

From to a ten

Congestion analysis on the
A4 between Hoogmade and
Zoeterwoude-Dorp

BART GUINÉE

From A4 to a ten

Congestion analysis on the A4 between Hoogmade and Zoeterwoude-Dorp

By

Bart Guinée

4365275

In partial fulfilment of the requirements for the degree of

Bachelor of Science

in Civil Engineering

at the Delft University of Technology,

Supervisors: Yufei Yuan TU Delft

Maria Salamons TU Delft

Cover page picture by (Any to Many, 2014)



PREFACE

This report is written as my bachelor thesis for the fulfillment of the Degree of Bachelor of Science in Civil Engineering. The report has been written in the section of Transport and Planning. I chose this section as I am still in doubt in which section I would like to do my master and Transport and Planning sure is one of the options.

The report is an analysis of the congestion problems on the highway A4 between Hoogmade and Zoeterwoude-Dorp. I chose this subject as I think it's a very interesting one. The highway has been reconstructed for a huge amount of money over the last years while the situation only got worse. At the moment there are more congestion problems than there were before the reconstruction of the road. Also, I personally know the surrounding area and the highway itself very well, as I grew up in Leiderdorp. My parents still live there, which is why I still use the road often, which makes it an interesting case for me.

As said before the report is an analysis on the congestion problems on the A4 around Leiden. The bottlenecks in the layout of the road are defined as well as the problems which occur at these bottlenecks. This is all done by an analysis of different data sets on the traffic flow on the road.

Most of the techniques that I used to do this were actually new for me. I learned a lot about traffic flows on highways and theories behind congestion problems during the project. Also I improved my programming skills to be able to perform the data analyses. I learned a lot about the setup, working plan, presentation and the overall process of a big project of course.

At last I would like to thank Yufei Yuan and Maria Salamons for mentoring me during the project. I would also like to thank Rijkswaterstaat and the NDW for providing me all the information and data that I needed. Without their help the project couldn't have been completed.

Bart Guinée

SUMMARY

This report is an analysis of the congestion problems on the A4 between Hoogmade and Zoeterwoude-Dorp as the subtitle of the report states. This part of the highway has been one of the bottlenecks in the Netherlands with the most congestion problems for more than a decade. The congestion problems eventually got so huge that the Ministerie van Verkeer en Waterstaat decided to reconstruct the highway in 2009. The reconstruction of the road resulted to promising results during the construction, with more than a halving of the total level of congestion in 2013 in the direction of Amsterdam in comparison to the level of congestion in 2009 on the same track. But more than two years after completion it can be stated that the project wasn't a success. The level of congestion in the direction of Amsterdam was in 2016 five times as large as it was in 2013 and more than two times as large than 2009. Similar numbers are applicable to the track in the direction of The Hague.

In this report these current congestion problems on the track are analyzed. With the main research question: What is the main cause of the current congestion problem on the A4 between Hoogmade and Zoeterwoude-Dorp? The following sub-questions were defined: Where can the bottlenecks on the layout of the road be found? What are the capacity and the demand at these bottlenecks? What percentage of the total level of congestion on the road is caused by accidents? What percentage of the total level of congestion caused by other factors, such as road maintenance and road blockage?

To answer these questions a lot of data analyses has been performed. Congestion data is analyzed to find the point on the layout of the road where most congestions start. This to find the most important bottlenecks on the layout of the road. These bottlenecks are further investigated to answer the second sub-question. To do so the demand and the capacity of these bottlenecks is estimated by analyzing the congestion data together with the intensity data according to the vertical queuing model. Furthermore accident data and congestion data have been analyzed to find out what the part of accidents in the congestion problems is. The exact time of when the accidents occurred have been compared to the starting time of the congestions on the track to find this part. A similar analysis is performed with data of road maintenance and road blockage together with the congestion data to find out the part of road maintenance and road blockage is in causing congestion problems.

The outcome of the bottleneck analysis stated that the most important bottlenecks occur at the points where traffic from multiple strips has to merge into a traffic flow on less strips. This happens at almost all highway entries and at the points in both directions where the layout of the road changes from three to two strips on the long-distance traffic lanes. The results of the accident analysis show that 22.5% and 11.5% of the total level of congestion in 2016 in the directions of respectively The Hague and Amsterdam was caused by traffic incidents. The part of the total level of congestion which was caused by road blockage and road maintenance in 2016 is negligibly small.

Out of these results can be concluded that the congestion problems on the road are mainly caused by the fact that the average demand during rush hours is higher than the capacity of the bottlenecks. The accidents on the track do have a role in the total level of congestion on the road. But this role is not higher than the average role of accidents in causing congestion on all the highways in the Netherlands.

There are some options which can be worked out further which could reduce congestion problems on the road. At the bottlenecks at which the traffic flow on the long-distance lane has to merge from a flow on three to a flow on two strips should be looked into adding an extra strip. Ramp metering systems could be introduced at the bottlenecks at which traffic entries the road. This to control the flow entering the highway as this flow is causing the congestion at these points. The layout of road could be changed at the bottlenecks at which the layout is illogical at the moment. But these options are all just recommendations and should be investigated further in future research before applying them to the road.

CONTENTS

Preface.....	2
Summary	3
1 Introduction.....	6
2 Situation before and after 2014.....	8
1.1 Situation before 2014.....	8
1.2 Situation after 2014.....	8
3 Analysis Methodology	10
3.1 Methodology determining total level of congestion	10
3.2 Methodology bottleneck analysis	11
3.3 Methodology peak hour analysis	11
3.4 Methodology Demand and capacity analysis.....	11
3.4.1 Methodology demand analysis	13
3.4.2 Methodology capacity analysis	13
3.4.3 Methodology Comparison demand and capacity in each bottleneck.....	14
3.5 Methodology analysis other congestion causes	15
3.5.1 Methodology Accident analtsis	15
3.5.2 Methodology road maintenance analysis	15
3.5.3 Methodology road blockage analysis.....	16
4 Results	17
4.1 Results bottleneck analysis	17
4.2 Results peak hour analysis	19
4.3 Results demand and capacity analysis	20
4.4 Results analysis of other causes	20
4.4.1 Congestion caused by accidents.....	20
4.4.2 Congestion caused by road maintenance	21
4.3.3 Congestion caused by road blockage	21
5 Conclusion	22
6 Recommendations	23
6.1 Increase of capacity.....	23
6.1.1 Extra strips.....	23
6.1.2 Change the road layout	23
6.2 Decrease of demand	24
6.2.1 Ramp metering systems	24

6.2.2	Avoid peak hour compensation	24
References.....		25
Appendices		26
Used data		26
Python scripts.....		33

The Netherlands is a small but very densely populated country. This leads to busy roads and congestion on the main road network. This report is a congestion analysis on one of the roads with the biggest congestion problems in the Netherlands, the highway A4 (E19) near Leiden. The highway is positioned in the middle of the Randstad, the most densely populated area of the Netherlands. The A4 connects the two main cities of the Netherlands, Amsterdam and Rotterdam, with each other as well as the two main ports of the Netherlands, Schiphol airport and the Port of Rotterdam. Along the route the road passes some other major cities, which are in order from north to the south: Hoofddorp, Leiden, The Hague and Delft. This all together makes it one of the most important and busiest roads of the Netherlands.

As said before, busy roads can lead to congestion problems, which is the case at the A4 near Leiden. Before the year of 2009, the congestion problems became so huge that the Dutch government decided to transform the highway (Rijkswaterstaat, 2009). The project was meant to increase the capacity of the road and reduce the congestion problems. The project started in 2009 and was finished in 2014. But the results and the impact of the project weren't as expected. The road was after the construction still one of the biggest bottlenecks of the country and the situation did not improve compared to the situation before the start of the project. Before the project the bridge across the Oude Rijn or the bottleneck at Zoeterwoude-Rijndijk stood in the top three of the File top 50 of the Dutch traffic information service (VID) every year between 2006 and 2012 (VID, 2006-2012). After the reconstruction in 2016 the bottleneck at Zoeterwoude-Dorp was back at the top of the File top 50 (VID, 2016), out of which can be concluded that the situation did not improve. At the moment there are some not final plans to reconstruct the road again (Rijkswaterstaat, 2017). But before this is done the causes of the congestion problems on the road need to be found.

In this report the traffic congestion on the new A4 between Hoogmade and Zoeterwoude-Dorp is analyzed. The main research question was: What is the main cause of the current traffic congestion problem on the A4 between Hoogmade and Zoeterwoude-Dorp? Furthermore the following sub-questions were defined and investigated to come to an answer on the main question: Where can the bottlenecks on the layout of the road be found? What is the capacity and the demand at these bottlenecks? What is the percentage of the level of congestion caused by accidents on the road? What is the percentage of the level of congestion caused by other factors, such as road maintenance and road blockage? The level of congestion (kmmin) is the length of a congestion (km) multiplied by the duration of the congestion (min). The outcome of the main- and sub-questions are useful for further research on the improvement of the road. The design of the road can be improved when the weak spots are known.

The parties which will be the most interested in the results are the Ministerie van Verkeer en Waterstaat and Rijkswaterstaat. As they are respectively the client for the reconstruction of the road and the administrator of the road. Other stakeholders are the province of Zuid-Holland as the road is located on their territory and the municipalities of Leiderdorp, Leiden, Zoeterwoude and Kaag en Braassem as their inhabitants are directly influenced by the redesign and reconstruction of the road.

The research is also limited within the boundaries of these municipalities. More precise the exact track which is investigated is the track between highway exit 6 at Hoogmade in the north and highway exit 7 at Leiden-Zuid in the south. A clarifying image of the location can be found in figure 1 at the end of the chapter. Furthermore the data analysis of the research is limited to data from the year 2016 unless indicated differently. This as 2016 is the last year with a complete set of measurement data.

As stated above a data analysis is executed to estimate the answers on the main- and sub-questions of the research. In first the most important bottlenecks are determined. After this the demand and capacity of these bottlenecks determined. The results of these two are compared afterwards. The percentage of level of congestion caused by accidents is estimated by comparing the congestion data with data of all the accidents which occurred on the investigated track. The percentage of the

congestion caused by road maintenance and road blockage is found in same manner but then by comparing the congestion data with the road maintenance and road blockage data. Afterwards the results of all the executed data analyses will be compared with each other to give a final answer on the research question.

To identify the problem more precise the exact situation before and after the reconstruction of the road is described in chapter two. The detailed methodology on the data analysis is described in chapter three. The results of this data analysis can be found in chapter four, with the conclusion coming from these results in chapter five. A recommendation based on the conclusion of the results can be read in chapter six. As last the bibliography and appendices are added at the end of the report.

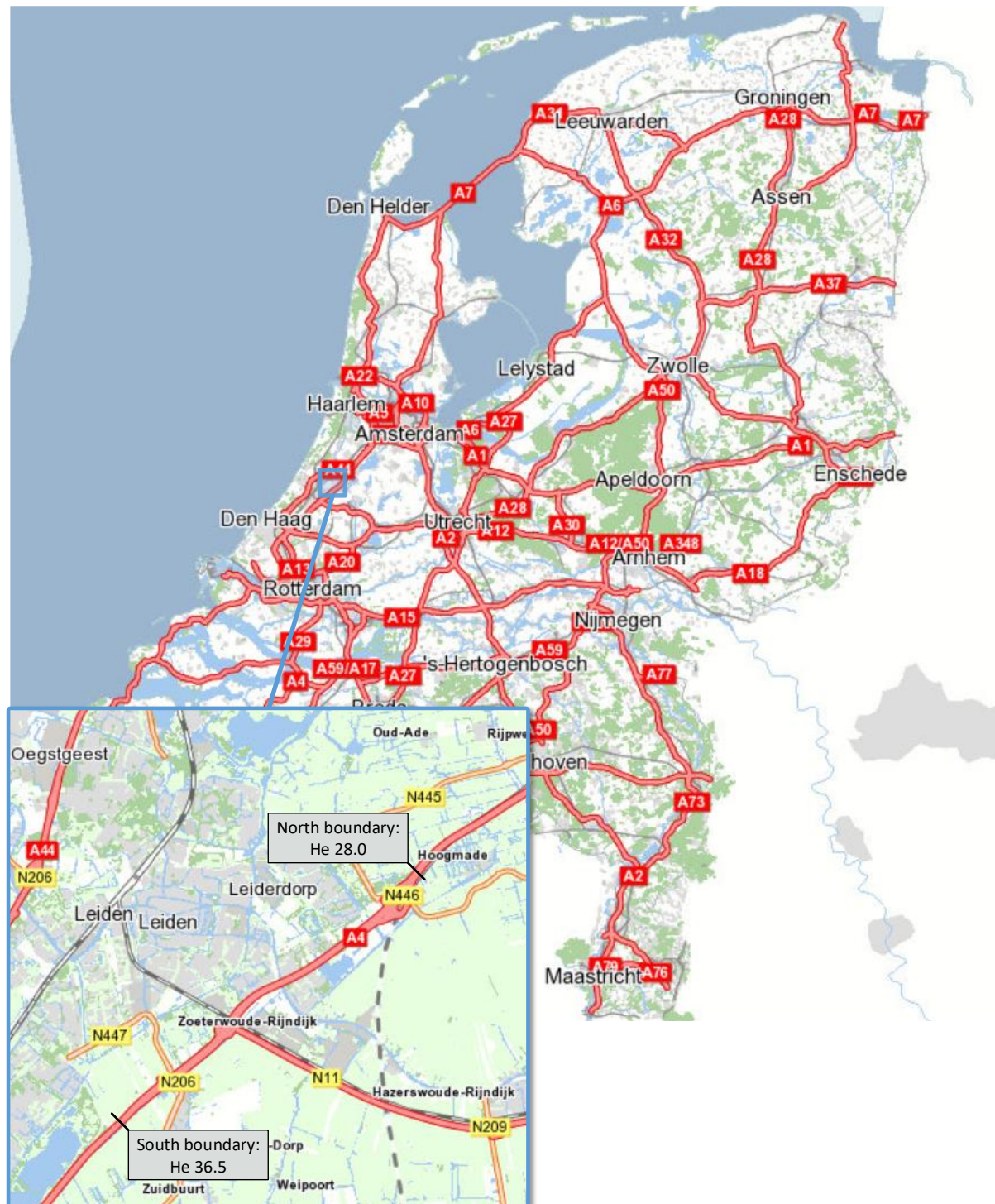


Figure 1: Location of the research area (OpenStreetMap, 2017)

2 SITUATION BEFORE AND AFTER 2014

1.1 SITUATION BEFORE 2014

As stated in the introduction this report is concentrated on the part of the A4 between Hoogmade and Zoeterwoude-Dorp. This part of the A4 has undergone a big transformation in the last years. Before the transformation the road existed out of two times two strips. It was the part of the highway with the fewest amount of strips in both directions. As the road was already widened to two times three strips in the north and the south in the directions to respectively Amsterdam and The Hague. This all is illustrated in figure 2 at the end of the paragraph. The layout of the figures which are used to illustrate the road in this report, are inspired by the layout of the figures used by Rijkswaterstaat (Rijkswaterstaat, 2017).

