Germany: dpunkt.verlag GmbH.
- Laborenz, K. (2016), *CSS – Das umfassende Handbuch*. Bonn, Germany: Rheinwerk Verlag GmbH.
- Lammertsma, P.F. (2005), Satellite Navigation. Utrecht University, February 2nd, 2005.
- Leaflet (2018), Leaflet 1.3.1 Reference. Available at: http://leafletjs.com/reference-1.3.0.html. Accessed on: 05/02/2018.
- Letzel, S. & R. Gacki (2001), *Jetzt lerne ich MySQL & PHP*. München, Germany: Markt + Technik Verlag.
- Man (2017), Man TGX D38. Available at: https://www.truck.man.eu/de/de/lkw/tgx/tgx-d38/Highlights.html. Accessed on: 03/11/2017.
- Meredith, J.R., S.M. Shafer, S.J. Mantel Jr. & M.M. Sutton (2014). *Project Management in Practice*. Hoboken, NJ: Wiley.
- Michelin (2017), ViaMichelin. Available at: https://www.viamichelin.com/. Accessed on: 20/09/2017.
- Ministerie van Infrastructuur en Milieu (2017), Objectenhandboek BGT – IMGeo. Available at: http://imgeo.geostandaarden.nl/def/imgeo-object/paal/verkeersregelinstallatiepaal-niet-bgt. Accessed on: 20/11/2017.
- Missouri University of Science and Technology (2017), Probabilistic Engineering Design. Available at: http://web.mst.edu/~dux/repository/me360/ch8.pdf. Accessed on: 09/10/2017.
- Nationale Databank Wegverkeersgegevens 2017), Actuele Verkeersgegevens. Available at: http://www.ndw.nu/pagina/nl/4/databank/31/actuele_verkeersgegevens/. Accessed on: 07/11/2017.
- NOS (2017), TomTom wil zelfrijdende auto's straks de weg wijzen. Available at: https://nos.nl/artikel/2153666-tomtom-wil-zelfrijdende-auto-s-straks-de-weg-wijzen.html. Accessed on: 30/09/2017.
- OpenStreetMap (2017a), OpenStreetMap Wiki – key:surface. Available at: http://wiki.openstreetmap.org/wiki/Key:surface. Accessed on: 30/10/2017.
- OpenStreetMap (2017b), OpenStreetMap Wiki – Tag:highway=traffic_signals. Available at: http://wiki.openstreetmap.org/wiki/Tag:highway%3Dtraffic_signals. Accessed on: 20/11/2017.
- Overheid.nl (2016), Wet hergebruik van overheidsinformatie. Available at: http://wetten.overheid.nl/BWBR0036795/2016-10-01. Accessed on: 03/10/2017.
- Overheid.nl (2017a), NWB – National Wegen Bestand – Wegen – Wegvakken. Available at: https://data.overheid.nl/data/dataset/nationaal-wegen-bestand-wegen-wegvakken. Accessed on: 03/10/2017.
- Overheid.nl (2017b), Wegkenmerken +/ Speed. Available at: https://data.overheid.nl/data/dataset/wegkenmerken-speed. Accessed on: 03/10/2017.
- PDOK (2017), AHN3 downloads. Available at: https://www.pdok.nl/nl/ahn3-downloads. Accessed on: 07/11/2017.
- pgRouting (2018), Documentation. Available at: http://pgrouting.org/documentation.html. Accessed on: 19/01/2018.