Another problem was the bascule bridge to cross the river the Oude Rijn. This movable bridge resulted to a lot of congestion problems when it was opened. The combination of the movable bridge and the amount of strips resulted in the fact that the highway between Hoogmade and Zoeterwoude-Dorp stood in the top three of the File top 50 of the VID every year between 2006 and 2012 (VID, 2006-2012), as also stated in the introduction.

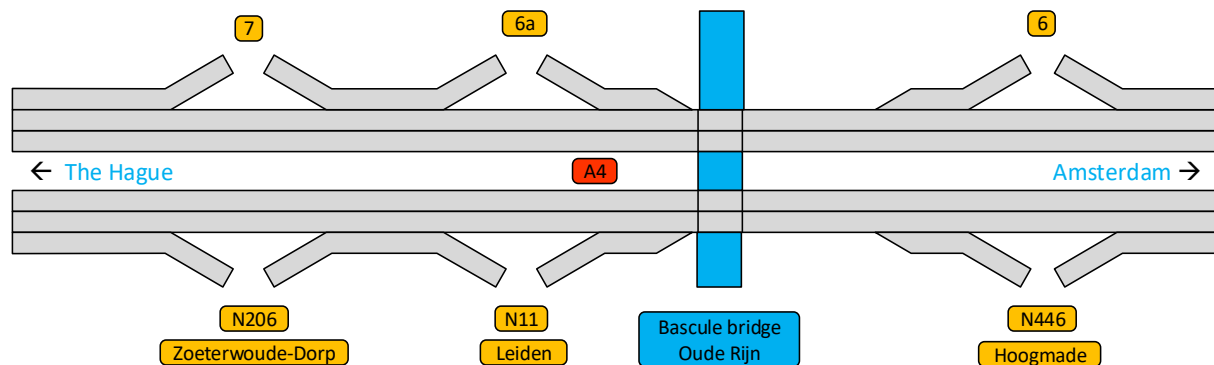


Figure 2: Layout of the A4 before 2014

1.2 SITUATION AFTER 2014

In 2009 the decision was made to widen and deepen the highway around Leiden. This to reduce the congestion problems on the track from Hoogmade to Zoeterwoude-Dorp. The moveable bascule bridge was replaced by the Limesaqueduct. The name refers to the old border of the Roman Empire (Limes = border). The new aqueduct meant that ships were from now on able to sail on the Oude Rijn without disturbing the traffic on the A4. The aqueduct is connected to the new deepened 1.4 km long part of the highway. The highway is deepened to decrease the noise disturbance of the highway for the citizens of Leiden, Leiderdorp and Zoeterwoude. The deepened highway consists out of two open tunnels, one in each direction. To make future highway expanding possible there is enough space for five strips in both tunnels.

As said before not only the bascule bridge was removed but the road was also widened. The A4 is widened to three strips in both directions from Hoogmade until the end of the Limesaqueduct. This is done to remove the bottleneck at exit six in the direction of The Hague and the bottleneck at exit seven in the direction of Amsterdam. Highway exit six is replaced a few hundred meters to the north to create enough space for the extra strips. The new fly-over has room for two times four strips, which means that there is enough space for a further expansion of the highway in the future.

The highway is expanded in a different manner from the end of the Limesaqueduct until exit seven. In this track the highway is split in two lanes with a varying amount of strips over the length of the track in each direction. This is done to separate the local traffic to the city of Leiden from the long distance traffic in the directions of The Hague and Amsterdam. In this way the number of bottlenecks on the

long distance lane is reduced from four to one in each direction. Downstream exit seven in the direction of The Hague and downstream exit 6 in the direction of Amsterdam the road continues as it was before with two times three strips. The information about the new layout of the road can all be read in the track decision article 'Tracebesluit A4 Burgerveen-Leiden 2009, gedeelte Leiderdorp-Leiden' (Rijkswaterstaat, 2009). The exact map of the new situation can be found in figure 3 below.

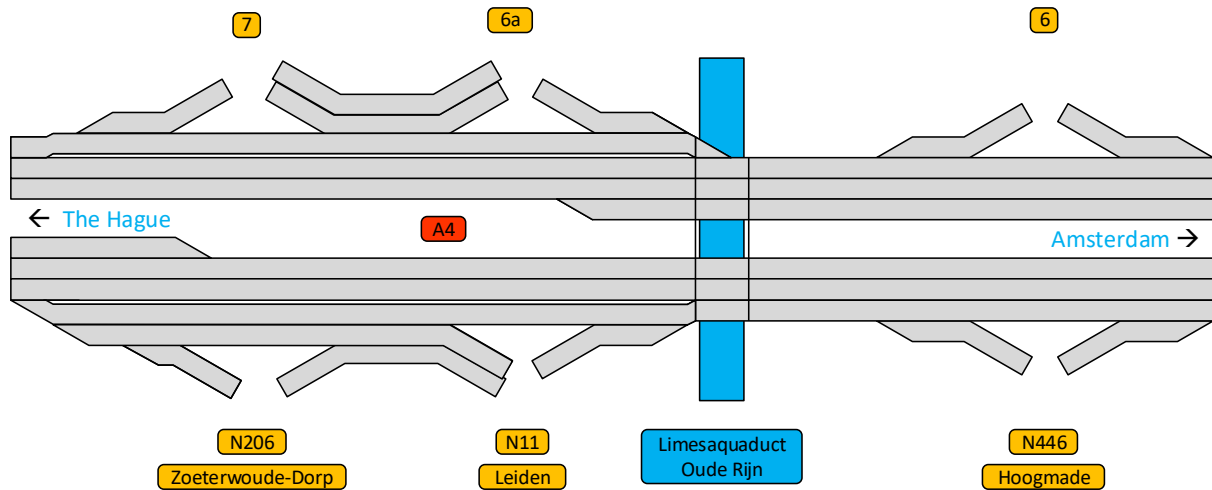


Figure 3: Layout of the A4 after 2014

The goal of the transformation of the A4 around the city of Leiden was of course to reduce the congestion problems on the highway. Eventually the project seemed to be a success as stated in the introduction. In the years 2012 and 2013 the highway existed out of three narrow strips in both directions, which were lead through the west tunnel of the Limesaqueduct. The highway had one extra strip compared to the situation before and was not split up in local- and long-distance traffic yet. The extra strip on both sides of the road resulted in a decrease in the total level of congestion on the track. In 2013 and 2014 the highway fell even out of the congestion top 50 of the VID. According to the data gathered by the VID, in 2013 the level of congestion was 43.537 kmmin in the direction to Amsterdam while this was 120.488 kmmin in the year of 2011 just before the reconstruction of the road (VID, 2011-2014).

Out of these numbers can be concluded that initially the project was a success, but two years after completion, the opposite can be stated. The total level of congestion in 2016 was with 222.787 kmmin in the direction of Amsterdam more than five times as high as in 2013. The situation was even worse than before the start of the project and the highway can be found back at the top of the congestion top 50 again (VID, 2016).

As stated in the introduction, the main research question is: What is the main cause of the current traffic congestion problem on the A4 between Hoogmade and Zoeterwoude-Dorp? To answer this question, five sub-questions are answered:

- Where can the bottlenecks on the layout of the road be found?
- What is the capacity and the demand at these bottlenecks?
- What percentage of the level of congestion is caused by accidents on the road?
- What percentage of the total level of congestion is caused by other factors, such as road maintenance and road blockage?

The methods on how the answers on the four sub-questions were estimated are given in this chapter. To start the methodology to determine the total level of congestion according to the data in both directions is explained (3.1). In second methodology behind to bottleneck analysis is explained (3.3). This analysis is done to see where the most congestions originate and to answer the first sub-question. In third the methodology behind the peak hour analysis is explained (3.2). The answer on the second sub-question is estimated by doing a capacity and demand analysis (3.4), the answer on the third question by doing an accidents analysis (3.4) and the answer on the fourth and last question by analyzing the road maintenance (3.5) and road blockage data (3.5).

All the sub-questions are answered by doing data analyses in programming software Python. The used congestion data is coming from Rijkswaterstaat, all the other used data is coming from the Nationale Database Wegverkeergegevens (NDW).

3.1 METHODOLOGY DETERMINING TOTAL LEVEL OF CONGESTION

In this paragraph the methodology used to determine the total level of congestion according to the used data in both directions is explained. This total level of congestion of 2016 in both directions was needed to form a good image of the problem and to be able to determine the percentages of level of congestion caused by accidents, road maintenance and road blockage later. The total level of congestion is determined by doing a data analysis.

The following congestion data is used:

- The data of all congestions on the A4 in both directions with a starting point between hectometer 28.0 and 36.5 over whole 2016. A sample of this data can be found in appendix A in the direction of The Hague.

This data contains the starting month, day, hour and minute as well as the ending hour and minute of all congestions on the track. It also contains the starting and ending points of the congestions. With the starting point defined as the most downstream hectometer and the ending point as the most upstream hectometer of the congestion. Furthermore the data contains the level of congestion of each congestion. The data set is split up in one set for the direction of The Hague and one for the direction of Amsterdam. These two data sets are meant when later in the report is referred to congestion data.

The total level of congestion is found by performing a simple loop which takes the total sum of the levels of congestion of each single congestion in Python. The exact Python script for the total level of congestion in 2016 in the direction of The Hague is explained in appendix I, an overview of the Python script used to determine the total level of congestion in the direction of Amsterdam can be found in appendix II.

3.2 METHODOLOGY BOTTLENECK ANALYSIS

In this paragraph the methodology of the bottleneck analysis is explained. It is important to know where the bottlenecks are exactly located to perform a congestion analysis. After the bottleneck analysis it is known where congestions originate and which sections of the road has to be investigated further.

The data used for the bottleneck analysis is the same congestion data as defined in paragraph 3.1. The bottleneck analysis is executed by using a loop in Python again. This loop counts the amount of congestions which started at every hectometer in one direction. The exact Python script for the direction of The Hague is explained in appendix III, an overview of the Python script used for the direction of Amsterdam can be found in appendix IV.

To get a better image which bottlenecks cause the biggest problems, the total level of congestion of all the congestions starting per bottleneck is also plotted. This is done by using the same data and a slightly different Python script. The biggest difference in the used loop is that instead of just counting, the level of congestion of a certain congestion is added to the total sum of the level of congestion of all the previous congestions which started at the same hectometer. The exact Python script for the direction of The Hague is explained in appendix V, an overview of the Python script used for the direction of Amsterdam can be found in appendix VI.

The next step and last of the bottleneck analysis was to compare the results of both analyses with each other and compare them with the layout of the highway. In this way the bottlenecks on the actual road were found.

3.3 METHODOLOGY PEAK HOUR ANALYSIS

In this paragraph the methodology of the peak hour analysis is explained. A peak hour analysis is executed to determine when the most congestion problems occur. Out of the results can be concluded if congestion problems are evenly distributed over the day or if they are mainly concentrated in peak hours. It can also be concluded out of the results if the congestion problems in the morning occur in the same or a different direction than in evening.

The exact same two congestion data sets as in the two previous chapter have been used again for the data analysis. The analysis is executed in Python again. The script which is used is very similar to the script which is used in the bottleneck analysis. In fact only the length of the array, from the length of the amount of hectometers on the track to the amount of hours in one day, and the loaded data, from the starting points of the congestions to the starting hours of the congestions, are changed. The explanation and overview of the Python script can be found respectively in appendix VII and VIII.

3.4 METHODOLOGY DEMAND AND CAPACITY ANALYSIS

To estimate the answer on the third sub-question a capacity analysis and a demand analysis were needed. The methodology behind these analyses is explained in this chapter. In the demand analysis the amount of vehicles that want to use the highway is determined (3.4.1). In the capacity analysis the maximum amount of vehicles that should be able to use the road without forming congestion is determined (3.4.2). The results of these two analyses are combined to control whether the congestion problems starting in the bottlenecks are due to capacity problems or not (3.4.3).

The data analyses are executed according to the idea of the vertical queuing model. This model states that the inflow of a congestion is equal to the demand and the outflow of a traffic is equal to the capacity of the road. This is further explained in figure 4 and figure 5, both coming from the reader Traffic flow theory and simulation (Knoop, 2017).

In figure 4 can be seen that the demand of a road differs a lot over the time. A congestion develops when the demand of a road becomes higher than the capacity of the road. After this moment, which can be seen in the first crossing of the demand line and the capacity line in figure 4, the outflow of the congestion is equal to the capacity, as this is the maximum amount of vehicles that the road can let through. This can also be seen in figure 5, the cumulative curve. The blue line is the cumulative outflow of the section over the measured time. The red line is the cumulative inflow into the section over the measured time. It can be seen that the cumulative outflow becomes linear in figure 5 when the demand becomes higher than the capacity in figure 4. This means that the outflow is equal to the capacity. The congestion starts to dissolve when the demand of the road becomes lower than the capacity again. This happens when the demand line crosses the capacity line for the second time in figure 4. The outflow of the congestion is equal to the capacity of the road until the complete congestion has been dissolved. This can also be seen in figure 5, the blue cumulative outflow line stays linear when the demand line in figure 4 is already below the capacity line. In these figure this happens around 175 seconds. Thus the outflow of a congestion is equal to the capacity of the road during the complete congestion. This means that the capacity of the road can be measured as the average of the intensity (vehicles per hour) at a point just downstream of the starting point of the congestion, during the whole congestion. This period of measuring is called the aggregation period.

This is called the discharge capacity. Another option for measuring the capacity is the free flow capacity. This is the intensity on the road measured just before the moment that a congestion developed. The free flow capacity is always 10-15% higher than the discharge capacity, as the capacity of a road reduces during a congestion (Heikoop, 2015). In this report the discharge capacity is used. This is done because the discharge capacity takes a lot more data into account, which will improve the precision of the estimation.

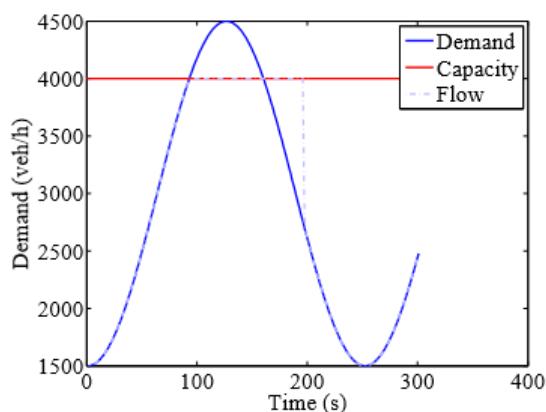


Figure 4 Capacity and demand

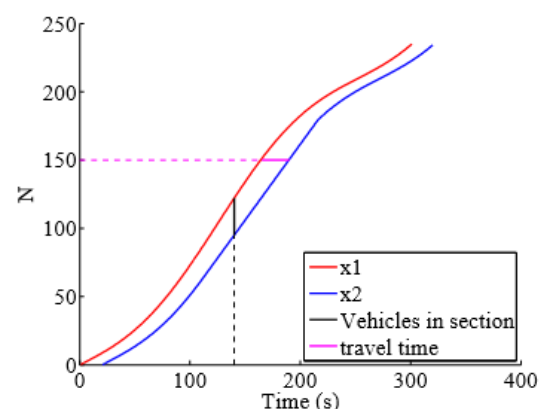


Figure 5 Cumulative curves

A same sort of measurement can be made for the demand of the road. The demand is the amount of cars which want to use the road. In the previous paragraph is stated that a congestion develops when the demand of a road is higher than the capacity of this road. This means that the highest demand of the road when the road does not have enough capacity occurs during a congestion. This demand can be measured as the intensity at a measuring point just upstream of the upstream end of the congestion. It is the inflow of the congestion.

As stated before the capacity of a road can be measured during the entire congestion as it does not change too much during the period of the congestion. This isn't the case for the demand of a road as the demand changes a lot during a congestion. The demand is equal to the capacity of the road when the congestions starts developing, then rises to a certain top point and is equal to the capacity again when the congestion starts to dissolve again. Which means that for a demand estimation only the maximum inflow intensity is interesting.

3.4.1 METHODOLOGY DEMAND ANALYSIS

As stated in the previous paragraph, the demand is the number of vehicles which want to pass a certain point of the road per hour (vehicles/hour). To find a good representative value a data analysis is executed. The demand of the road is determined in all bottlenecks which are determined according to the method described in paragraph 3.1. For each bottleneck intensity data is used of the most upstream measuring point at which the layout of the road is still the same as just upstream of the bottleneck. These measuring points can be seen in appendix H.

This intensity data contains an indication of the strip at which the measurement was done, the time and date of the measurement and the measured intensity in vehicles per hour. A sample of the used intensity data can be found in appendix B. All the used data intensity data for the demand and capacity analysis is of the same form which is why only one sample is included. Furthermore for the congestion data as described in paragraph 3.1 is used.

The data analysis is executed by performing a loop in Python again. The input for the loop are the intensity upstream of the bottleneck and the congestion data in 2016 as stated above. The loop takes the maximum measured inflow intensity per congestion. After the loop it takes the average of all these maximum intensities to find the average maximum demand during congestion in a bottleneck. The exact Python script for January 2016 at hectometer 33.5 in the direction of The Hague is explained in appendix IX. An overview of the Python script for January 2016 in at hectometer 34.5 in the direction of Amsterdam can be found in appendix X. This demand analysis is executed for all months in 2016 for every bottleneck.

3.4.2 METHODOLOGY CAPACITY ANALYSIS

The second step to answer the third sub-question is to do a capacity analysis of the highway. After the capacity analysis the maximum number of vehicles that should be able to use the road in a certain timespan without forming congestion is known. In this report the discharge capacity is determined as stated in paragraph 3.4. The capacity of the road is also determined in all bottlenecks which are determined according to the method described in paragraph 3.1. For each bottleneck intensity data is used of the first measuring point just downstream of the bottleneck. These measuring points can be seen in appendix I. The aggregation periods of the demand and capacity analysis are the same, namely the duration of each congestion. Furthermore the congestion data described in paragraph 3.1 is used again.

The capacity of the bottlenecks is determined by using a Python loop again. The Python script used to determine the capacity is also almost the same as the Python script used to determine the demand. The two main differences are the input, intensity measured downstream instead of upstream of the congestion, and the capacity is measured as the average of all the intensities during congestion. This can be done as the capacity stays equal during the congestion and no maximum needs to be found as is explained in the first paragraph of chapter 3.3. The exact Python script for January 2016 at hectometer 33.5 in the direction of The Hague is explained in appendix XI. An overview of the Python script for January 2016 in at hectometer 34.5 in the direction of Amsterdam can be found respectively in appendix XII. This demand analysis is executed for all months in 2016 for every bottleneck.

3.4.3 METHODOLOGY COMPARISON DEMAND AND CAPACITY IN EACH BOTTLENECK

When the demand and the capacity per bottleneck were analyzed, the average of the capacity and demand per bottleneck over all months of 2016 were taken. Thereafter the difference between the average demand and average capacity is calculated for every bottleneck by subtracting the average capacity from the average demand.

Another option to compare the demand and the capacity in the bottleneck is by calculating the I/C-ratios. This is the demand (intensity) divided by the capacity. The I/C-ratio of a road states a lot about the situation on the road. Grontmij executed a big number of simulations in the highway simulation software Fosim to determine the chances on forming of congestion within 30 minutes at different I/C-ratios (Heikoop, 2015). The results of these simulations can be seen in table 1. The I/C-ratios of the bottlenecks can be compared with the values stated in this table to determine the status of the at the different bottlenecks.

I/C-ratio	Description	Chance on forming of congestion within 30 minutes
<i>Smaller than 0.3</i>	Smooth traffic flow.	0 %
<i>0.3-0.8</i>	Smooth traffic flow without noteworthy forming congestion apart from accidents.	<< 1 %
<i>0.8-0.9</i>	Moderate traffic flow with structural forming of congestion. Traffic flow sensitive for small disturbances.	< 20 %
<i>0.9-1.0</i>	Bad traffic flow. Structural forming of congestion. Small disturbances in the traffic lead immediately to congestion.	20 -100 %
<i>Bigger than 1.0</i>	Really bad traffic flow. Structural and daily forming of congestion with stationary flow.	100 %

Table 1: I/C-ratio and its impact on the forming of congestion

3.5 METHODOLOGY ANALYSIS OTHER CONGESTION CAUSES

When congestion isn't caused by a capacity problem, there must be another reason for the congestion to occur. In this chapter three reasons are investigated and their part in the problem is estimated. These reasons are: congestion caused by accidents (3.5.1), congestion caused by road maintenance work (3.5.2) and congestion caused by road blockage (3.5.3).

3.5.1 METHODOLOGY ACCIDENT ANALYSIS

In this paragraph it will be explained how the level of congestion caused by accidents is estimated. After this was estimated the percentage of the total level of congestion over 2016 caused by accidents is determined. After this number was known an indication is made of how big the influence of the accidents is on the congestion on the road.

Data of all the accidents in both direction between Hoogmade and Zoeterwoude-Dorp is used to determine this percentage. This data consists of the date and time of the accidents per direction and a lot of metadata. Unfortunately the data doesn't contain the locations of the accidents. Furthermore the congestion data described in paragraph 3.1 is used again.

The data analysis is executed by building a loop in Python again. The loop which is built searches for a congestion which developed in timespan of an hour after an accident in the same direction as the accident. When this is the case this congestion is counted as caused by an accident. After the total level of congestion caused by accidents was determined, the level of congestion caused by accident is divided by the total level of congestion in the same direction, to determine the percentage of the total level of congestion which is caused by accidents. The exact Python script in the direction of The Hague is explained in appendix XIII. An overview of the Python script for in the direction of Amsterdam can be found in appendix XIV. After the level of congestion which is caused by accidents was known it is compared to the overall data of the Netherlands, to conclude whether the road is extraordinary dangerous or not.

3.5.2 METHODOLOGY ROAD MAINTENANCE ANALYSIS

This paragraph will explain how the level of congestion caused by road maintenance is estimated. After this was estimated the percentage of the total amount of congestion over 2016 caused by road maintenance is determined. After this number is known an indication is made of how big the influence of road maintenance in 2016 was on the congestion on the road. This analyzation is very similar to the accident analysis described in paragraph 3.5.1.

Data of all the road maintenance works in both direction between Hoogmade and Zoeterwoude-Dorp is used to determine this percentage. This data consists of the starting and ending date and time of the road maintenance works per direction, the reason of the executed work and a lot of metadata. A sample of this data can be seen in appendix D. Furthermore the congestion data described in paragraph 3.1 is used again.

The data analysis is executed in Python again. The Python script which is used is very similar to Python the Python script which is used during the accident analysis. The only difference are the input and the fact that road maintenance work most of the time has an exact ending time. This makes it possible to search for congestions starting between the starting and ending time of a road maintenance work. The exact Python script for the congestion caused by road maintenance in the direction of The Hague is explained in appendix XIII. An overview of the Python script in the direction of Amsterdam can be found in appendix XIV.

3.5.3 METHODOLOGY ROAD BLOCKAGE ANALYSIS

The method to estimate the level of congestion caused by road blockage is exactly the same as the method to estimate the level of congestion caused by road maintenance. The only difference is the input data. Data of road blockage instead of data of road maintenance. Data of all the road occurred road blockages in both direction between Hoogmade and Zoeterwoude-Dorp is used to determine this percentage of the total level of congestion caused by road blockage. This data consists of the starting and ending date and time of the road blockages per direction, the reason of the blockage and a lot of metadata. The kind of road blockages which are included in the file are: road blockage by dropped loads, stranded cars and by presence of animals. A sample of this data can be seen in appendix E. Furthermore the congestion data described in paragraph 3.1 is used again. The Python script which is used for this analysis is exactly the same as described in paragraph 3.5.2.

4 RESULTS

4.1 RESULTS BOTTLENECK ANALYSIS

The results of the bottleneck analysis are shown in the graphs below.