- PHP Documentation Group (2017), PHP Manual. Available at: http://php.net/download-docs.php. Accessed on: 03/10/2017.
- PostGIS (2017), PostGIS 2.3.4dev Manual. Available at: http://postgis.net/stuff/postgis-2.3.pdf. Accessed on: 03/10/2017.
- PostGIS (2018), ST_Transform. Available at: https://postgis.net/docs/ST_Transform.html. Accessed on: 19/01/2018.
- PostgreSQL (2017), PostgreSQL Wiki. Available at: https://wiki.postgresql.org/wiki/Main_Page. Accessed on: 03/10/2017.
- Rexer Analytics (2015), Rexer Analytics Survey Results. Available at: http://r4stats.com/2015/10/13/rexer-analytics-survey-results/. Accessed on: 03/10/2017.
- Rijkswaterstaat (2014), Handleiding Nationaal Wegenbestand. Available at: https://www.rijkswaterstaat.nl/apps/geoservices/geodata/dmc/nwb-wegen/productinfo/beschrijvende_documentatie/. Accessed on: 03/10/2017.
- Rijkswaterstaat (2017), Maximumsnelheden vanaf 23 juni 2017. Available at: https://staticresources.rijkswaterstaat.nl/binaries/Wegenkaart%20maximumumsnelheden%2023%20juni%202017_tcm21-115468.pdf. Accessed on: 07/11/2017.
- Ross, M. (1994), Automobile Fuel Consumption and Emissions: Effects of Vehicle and Driving Characteristics. Annual Review of Energy and the Environment 1994(19), pp. 75-112.
- Shelquist, R. (2016), An Introduction to Air Density and Density Altitude Calculations. Available at: http://wahiduddin.net/calc/density_altitude.htm. Accessed on: 30/10/2017.
- Smith, M.J. de, M.F. Goodchild & P.A. Longley (2015), *Geospatial Analysis – A Comprehensive Guide to Principles, Techniques and Software Tools*. Leicester, United Kingdom: Troubador Publishing Ltd.
- Stoica, M., M. Mircea & B. Ghilic-Micu (2013), Software Development: Agile vs. Traditional. Informatica Economiča 17(4), pp. 64-76.
- Tao, Z., P. Bonnifait, V. Frémont, J. Ibanez-Guzman & S. Bonnet (2017). Road-Centered Map Aided Localization for Driverless Cars Using Single-Frequency GNSS Receivers. Journal of Field Robotics 34(5), pp. 1010-1033.
- Theis, T. (2013), *Einstieg in PHP 5.5 und MySQL 5.6*. Bonn, Germany: Galileo Press.
- The New York Times (2013), G.M. Had a Version of OnStar in 1966. Available at: https://wheels.blogs.nytimes.com/2013/08/05/g-m-had-a-version-of-onstar-in-1966/. Accessed on: 20/09/2017.
- TNO (2013), De truck van de toekomst. Den Haag, The Netherlands: TNO. Norouzi, N., M. Sadegh-Amalnick & R. Tavakkoli-Moghaddam (2017), Modified particle swarm optimization in a time-dependent vehicle routing problem: minimizing fuel consumption. Optimization Letters, 11(1), pp. 121-134.
- US Department of Energy (2003), Just the Basics – Diesel Engine. Available at: https://www1.eere.energy.gov/vehiclesandfuels/pdfs/basics/jtb_diesel_engine.pdf. Accessed on: 31/10/2017.
- Vocht, A. de (2013a), Variantieanalyse en niet-parametrische toetsen. Slides pertaining to the ‘Statistische Methoden’ course at Utrecht University. Accessed on: 06/02/2018.
- Vocht, A. de (2013b), Enkelvoudige regressie en correlatie. Slides pertaining to the ‘Statistische Methoden’ course at Utrecht University. Accessed on: 06/02/2018.
- Watzenig, D. & B. Brandstätter (2017), *Comprehensive Energy Management – Eco Routing & Velocity Profiles*. Cham, Switzerland: Springer.
- Wieczorek, W.F. & A.M. Delmerico (2009), Geographic Information Systems. Computational Statistics 1(2), pp. 167-186.
- Wu, J. & J. Liu (2012), A Forecasting System for Car Fuel Consumption Using a Radial Basis Function Neural Network. Expert Systems with Applications 39(2012), pp. 1883-1888.
- Yan, M. (2017), Dijkstra’s Algorithm. Available at: http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf. Accessed on: 20/11/2017.
- Zhan, F.B. (1997), Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures. Journal of Geographic Information and Decision Analysis 1(1), pp. 70-82.

- Zhou, M., H. Jin & W. Wang (2016), A Review of Vehicle Fuel Consumption Models to Evaluate Eco-driving and Eco-routing. Transportation Research 49(2016), pp. 203-218.
- Zhu, L., J. Holden, J. Gonder & E. Wood (2017), Green Routing Fuel Saving Opportunity Assessment: A Case Study on California Large-Scale Real-World Travel Data. 28th IEEE Intelligent Vehicles Symposium.

Appendix II – Glossary

- **AHN**: 'Actueel Hoogtebestand Nederland', a heightmap of the Netherlands. See Chapter 6.1.1.
- **API**: 'Application Programming Interface', website plugins relying on external applications. See Chapter 6.3.3.
- **ArcGIS/ArcMap**: See Chapter 3.3.5.
- **BGT**: 'Basisregistratie Grootschalige Topografie', a large-scale map of topography in the Netherlands. See Chapter 6.3.6.
- **DBMS**: 'Database Management System', a system allowing for the storage, manipulation and viewing of data.
- **GIS**: 'Geographic Information System' or 'Geographic Information Science'.
- **NWB**: 'Nationaal Wegenbestand', see Chapter 3.2.1.
- **OSM**: 'OpenStreetMap', an open-content mapping service. See Chapter 6.3.6.
- **pgRouting**: a path-finding extension to PostGIS. See Chapter 8.5.
- **PostGIS**: See Chapter 3.3.2.
- **PostgreSQL**: See Chapter 3.3.1.
- **SPSS**: See Chapter 3.3.4.
- **Stand-alone Application**: A computer program that needs to be installed and run on every system that it is used on, as opposed to a website or platform design.
- **XAMPP**: See Chapter 3.3.3.