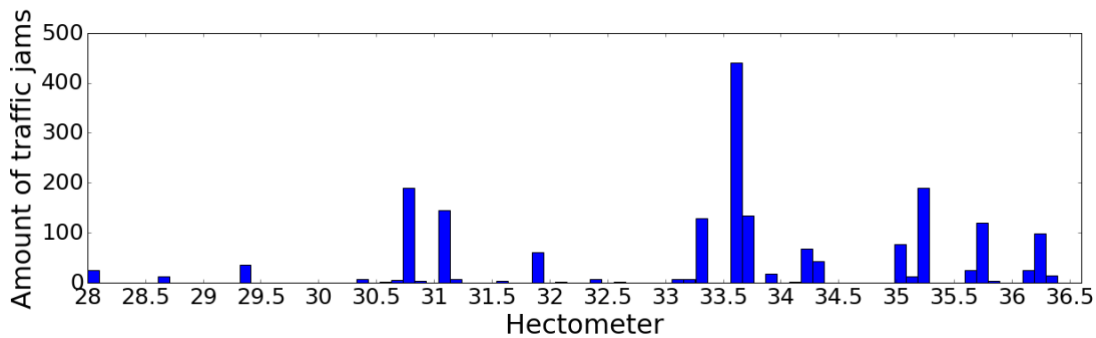


Figure 6: Number of congestions per hectometer over 2016 in the direction of The Hague

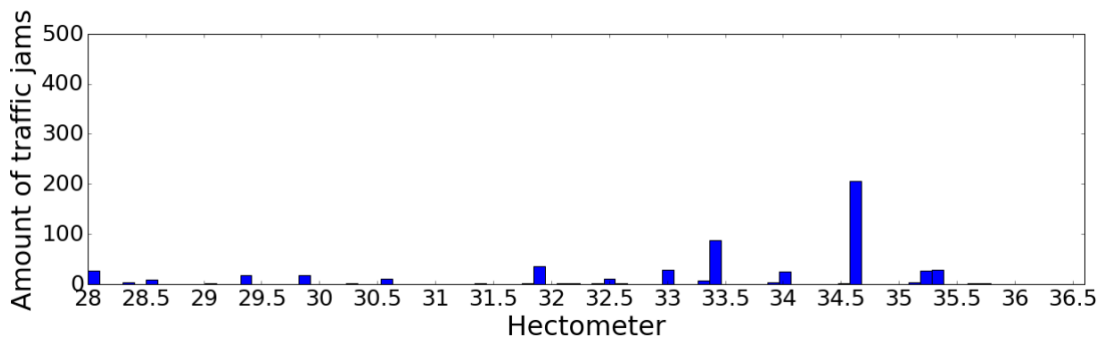


Figure 7: Number of congestions per hectometer over 2016 in the direction of Amsterdam

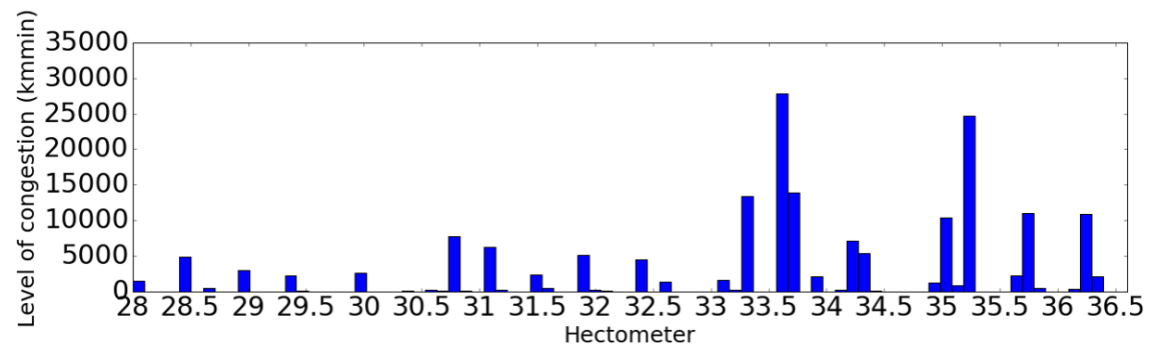


Figure 8: Total level of congestion of all congestions per hectometer over 2016 in the direction of The Hague

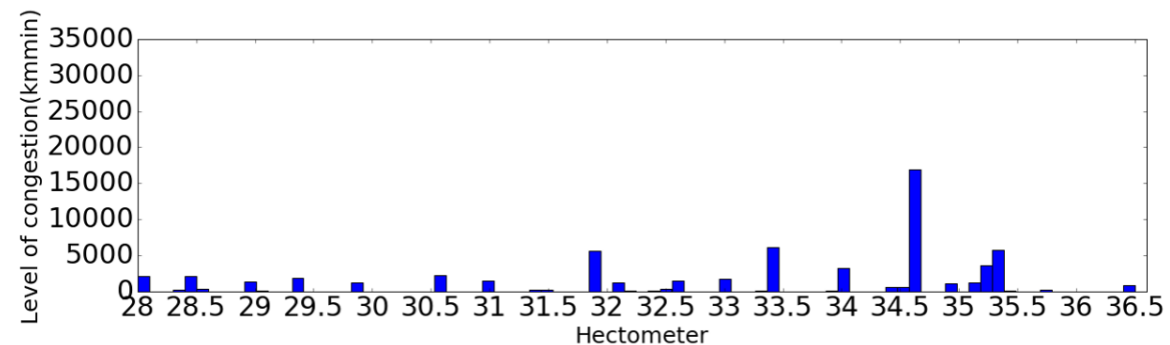


Figure 9: Total level of congestion of all congestions per hectometer over 2016 in the direction Amsterdam

It can be seen from figure 6 that the most import bottlenecks in the direction of The Hague are positioned at hectometers 30.7, 33.5 and 35.1. In figure 8 can be seen that there are also bottlenecks positioned at hectometers 35.6 and 36.1. In figure 7 and 9 can be seen that the most important bottlenecks in the direction of Amsterdam are situated at hectometers 31.8, 33.3, 34.5 and a combination of 35.1 and 35.2. All the bottlenecks are indicated with red lines in figure 10. The exact hectometers shown in the figure come from the Tracékaarten, Ontwerp-tracébesluit A4 Burgerveen-Leiden (Directoraat-Generaal Rijkswaterstaat, 2009).

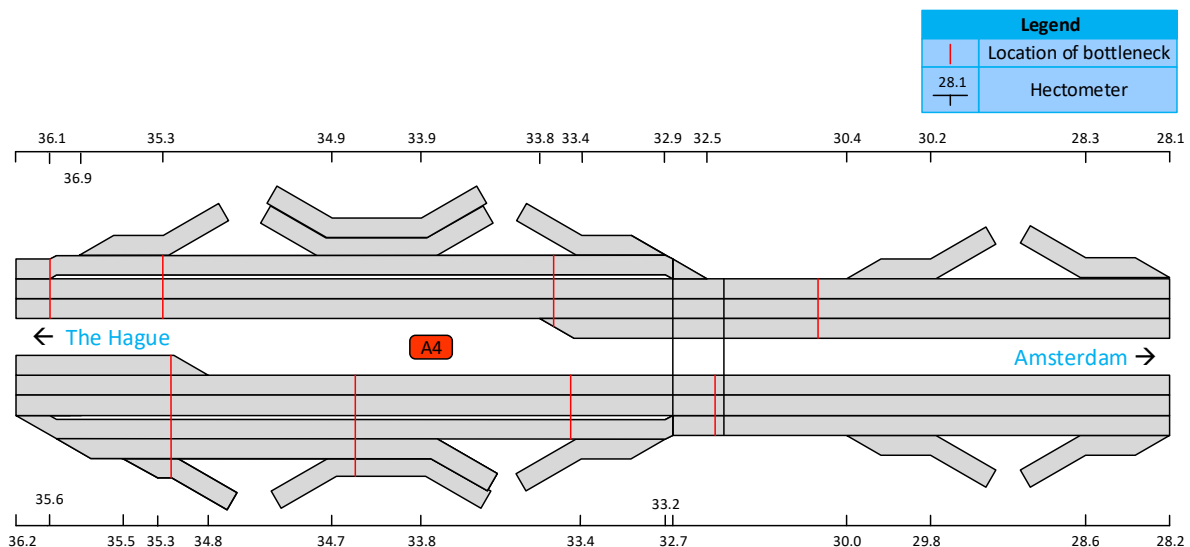


Figure 10: Locations where the most congestion problems start

The road is split up in a long-distance and a local traffic lane as stated before. This means that a bottleneck can be located on one of these lanes while the traffic flows smoothly on the other. The layout of the road has been investigated further to determine at which strips or lanes all the bottlenecks are located.

A logical location for a bottleneck is point where the traffic has to merge from a flow on multiple strips to a flow on less strips. This is the case on both long-distance lanes, at hectometer 33.5 in the direction of The Hague and at the hectometers 35.1 and 35.2 in the direction of Amsterdam. Out of this can be concluded that these bottlenecks are located at the long-distance lanes.

Another logical location for a bottleneck is a point where traffic from a highway entry has to merge into the flow on the highway. This is the case on both local traffic lanes, at hectometer 35.1 and 35.6 in the direction of The Hague and at hectometer 33.3 in the direction of Amsterdam. The both bottlenecks in the direction of The Hague are seen as one because the layout of the road does not change between them. The bottleneck at hectometer location 30.7 in the direction of The Hague occurs probably due to the same and is of course located on all three strips.

The bottleneck at hectometer 34.5 in the direction of The Hague is located on the local traffic lane as the layout of the long-distance traffic lane does not change at this section. This bottleneck is probably caused by all the drivers changing strips at this section. In the reader 'Capaciteitswaarden Infrastructuur Snelwegen' (Heikoop, 2015) can be read that strip changes influence the capacity of a road. The last two bottlenecks, hectometers 36.1 and 31.8 in the direction of respectively The Hague and Amsterdam are of course located on all three strips.

The bottleneck at 36.1 in the direction of The Hague is also probably caused by drivers changing their strip. The bottleneck at 31.8 in the direction of Amsterdam can be explained by the fact that this bottleneck is located just upstream of the start of the ascend out of the Limesaqueduct. The report 'Car-following Behavior at sags and its impacts on Traffic Flow' states that the capacity of a road indeed does decrease on an uphill section (Ros, Knoop, van Arem, & Hoogendoorn, 2012).

The eight bottlenecks described in this chapter are investigated further in this report. All the bottlenecks as described above are marked with a red line in figure 11.

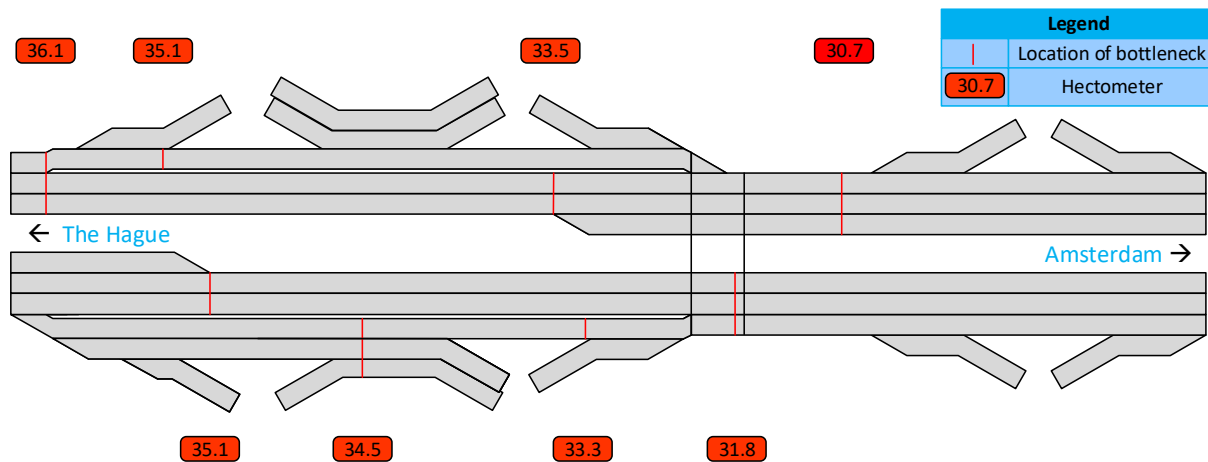


Figure 11: Locations of the bottlenecks

4.2 RESULTS PEAK HOUR ANALYSIS

The results of the peak hour analysis are shown in two figures in this paragraph. Figure 12 shows the amount of congestions which started in a certain hour in the direction of The Hague. Figure 13 shows the same thing in the direction of Amsterdam.

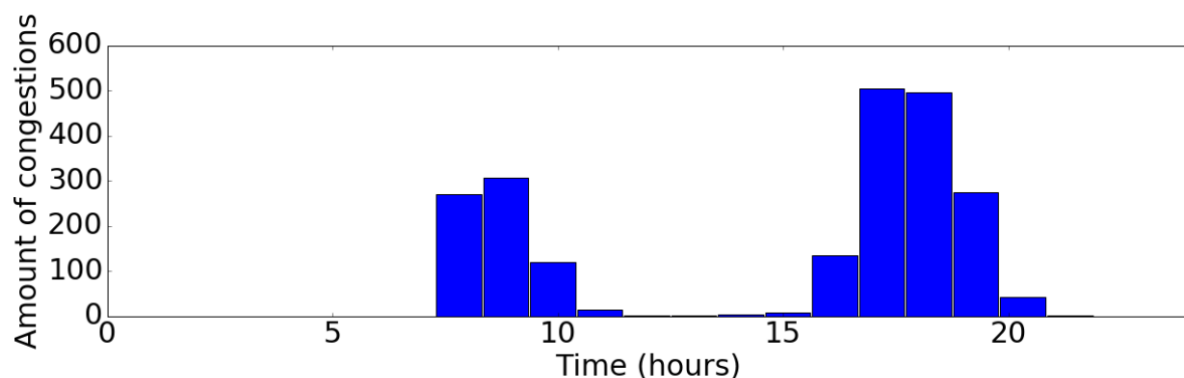


Figure 12: Number of congestions starting per hour over 2016 in the direction of The Hague

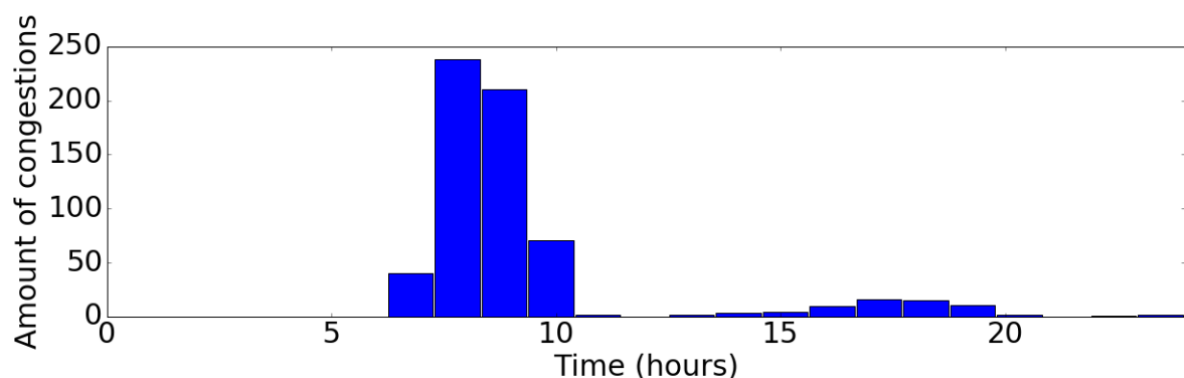


Figure 13: Number of congestions starting per hour over 2016 in the direction of Amsterdam

It can be seen that the congestion problems are mainly concentrated in the peak hours. In the direction of The Hague more congestions develop in the evening peak hours than in the morning peak hours. In the direction Amsterdam almost all congestions develop in the morning peak hours.

4.3 RESULTS DEMAND AND CAPACITY ANALYSIS

In this paragraph the results of the demand and capacity analysis are shown. These are the results of the methodology described in chapter 3.3. The complete results per month of each bottleneck of demand and capacity analysis can be found in respectively appendix F and G.

Location	Average demand	Average capacity	Difference	I/C-ratio
30.7	8066	5029	3037	1.60
33.5	5763	3670	2093	1.57
35.1	5022	1921	3101	2.63
36.1	7305	4780	2525	1.53

Table 2: Comparison demand and capacity on the bottlenecks in the direction of The Hague

Location	Average demand	Average capacity	Difference	I/C-ratio
31.8	6969	4812	2157	1.45
33.3	3856	1572	2284	2.45
34.5	6883	1665	5218	4.13
35.1	5132	3179	1953	1.61

Table 3: Comparison demand and capacity on the bottlenecks in the direction of Amsterdam

It can be seen that the average demand is higher at each bottleneck than the average capacity. The differences between demand and capacity are all very comparable except for the difference at bottleneck 34.5 which is extraordinary high. This difference could be used to determine how many extra strips are needed. For all bottlenecks the I/C-ratio is far above 1, which means that there is a 100 % chance on the forming of a congestion within a period of 30 minutes as can be seen in table 1 in chapter 3.4.3.

4.4 RESULTS ANALYSIS OF OTHER CAUSES

4.4.1 CONGESTION CAUSED BY ACCIDENTS

In this paragraph the results of the accident analysis are shown. The results are coming from the methodology described in chapter 3.5.1. In first, in table 4, the percentages of the total level of congestion caused by accidents on the A4 between Hoogmade and Zoeterwoude-Dorp are shown.

Direction	The Hague	Amsterdam
<i>Level of congestion caused by accidents (kmmin)</i>	20 633.6	14 507.63
<i>Total level of congestion (kmmin)</i>	179 761.5	64 475.3
<i>Percentage of congestion caused by accidents</i>	11.5 %	22.5 %

Table 4: Results accident analysis

This data can now be compared to the data of the VID which state the overall percentage of congestion caused by traffic accidents on the main road network in the Netherlands. (VID, 2017). This data can be seen in table 5 on the next page.

	2016	2011	2006
<i>Level of congestion caused by accidents in the Netherlands (kmmin)</i>	1 986 167	1 508 641	1 578 469
<i>Total level of congestion in the Netherlands (kmmin)</i>	10 931 110	10 437 738	14 573 500
<i>Percentage of congestion caused by accidents</i>	18.2 %	14.5 %	10.8%

Table 5: Congestion caused by accidents in the Netherlands

It can be seen that the percentages on the A4 between Hoogmade and Zoeterwoude-Dorp are around average.

4.4.2 CONGESTION CAUSED BY ROAD MAINTENANCE

The results of the road maintenance data analysis can be seen in table 10. These are the results coming from the methodology described in chapter 3.5.2. It can be seen that the influence of road maintenance on the forming congestion was very small in 2016.

<i>Direction</i>	The Hague	Amsterdam
<i>Level of congestion caused by road works (kmmin)</i>	1693.5	778.5
<i>Total level of congestion (kmmin)</i>	179 761.5	64 475.3
<i>Percentage of congestion caused by road works</i>	0.94 %	1.2 %

Table 6: Congestion caused by road maintenance work

4.3.3 CONGESTION CAUSED BY ROAD BLOCKAGE

There were no congestion caused by blockage of the road in 2016. This can be seen in table 11. These are the results coming from the methodology described in chapter 3.5.3.

<i>Direction</i>	The Hague	Amsterdam
<i>Level of congestion caused by road blockage (kmmin)</i>	0	0
<i>Total level of congestion (kmmin)</i>	179 761.5	64 475.3
<i>Percentage of congestion caused by road blockage</i>	0 %	0 %

Table 7: Congestion caused by blockage of the road

Out of the results of the bottleneck analysis can be concluded that there are eight bottlenecks in the layout of the A4 between Hoogmade and Zoeterwoude-Dorp. These bottlenecks are located at hectometers 30.7, 33.5, 35.1 and 36.1 in the direction of The Hague and at hectometers 31.8, 33.3, 34.5 and a combination of 35.1 and 35.2 in the direction of Amsterdam. In all of these eight bottlenecks the average demand during peak hours is higher than the average capacity of the bottlenecks. All the bottlenecks have an I/C-ratio of above 1. Which means that there is, with the average demand during peak hours, a 100 % chance on the forming of a congestion within a period of 30 minutes. This answers the first two sub-questions of the research.

Out of the results of the other analyses can be concluded that the influence of accidents on the total level of congestion on the road is on the average level of influence of accidents on the total level of congestion on the complete main road network in the Netherlands. Thus the influence of accidents is not extraordinary high. The influence of road maintenance works is negligible small. The same applies to congestion caused by road blockage. Which answers the second two research questions.

The main research question as stated in the introduction was: What is the main cause of the current congestion problems on the A4 between Hoogmade and Zoeterwoude-Dorp? Out of all these results above it can be concluded that the congestion problems on the A4 around Leiden are mainly caused by the under capacity of the eight bottlenecks.

The results and conclusions are based on the traffic intensities and traffic data in 2016. This means that it can't be concluded out of the results that the reconstruction of the A4 between Hoogmade and Zoeterwoude-Dorp is a failure, as the demands in the years before 2016 are not investigated. It could be that the total level of congestion on the track before and during the construction was lower due to a lower demand at this time. Furthermore it could be that one of the bottlenecks is caused by a lot traffic accidents which happen at this certain bottleneck. This isn't investigated further as the locations of the accidents is not known.

In future research these two could investigated further. Furthermore the bottlenecks could be investigated further and an improved design could be made for the future. To do this the recommendations in chapter 6 could be used as a start.

6 RECOMMENDATIONS

As is stated in the conclusion the congestion problems mainly occur due to the higher demand in the bottlenecks than the capacity in these bottlenecks. This means that there are two options to reduce the congestion problems, the first is to increase the capacity of the road and the second is to decrease the demand of the road. Some options which could potentially fulfil these two are described in this chapter. These options are all just recommendations and should be investigated further in future research before applied to the road.

6.1 INCREASE OF CAPACITY

6.1.1 EXTRA STRIPS

The simplest option to increase the capacity of a road is of course to increase the amount of strips, but it is also a very expensive and profound one. As stated before in chapter two, space for extra strips is made available in the reconstruction of the road. An option would be to add an extra lane on the long-distance traffic lane in both directions. This will immediately scrap two bottlenecks, namely the bottleneck at hectometer 33.5 in the direction of The Hague and the bottleneck at hectometer 35.1 in the direction of Amsterdam. A consequence will be that new bottlenecks are formed at most downstream point until which the extra strip will be extended. This means that increasing the capacity will only have a positive effect on reducing the total level of congestion when extra strip is added along the complete route between Burgerveen and Leidschendam. Which are respectively the northern and southern point in the road at which the layout changes completely (OpenStreetMap, 2017). As stated in the introduction, Rijkswaterstaat is already planning to add extra strips like described in this paragraph (Rijkswaterstaat, 2017). But these plans aren't final yet.

6.1.2 CHANGE THE ROAD LAYOUT

Another profound option is to change the layout of the road. As stated before, the capacity of a road decreases when the number of strip-changes by drivers increases (Heikoop, 2015). The number of times that a driver has to change their strip is influenced by the layout of the road. This is for example the case at bottleneck 34.5, the bottleneck with the highest I/C-ratio.

At the moment the inflowing traffic from exit 7 has to merge from the right side of the three strips. While the traffic coming from The Hague enters the road at the left two strips. It can be assumed that all the traffic coming from The Hague will leave the highway at exit 6a, as otherwise there would have been no reason for these drivers to choose the local-traffic lane. This current situation is illustrated in the left image of figure. The layout of the road could be changed to reduce the number of strip-changes and increase the capacity of the bottleneck. This can be done by letting the traffic entering the highway from exit 7 merge at the left side of the three strips. This would mean that all the drivers coming from The Hague won't have to change their strip. Which will lead an increase of the capacity of the road. The possible needed strip changes are indicated with the arrows figure 13.



Figure 13: Current situation and recommended future situation of bottleneck 34.5

6.2 DECREASE OF DEMAND

6.2.1 RAMP METERING SYSTEMS

It can be seen in the bottleneck analysis and be read in the conclusion that a big part of the congestions start at bottlenecks at which traffic enters the highway. This traffic flow entering the highway causes a too high demand of the road. To prevent this the traffic flow entering the highway can be controlled and adjusted to the capacity using ramp metering systems. Ramp metering systems are traffic lights at the access roads to the highway, which led a certain amount of cars pass in a certain timespan. A ramp metering system could increase the capacity of a road with a factor of 1.05 (Heikoop, 2015).

6.2.2 AVOID PEAK HOUR COMPENSATION

As can be read in the results of the peak hour analysis and the conclusion the congestion mainly occur due to capacity problems in the peak hours. An option to try to reduce the demand in these peak hours is to reward people who avoid the peak hours and travel outside of them. In this way a more even distribution of the total traffic flow over the day can be created. Regulations like this have been executed before. For example in the project slimopdeA27 (Ijberg Holding BV, 2017), in which car drivers were rewarded when they avoided the road in peak hours during a period of road maintenance work.

REFERENCES

- Any to Many. (2014). *Opendag Limesaquaduct*. Alphen aan den Rijn. Retrieved October 18, 2017, from www.anytomany.nl
- Directoraat-Generaal Rijkswaterstaat. (2009). *Ontwerp-tracébesluit A4 Burgerveen-Leiden*. Ministerie van Verkeer en Waterstaat, Rotterdam.
- Heikoop, i. H. (2015). *Capaciteitswaarden Infrastructuur Autosnelwegen*. Grontmij. Delft: Dienst Water, Verkeer en Leefomgeving.
- Ijberg Holding BV. (2017). *Slim op de A27*. Amsterdam. Retrieved October 18, 2017, from <https://slimopdea27.nl/>
- Knoop, V. (2017). *Reader Traffic Flow Theory and Simulation*. TU Delft, Delft.
- NDW. (2016). *Blokkades*. Situatieberichten, NDW, Utrecht.
- NDW. (2016). *Gebeurtenis*. Situatieberichten, NDW, Utrecht.
- NDW. (2016). *Intensiteit*. Export aanvraag, NDW, Utrecht.
- NDW. (2016). *Werkzaamheden*. Situatieberichten, NDW, Utrecht.
- OpenStreetMap. (2017). *Kaarten*. Rijkswaterstaat, Rijswijk. Retrieved September 29, 2017, from Rijkswaterstaat: <https://www.rijkswaterstaat.nl/kaarten/wegbeheerders.aspx>
- Rijkswaterstaat. (2009). *Tracebesluit A4 Burgerveen-Leiden 2009, gedeelte Leiderdorp-Leiden*. Ministerie van Verkeer en Waterstaat, Rotterdam.
- Rijkswaterstaat. (2016). *Historische filegegevens*. Ministerie van infrastructuur en milieu, Rijswijk.
- Rijkswaterstaat. (2017). *A4: optimalisatie Leiden*. Rijswijk. Retrieved September 18, 2017, from <https://www.rijkswaterstaat.nl/wegen/projectenoverzicht/a4-optimalisatie-leiden/index.aspx>
- Ros, B. G., Knoop, V. L., van Arem, B., & Hoogendoorn, S. P. (2012). *Car-following Behavior at Sags and its Impacts on Traffic Flow*. TU Delft, Delft.
- VID. (2006-2012). *File-top50*. VID Nederland BV, Amsterdam. Retrieved September 10, 2017, from <http://www.vid.nl/top50.2006.html>: <http://www.vid.nl/top50.2006>
- VID. (2011-2014). *File-top50*. VID Nederland BV, Amsterdam. Retrieved September 10, 2017, from <http://www.vid.nl/top50.2011.html>: <http://www.vid.nl/top50.2011.html>
- VID. (2016). *File-top50*. VID Nederland BV, Amsterdam. Retrieved September 10, 2017, from <http://www.vid.nl/top50.2016.html>
- VID. (2017). *Meer files door ongevallen*. VID Nederland BV, Amsterdam. Retrieved October 8, 2017, from VID: <https://www.vid.nl/Nieuws/article/VID.2017.097.01>

APPENDICES

USED DATA

A. CONGESTION DATA SAMPLE

This following table shows the form of the congestion data files. The shown data shows the first ten columns (35 in total) and the first thirteen rows (2184 in total in the direction of The Hague and 629 in total in the direction of Amsterdam) in the direction of The Hague. The form of the congestion file in the direction of Amsterdam is exactly the same. This data is delivered by Rijkswaterstaat (Rijkswaterstaat, 2016).

Datum	Jaar	Mnd	Dag	Richt	Hm	Begt	St Uur	St Min	Zwaarte
20160106	2016	1	6	1	36.1	837	8	37	14
20160106	2016	1	6	1	36.1	905	9	5	5.367
20160112	2016	1	12	1	36.1	756	7	56	11.467
20160112	2016	1	12	1	36.1	828	8	28	11.533
20160113	2016	1	13	1	36.1	1644	16	44	142.017
20160113	2016	1	13	1	36.1	1624	16	24	64.117
20160121	2016	1	21	1	36.1	948	9	48	135.667
20160121	2016	1	21	1	36.1	914	9	14	12.933
20160128	2016	1	28	1	36.1	1606	16	6	0.767
20160119	2016	1	19	1	29.3	1756	17	56	1829
20160119	2016	1	19	1	29.9	1657	16	57	1702

B. TRAFFIC INTENSITY DATA SAMPLE

This following table shows the form of the intensity data files. The shown data shows all the columns and the first thirteen rows (total rows is different per file depending on the amount of lanes) in the direction of The Hague for the demand analysis of bottleneck 33.5. The form of the intensity files used for all the demand and capacity analysis in all bottlenecks in both direction are all exactly the same. This data is delivered by the Nationale Databank Wegverkeersgegevens (NDW, 2016)

Rijbaan	Dag	Maand	Jaar	Uur	Minuut	Intensiteit
1A	1	1	2016	0	0	0.00
3A	1	1	2016	0	0	60.00
5A	1	1	2016	0	0	420.00
1A	1	1	2016	0	1	0.00
3A	1	1	2016	0	1	180.00
5A	1	1	2016	0	1	120.00
1A	1	1	2016	0	2	0.00
3A	1	1	2016	0	2	0.00
5A	1	1	2016	0	2	60.00
1A	1	1	2016	0	3	0.00
3A	1	1	2016	0	3	0.00

C. TRAFFIC ACCIDENT DATA SAMPLE

This following table shows the form of the traffic incident data files. The shown data shows columns nine until sixteen (47 columns in total) and the first thirteen rows (of the total of 162 rows in the direction of The Hague and by chance the same number of rows in the direction of Amsterdam) of the accident data in the direction of The Hague. The form of the traffic accident file in the direction of Amsterdam is exactly the same. This data is delivered by the Nationale Databank Wegverkeersgegevens (NDW, 2016).

Beschrijving	Day	Month	Year	Hour	Minute	Second
ongeval	28	1	2016	19	19	4
ongeval	30	1	2016	8	8	53
ongeval	2	2	2016	7	7	27
ongeval	2	2	2016	9	9	10
ongeval	4	2	2016	17	17	41
ongeval	4	2	2016	17	17	26
ongeval	5	2	2016	17	17	30
ongeval	9	2	2016	13	13	39
ongeval	10	2	2016	15	15	26
ongeval	10	2	2016	15	15	19
ongeval	10	2	2016	19	19	7

D. ROAD MAINTENANCE DATA SAMPLE

This following table shows the form of the road maintenance data files. The shown data shows columns twelve until 21 (47 columns in total) and the first twelve rows (of the total of 481 rows in the direction of The Hague and 375 rows in the direction of Amsterdam) of the road maintenance data in the direction of The Hague. The form of the road maintenance data file in the direction of Amsterdam is exactly the same. This data is delivered by the Nationale Databank Wegverkeersgegevens (NDW, 2016).

Day Start	Month Start	Year	Hour Start	Minute Start	Day End	Month End	Hour End	Minute End	expliciet afgesloten
13	1	2016	19	28	13	1	21	0	TRUE
13	1	2016	19	28	13	1	21	0	TRUE
28	1	2016	21	3	28	1	21	30	TRUE
28	1	2016	21	40	28	1	22	0	TRUE
28	1	2016	21	3	28	1	21	40	TRUE
28	1	2016	21	58	28	1	22	30	TRUE
28	1	2016	21	40	28	1	22	3	TRUE
28	1	2016	21	58	28	1	22	34	TRUE
28	1	2016	22	35	28	1	23	0	TRUE
28	1	2016	22	35	28	1	22	52	TRUE
4	2	2016	20	4	4	2	22	0	TRUE

E. ROAD BLOCKAGE DATA SAMPLE

This following table shows the form of the road blockage data files. It is in the exact same form as the road maintenance data. The shown data shows columns twelve until 21 (47 columns in total) and the first twelve rows (of the total of 420 rows in the direction of The Hague and 401 rows in the direction of Amsterdam) of the road blockage data in the direction of The Hague. The form of the road blockage data file in the direction of Amsterdam is exactly the same. This data is delivered by the Nationale Databank Wegverkeersgegevens (NDW, 2016)

Day Start	Month Start	Year	Hour Start	Minute Start	Day End	Month End	Hour End	Minute End	expliciet afgesloten
29	1	2016	8	2	29	1	8	30	TRUE
30	1	2016	11	1	30	1	11	8	TRUE
30	1	2016	17	33	30	1	18	34	TRUE
2	2	2016	22	57	2	2	23	56	TRUE
3	2	2016	0	42	3	2	1	41	TRUE
3	2	2016	13	14	3	2	14	14	TRUE
5	2	2016	15	22	5	2	16	22	TRUE
6	2	2016	22	25	6	2	23	24	TRUE
7	2	2016	18	13	7	2	19	13	TRUE
7	2	2016	20	14	7	2	21	13	TRUE
29	1	2016	8	2	29	1	8	30	TRUE

F. RESULTS DEMAND ANALYSIS PER BOTTLENECK PER MONTH

Month	Demand 30.7 (veh/h)	Demand 33.5 (veh/h)	Demand 35.1 (veh/h)	Capacity 36.1 (veh/h)
<i>January</i>	7806	6041	4812	6714
<i>February</i>	7913	6213	5220	6760
<i>March</i>	8340	5949	5386	7180
<i>April</i>	8186	5810	5013	8200
<i>May</i>	8085	5606	5035	6750
<i>June</i>	8980	5723	4984	7920
<i>July</i>	7915	5676	5187	7733
<i>August</i>	7623	5400	4606	7180
<i>September</i>	8329	5971	5232	7140
<i>October</i>	8236	5858	4906	7856
<i>November</i>	7559	5602	4979	7106
<i>December</i>	7821	5313	4914	7120
Average	8066	5763	5022	7305

Month	Demand 31.8 (veh/h)	Demand 33.3 (veh/h)	Demand 34.5 (veh/h)	Demand 35.1 (veh/h)
<i>January</i>	7469	3306	6442	4500
<i>February</i>	6717	4200	6849	N/A*
<i>March</i>	7000	3477	7420	5633
<i>April</i>	6878	4618	7409	5010
<i>May</i>	7257	3804	6562	4536
<i>June</i>	7426	3782	6925	5040
<i>July</i>	N/A*	4106	7280	5460
<i>August</i>	6060	3000	7176	5760
<i>September</i>	N/A*	4151	7196	5172
<i>October</i>	6972	3458	6310	4692
<i>November</i>	7028	4443	6962	5601
<i>December</i>	6880	3932	6069	5050
Average	6969	3856	6883	5132

* Data unavailable due to the absence of a congestion starting at this point in this month.

G. RESULTS CAPACITY ANALYSIS PER BOTTLENECK PER MONTH

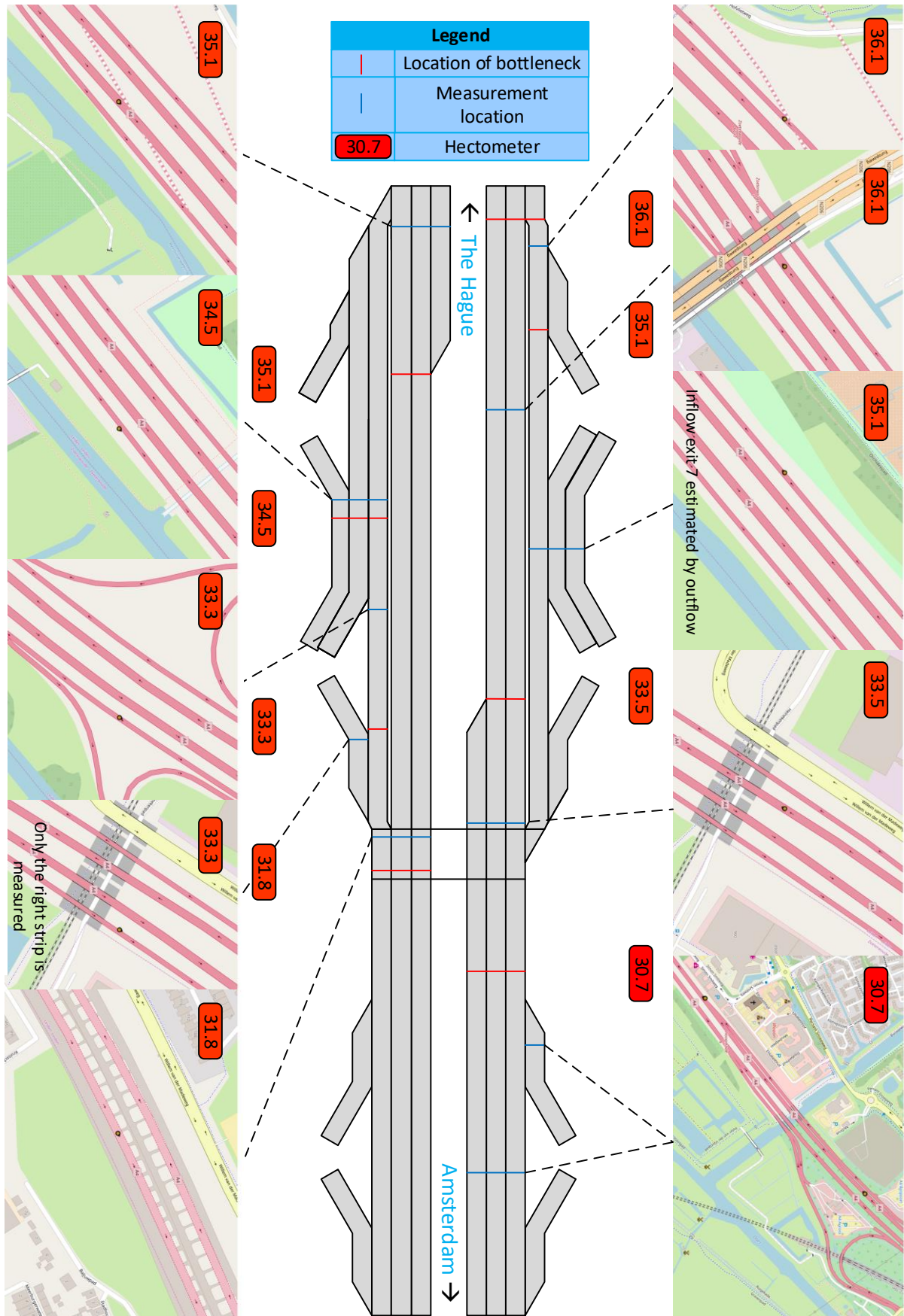
Month	Capacity 30.7 (veh/h)	Capacity 33.5 (veh/h)	Capacity 35.1 (veh/h)	Capacity 36.1 (veh/h)
<i>January</i>	5109	3556	1886	4783
<i>February</i>	4951	3545	1951	4825
<i>March</i>	5210	3861	1905	4830
<i>April</i>	5800	3549	1903	5345
<i>May</i>	4968	3848	1947	5120
<i>June</i>	5193	3713	1974	3969
<i>July</i>	4759	3869	1874	3549
<i>August</i>	4799	3371	1687	5218
<i>September</i>	4976	3889	1908	5215
<i>October</i>	4898	3751	2022	4241
<i>November</i>	4789	3472	1970	4892
<i>December</i>	4896	3613	2027	5369
Average	5029	3670	1921	4780

Month	Capacity 31.8 (veh/h)	Capacity 33.3 (veh/h)	Capacity 34.5 (veh/h)	Capacity 35.1 (veh/h)
<i>January</i>	4794	N/A**	1527	3673
<i>February</i>	4094	N/A**	1521	N/A*
<i>March</i>	5418	N/A**	1628	2555
<i>April</i>	4437	N/A**	1683	3806
<i>May</i>	5260	1544	1630	3467
<i>June</i>	4549	1523	1715	3462
<i>July</i>	N/A*	1503	1699	1770
<i>August</i>	5610	1297	1680	2916
<i>September</i>	N/A*	1625	1535	3558
<i>October</i>	4809	1786	1746	3060
<i>November</i>	4546	1660	1786	3139
<i>December</i>	4598	1640	1834	3561
Average	4812	1572	1665	3179

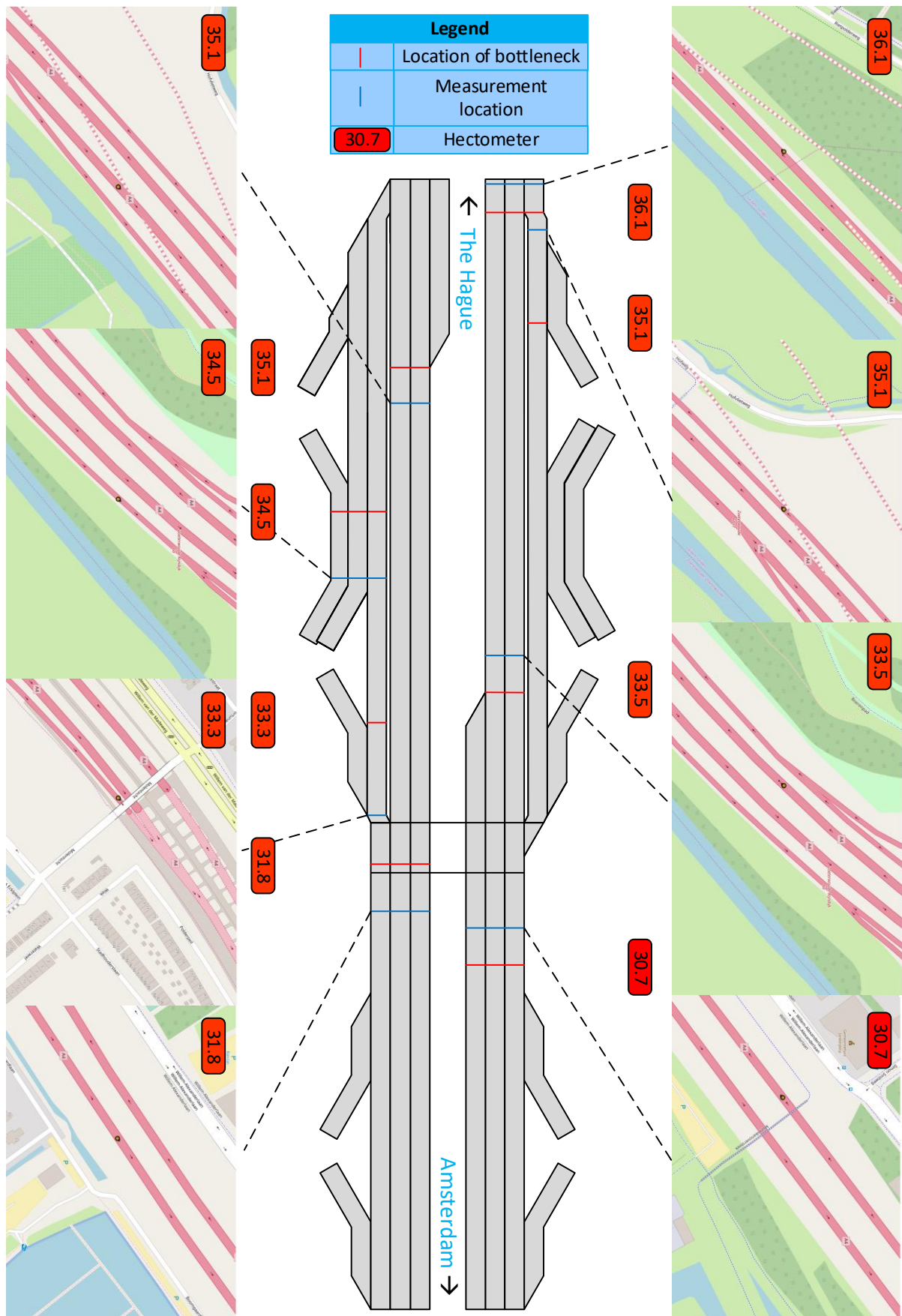
* Data unavailable due to the absence of a congestion starting at this point in this month.

** Data unavailable because intensity measurement did not start yet.

H. MEASURING POINTS DEMAND ANALYSIS



I. MEASURING POINTS CAPACITY ANALYSIS



I. PYTHON SCRIPT FOR DETERMINING TOTAL LEVEL OF CONGESTION IN THE DIRECTION OF THE HAGUE EXPLAINED

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
```

3. State a variable as zero before the start of a loop for the total level of congestion in the direction of The Hague.

```
FileTotalTH = 0
```

4. Start a loop in the length of the total congestion data file (2184 excel rules).

```
for i in range(2,2184):
```

5. In this loop add the level of congestion for every congestion to the sum of the level of congestion of all the congestions before. The level of congestion for every congestion is given in column 16 of the excel file.

```
FileTotalTH = FileTotalTH + A4FileH.cell(row=i, column=16).value
```

6. Print the outcome of the loop which gives the total level of congestion.

```
print(FileTotalTH, 'kmmin')
```

```
179578.59800000003 kmmin
```

II. PYTHON SCRIPT FOR DETERMINING TOTAL LEVEL OF CONGESTION IN THE DIRECTION OF AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAms = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAms.get_sheet_by_name('Sheet1')
```

```
FileTotalAms = 0
for i in range(2,629):
    FileTotalAms = FileTotalAms + A4FileA.cell(row=i, column=16).value
print(FileTotalAms, 'kmmin')
```

```
64697.287999999997 kmmin
```

III. PYTHON SCRIPT FOR DETERMINING THE LOCATIONS OF THE BOTTLENECKS IN THE DIRECTION OF THE HAGUE EXPLAINED

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
```

3. Define a variable which is equal to zero for every hectometer between 28.0 and 36.5.

```
Hm = np.linspace(28.0,36.5,86)
print (Hm)
a = np.zeros(len(Hm))
```

4. Start a loop in the length of the total congestion data file (2184 excel rules).

```
for i in range(1,2184):
```

5. Load the starting point of every congestion in the loop. The starting point is given as a hectometer in column 30 of the congestion data file.

```
l = A4FileH.cell(row=i, column=30).value
```

6. Create a new loop in the length of the number of hectometers.

```
for j in range(len(Hm)):
```

7. In this loop add +1 to a certain variable given in point 3 (a) when a congestion, loaded by the first loop, starts at this certain variable.

```
if Hm[j] == l:
    a[j] += 1
```

8. This leads to an array with the length of the number of hectometers (a) which states the number of congestions which started at a certain hectometer. This can be plotted according to the following statements. This plot in be seen in figure 6 in the report.

```
plt.figure(figsize=(20,5))
print (a)
plt.rc('xtick', labels=30)
plt.rc('ytick', labels=30)
plot = np.linspace(0,86,86)
plt.bar(plot,a, width=1)
plt.xlim(0, 86)
plt.ylim(0, 500)
plt.xlabel('Hectometer (km)', size = 30)
plt.ylabel('Traffic jams', size = 30)
plt.xticks([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85], ['28', '28.5', '29', '29.5', '30', '30.5', '31', '31.5', '32', '32.5', '33', '33.5', '34', '34.5', '35', '35.5', '36', '36.5'])
```

IV. PYTHON SCRIPT FOR DETERMINING THE LOCATIONS OF THE BOTTLENECKS IN THE DIRECTION OF AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAMS = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAMS.get_sheet_by_name('Sheet1')
Hm = np.linspace(28.0,36.5,86)
print (Hm)
a = np.zeros(len(Hm))
for i in range(1,629):
    l = A4FileA.cell(row=i, column=30).value
    for j in range(len(Hm)):
        if Hm[j] == l:
            a[j] += 1

plt.figure(figsize=(20,5))
print (a)
plt.rc('xtick', labels=25)
plt.rc('ytick', labels=25)
plot = np.linspace(0,86,86)
plt.bar(plot,a, width=1)
plt.xlim(0, 86)
plt.ylim(0, 500)
plt.xlabel('Hectometer', size = 30)
plt.ylabel('Amount of traffic jams', size = 30)
plt.xticks([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85], ['28', '28.5', '29', '29.5', '30', '30.5', '31', '31.5', '32', '32.5', '33', '33.5', '34', '34.5', '35', '35.5', '36', '36.5'])
```

V. PYTHON SCRIPT FOR DETERMINING THE TOTAL LEVEL OF CONGESTION STARTING PER HECTOMETER IN THE DIRECTION OF THE HAGUE EXPLAINED

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
```

3. Define a variable which is equal to zero for every hectometer between 28.0 and 36.5.

```
Hm = np.linspace(28.0,36.5,86)
print (Hm)
a = np.zeros(len(Hm))
```

4. Start a loop in the length of the total congestion data file (2184 excel rules).

```
for i in range(1,2184):
```

5. Load the starting point of every congestion in the loop. The starting point is given as a hectometer in column 30 of the congestion data file.

```
l = A4FileH.cell(row=i, column=30).value
```

6. Create a new loop in length of the number of hectometers.

```
for j in range(len(Hm)):
```

7. In this loop add the level of congestion of a congestion to a certain variable given in point 3 (a) when a congestion, loaded by the first loop, starts at this certain variable. A4FileH.cell(row=i, column=16).value loads the level of congestion from a congestion in row i from in step 2 loaded congestion data.

```
if Hm[j] == l:
    a[j] += a[j+1] + A4FileH.cell(row=i, column=16).value
```

8. This leads to an array with the length of the number of hectometers (a) which states the total sum of the level of congestion of all the congestions which started at a certain hectometer. This can be plotted according to the following statements. This plot in be seen in figure 8 in the report.

```
plt.figure(figsize=(20,5))
plt.rc('xtick', labels=30)
plt.rc('ytick', labels=30)
plot = np.linspace(0,86,86)
plt.bar(plot,a, width=1)
plt.xlim(0, 86)
plt.ylim(0, 35000)
plt.xlabel('Hectometer', size = 25)
plt.ylabel('Level of congestion (kmin)', size = 25)
plt.xticks([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85], ['28', '28.5', '29', '29.5', '30', '30.5', '31', '31.5', '32', '32.5', '33', '33.5', '34', '34.5', '35', '35.5', '36', '36.5'])
```

9. To control if the outcome of the loop is correct the sum of the array is printed as well as the earlier determined total level of congestion in the direction of The Hague (appendix I). These two have to be the same.

```
print(sum(a))
print(FileTotalTH)
```

```
179476.815
179476.81499999974
```

VI. PYTHON SCRIPT FOR DETERMINING THE TOTAL LEVEL OF CONGESTION STARTING PER HECTOMETER IN THE DIRECTION OF AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAms = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAms.get_sheet_by_name('Sheet1')
Hm = np.linspace(28.0,36.5,86)
a = np.zeros(len(Hm))
for i in range(2,629):
    l = round(A4FileA.cell(row=i, column=30).value,1)
    for j in range(len(Hm)):
        if Hm[j] == l:
            a[j] += a[j+1] + A4FileA.cell(row=i, column=16).value

print(sum(a))
print(FileTotalAms)
plt.figure(figsize=(20,5))
plt.rc('xtick', labels=30)
plt.rc('ytick', labels=30)
plot = np.linspace(0,86,86)
plt.bar(plot,a, width=1)
plt.xlim(0, 86)
plt.ylim(0, 35000)
plt.xlabel('Hectometer', size = 25)
plt.ylabel('Level of congestion(kmmin)', size = 25)
plt.xticks([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85], ['28', '28.5', '29', '29.5', '30', '30.5', '31', '31.5', '32', '32.5', '33', '33.5', '34', '34.5', '35', '35.5', '36', '36.5'])
64475.321
64475.32099999992
```

VII. PYTHON SCRIPT PEAK HOUR ANALYSIS IN THE DIRECTION OF THE HAGUE EXPLAINED

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
```

3. Define a variable which is equal to zero for every hour in a day.

```
H = np.linspace(0,23,24)
print (H)
a = np.zeros(len(H))
```

4. Start a loop in the length of the total congestion data file (2184 excel rules).

```
for i in range(1,2184):
```

5. Load the starting hour of every congestion in the loop.

```
h = A4FileH.cell(row=i, column=11).value
```

6. Create a new loop in length of the amount of hectometers.

```
for j in range(len(H)):
```

7. In this loop add +1 to a certain variable given in point 3 (a) when a congestion loaded by the first loop starts at this certain variable.

```
if H[j] == h:
    a[j] += 1
```

8. This leads to an array with the length of the number of hours in a day (a) which states the number of congestions which started at a certain hour. This can be plotted according to the following statements. This plot in be seen in figure 10 in the report.

```
plt.figure(figsize=(20,5))
print (a)
plt.rc('xtick', labels=30)
plt.rc('ytick', labels=30)
plot = np.linspace(0,24,24)
plt.bar(plot,a, width=1)
plt.xlim(0, 24)
plt.xlabel('Time (hours)', size = 30)
plt.ylabel('Number of congestions', size = 30)
```

VIII. PYTHON SCRIPT PEAK HOUR ANALYSIS IN THE DIRECTION OF AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAMS = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAMS.get_sheet_by_name('Sheet1')
H = np.linspace(0,23,24)
print (H)
a = np.zeros(len(H))
for i in range(1,629):
    h = A4FileA.cell(row=i, column=11).value
    for j in range(len(H)):
        if H[j] == h:
            a[j] += 1

plt.figure(figsize=(20,5))
print (a)
plt.rc('xtick', labels=30)
plt.rc('ytick', labels=30)
plot = np.linspace(0,24,24)
plt.bar(plot,a, width=1)
plt.xlim(0, 24)
plt.xlabel('Time (hours)', size = 30)
plt.ylabel('Number of congestions', size = 30)
```

IX. PYTHON SCRIPT FOR DETERMINING THE DEMAND OF BOTTLENECK 33.5 IN THE DIRECTION OF THE HAGUE EXPLAINED

This script is used to determine the demand of the highway at hectometer 33.5 in the direction of The Hague in January 2016. This script is adjusted to the layout of the road for all other bottlenecks. To determine the demand in other months the mentioned arrays are adjusted.

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague. And the excel file with the intensities for every minute and every strip in 2016 at the most upstream measuring point of the road where the layout is the same as just upstream from the bottleneck.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
Int = openpyxl.load_workbook('Demand intensiteit 335 jan-aug.xlsx')
I = Int.get_sheet_by_name('Sheet1')
```

3. Define an array for every single strip. This array will later on be filled with all intensities during one congestion on one strip in the loop. The strips are named A1, A3 and A5.

```
DemandA1j = []
DemandA3j = []
DemandA5j = []
```

4. Define another array for every single strip. This array will later on be filled with all the maximum intensities of all congestion on one strip in the loop.

```
DemandA1jmax = []
DemandA3jmax = []
DemandA5jmax = []
```

5. Start a loop in the length of the amount of congestions in the direction to The Hague in January 2016. This array is changed for every month. This is changed manually to reduce the needed computing power of the computer when running the script.

```
for i in range(2,177):
```

6. Define variables to represent the date and time of the congestion and starting location of a certain congestion. A4FileH.cell(row=i, column=4).value loads the day of a congestion in row i from the in step two loaded congestion data, A4FileH.cell(row=i, column=3).value loads the month, A4FileH.cell(row=i, column=10).value loads the starting time, A4FileH.cell(row=i, column=13).value loads the ending time and A4FileH.cell(row=i, column=30).value loads the starting hectometer of the starting location.

```
D = A4FileH.cell(row=i, column=4).value
M = A4FileH.cell(row=i, column=3).value
Ts = A4FileH.cell(row=i, column=10).value
Te = A4FileH.cell(row=i, column=13).value
Hm = A4FileH.cell(row=i, column=30).value
```

7. Start a new loop in the length of the measured intensities during one month. Simply calculated by multiplying the amount of strips by the amount of days in a month by the amount of hours in a day by the amount minutes in an hour. Thus in this case: $3 \times 31 \times 24 \times 60 + 1 = 133921$. This array is changed for every month.

```
for j in range(2,133921):
```

The explanation of the sheet continues on the next page.

8. Load the data and time of the measured intensity. `I.cell(row=i, column=2).value` loads the day of the measured in row `i` from in step two loaded intensity data, `I.cell(row=i, column=3).value` loads the month, `I.cell(row=i, column=5).value` loads hour and `I.cell(row=i, column=6).value` loads the minute. In last line the time is converted to the same form as the starting and ending time in the previous loop.

```
d = I.cell(row=j, column=2).value
m = I.cell(row=j, column=3).value
h = I.cell(row=j, column=5).value
mi = I.cell(row=j, column=6).value
t = h*100+mi
```

9. State an if statement which determines if the data from a certain measured intensity is useful. This is the case when the day and month of the first loop are equal to the day and month of the second loop and when the time of the second loop is higher than the starting time and lower than the ending time of the first loop. `I.cell(row=j, column=1).value` loads the strip from which the intensity is measured. At last it is stated that the starting hectometer of the congestion loaded by loop one must be equal to the hectometer of the investigated point.

```
if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='1A' and Hm==33.5:
```

10. When the previous statement is true load the intensity value loaded by `I.cell(row=j, column=7).value` into the area defined by in step in step 3. The `np.hstack` function puts all intensity values on one strip during one congestion in one array.

```
DemandA1j.append(I.cell(row=j, column=7).value)
np.hstack(DemandA1j)
```

11. Do exactly the same for the two other strips.

```
if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='3A' and Hm==33.5:
    DemandA3j.append(I.cell(row=j, column=7).value)
    np.hstack(DemandA3j)
if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='5A' and Hm==33.5:
    DemandA5j.append(I.cell(row=j, column=7).value)
    np.hstack(DemandA5j)
```

12. Delete the empty arrays. This is done as an error occurs when the maximum is taken of an empty error.

```
if DemandA1j != []:
```

13. Take the maximum of each array coming from step 10 and put these maximums in a new array defined in step four.

```
DemandA1jmax.append(max(DemandA1j))
np.hstack(DemandA1jmax)
```

14. Do exactly the same for the two other strips.

```
if DemandA3j != []:
    DemandA3jmax.append(max(DemandA3j))
    np.hstack(DemandA3jmax)
if DemandA5j != []:
    DemandA5jmax.append(max(DemandA5j))
    np.hstack(DemandA5jmax)
```

15. Calculate the average of all the maximum demand of all congestions for one strip. This done by dividing the sum of the array determined in step thirteen by the length of this same array.

```
AvgA1 = sum(DemandA1jmax)/len(DemandA1jmax)
```

16. Do exactly the same for the other two strips.

```
AvgA3 = sum(DemandA3jmax)/len(DemandA3jmax)
AvgA5 = sum(DemandA5jmax)/len(DemandA5jmax)
```

The explanation of the sheet continues on the next page.

17. Sum the average of the three arrays and print this as the demand.

```
Demand = AvgA1 + AvgA3 + AvgA5  
print(Demand)
```

6040.714285714286

X. PYTHON SCRIPT FOR DETERMINING THE DEMAND OF BOTTLENECK 34.5 IN THE DIRECTION AMSTERDAM OVERVIEW

```
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline  
import openpyxl  
A4FileAms = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')  
A4FileA = A4FileAms.get_sheet_by_name('Sheet1')  
Int = openpyxl.load_workbook('Demand intensiteit 345 jan-aug.xlsx')  
I = Int.get_sheet_by_name('Sheet1')  
  
#33.5 January  
DemandA1j = []  
DemandA3j = []  
DemandA5j = []  
DemandA1jmax = []  
DemandA3jmax = []  
DemandA5jmax = []  
for i in range(2,47):  
    D = A4FileA.cell(row=i, column=4).value  
    M = A4FileA.cell(row=i, column=3).value  
    Ts = A4FileA.cell(row=i, column=10).value  
    Te = A4FileA.cell(row=i, column=13).value  
    Hm = A4FileA.cell(row=i, column=30).value  
    L = A4FileA.cell(row=i, column=18).value  
    for j in range(2,133921):  
        d = I.cell(row=j, column=2).value  
        m = I.cell(row=j, column=3).value  
        h = I.cell(row=j, column=5).value  
        mi = I.cell(row=j, column=6).value  
        t = h*100+mi  
        if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='1A' and Hm==34.5:  
            DemandA1j.append(I.cell(row=j, column=7).value)  
            np.hstack(DemandA1j)  
        if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='3A' and Hm==34.5:  
            DemandA3j.append(I.cell(row=j, column=7).value)  
            np.hstack(DemandA3j)  
        if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='5A' and Hm==34.5:  
            DemandA5j.append(I.cell(row=j, column=7).value)  
            np.hstack(DemandA5j)  
    if DemandA1j != []:  
        DemandA1jmax.append(max(DemandA1j))  
        np.hstack(DemandA1jmax)  
    if DemandA3j != []:  
        DemandA3jmax.append(max(DemandA3j))  
        np.hstack(DemandA3jmax)  
    if DemandA5j != []:  
        DemandA5jmax.append(max(DemandA5j))  
        np.hstack(DemandA5jmax)  
AvgA1 = sum(DemandA1jmax)/len(DemandA1jmax)  
AvgA3 = sum(DemandA3jmax)/len(DemandA3jmax)  
AvgA5 = sum(DemandA5jmax)/len(DemandA5jmax)  
Demand = AvgA1 + AvgA3 + AvgA5  
print(Demand)
```

6441.818181818182

XI. PYTHON SCRIPT FOR DETERMINING THE CAPACITY OF BOTTLENECK 33.5 IN THE DIRECTION OF THE HAGUE EXPLAINED

This script is used to determine the capacity of the highway at hectometer 33.5 in the direction of The Hague in January 2016. This script is adjusted to the layout of the road for all other bottlenecks. To determine the capacity in other months the mentioned arrays are adjusted. The script is very similar to the script which is used to determine the demand. The mayor difference is that to determine the average demand the maximum demand per congestion is determined. This is not necessary for determining the capacity as the capacity of a road stays more or less the same during a congestion as explained in chapter 3.3. Therefore all intensities during all congestion are put into one array per lane. The capacity is determined by taking making a sum of the average of these arrays. The step which differ from the previous explanation are 3, 6, 10 and 11. All the other steps are also added to keep the overview.

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague. And the excel file with the intensities for every minute and every strip in 2016 at the most downstream measuring point of the road where the layout is the same as just downstream of the bottleneck.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
Int = openpyxl.load_workbook('Demand intensiteit 335 jan-aug.xlsx')
I = Int.get_sheet_by_name('Sheet1')
```

3. Define an array for every single strip. This array will later on be filled with all intensities during all congestions on one strip in the loop. The strips are named A1 and A3.

```
CapacityA1j = []
CapacityA3j = []
```

4. Start a loop in the length of the amount of congestions in the direction to The Hague in January 2016. This array is changed for every month. This is changed manually to reduce the needed computing power of the computer when running the script.

```
for i in range(2,177):
```

5. Define variables to represent the date and time of the congestion and starting location of a certain congestion. A4FileH.cell(row=i, column=4).value loads the day of a congestion in row i from the in step two loaded congestion data, A4FileH.cell(row=i, column=3).value loads the month, A4FileH.cell(row=i, column=10).value loads the starting time, A4FileH.cell(row=i, column=13).value loads the ending time and A4FileH.cell(row=i, column=30).value loads the starting hectometer of the starting location.

```
D = A4FileH.cell(row=i, column=4).value
M = A4FileH.cell(row=i, column=3).value
Ts = A4FileH.cell(row=i, column=10).value
Te = A4FileH.cell(row=i, column=13).value
Hm = A4FileH.cell(row=i, column=30).value
```

6. Start a new loop in the length of the measured intensities during one month. Simply calculated by multiplying the amount of strips by the amount of days in a month by the amount of hours in a day by the amount minutes in an hour. Thus in this case: $2 \cdot 31 \cdot 24 \cdot 60 + 1 = 89281$. This array is changed for every month.

```
for j in range(2,89281):
```

The explanation of the sheet continues on the next page.

7. Load the data and time of the measured intensity. `I.cell(row=i, column=2).value` loads the day of the measured in row `i` from in step two loaded intensity data, `I.cell(row=i, column=3).value` loads the month, `I.cell(row=i, column=5).value` loads hour and `I.cell(row=i, column=6).value` loads the minute. In last line the time is converted to the same form as the starting and ending time in the previous loop.

```
d = I.cell(row=j, column=2).value
m = I.cell(row=j, column=3).value
h = I.cell(row=j, column=5).value
mi = I.cell(row=j, column=6).value
t = h*100+mi
```

8. State an if statement which determines if the data from a certain measured intensity is useful. This is the case when the day and month of the first loop are equal to the day and month of the second loop and when the time of the second loop is higher than the starting time and lower than the ending time of the first loop. `I.cell(row=j, column=1).value` loads the strip from which the intensity is measured. At last it is stated that the starting hectometer of the congestion loaded by loop one must be equal to the hectometer of the investigated point.

```
if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='1A' and Hm==33.5:
```

9. When the previous statement is true load the intensity value loaded by `I.cell(row=j, column=7).value` into the area defined by in step in step 3. The `np.hstack` function puts all intensity values on one strip during one congestion in one array.

```
DemandA1j.append(I.cell(row=j, column=7).value)
np.hstack(DemandA1j)
```

10. Do exactly the same for the other strip.

```
if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='3A' and Hm==33.5:
    CapacityA3j.append(I.cell(row=j, column=7).value)
    np.hstack(CapacityA3j)
```

11. Make a sum of the average of the two strips and print this as the capacity.

```
Capacity=sum(CapacityA1j)/len(CapacityA1j)+sum(CapacityA3j)/len(CapacityA3j)
print(Capacity)
```

3556.258064516129

XII. PYTHON SCRIPT FOR DETERMINING THE CAPACITY OF BOTTLENECK 34.5 IN THE DIRECTION AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAms = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAms.get_sheet_by_name('Sheet1')
Int = openpyxl.load_workbook('Capaciteit intensiteit 345 jan-aug.xlsx')
I = Int.get_sheet_by_name('Sheet1')
CapacityA1j = []
CapacityA3j = []
CapacityA5j = []
for i in range(2,47):
    D = A4FileA.cell(row=i, column=4).value
    M = A4FileA.cell(row=i, column=3).value
    H = A4FileA.cell(row=i, column=11).value
    Mi = A4FileA.cell(row=i, column=12).value
    Hm = A4FileA.cell(row=i, column=30).value
    for j in range(2,133921):
        d = I.cell(row=j, column=2).value
        m = I.cell(row=j, column=3).value
        h = I.cell(row=j, column=5).value
        mi = I.cell(row=j, column=6).value
        t = h*100+mi
        if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='1A' and Hm==34.5:
            CapacityA1j.append(I.cell(row=j, column=7).value)
            np.hstack(CapacityA1j)
        if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='3A' and Hm==34.5:
            CapacityA3j.append(I.cell(row=j, column=7).value)
            np.hstack(CapacityA3j)
        if d==D and m==M and t>=Ts and t<=Te and I.cell(row=j, column=1).value=='5A' and Hm==34.5:
            CapacityA5j.append(I.cell(row=j, column=7).value)
            np.hstack(CapacityA5j)
Capacity=sum(CapacityA1j)/len(CapacityA1j)+sum(CapacityA3j)/len(CapacityA3j)+sum(CapacityA5j)/len(CapacityA5j)
print(Capacity)
```

1526.9182389937105

XIII. PYTHON SCRIPT FOR DETERMINING THE PERCENTAGE OF THE TOTAL LEVEL OF CONGESTION CAUSED BY ACCIDENTS IN THE DIRECTION OF THE HAGUE EXPLAINED

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague and the excel file with the data of all accidents in the direction of The Hague.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
AccTH = openpyxl.load_workbook('Ongevallen richting The Hague_situatieberichten_00001.xlsx')
AH = AccTH.get_sheet_by_name('Situatieberichten')
```

3. Introduce a variable for the total level of congestion caused by accidents in the direction of The Hague and set this variable to zero at the start of the loop.

```
FileTotalATH = 0
```

4. Start a loop with the length of the amount of accidents in the direction of The Hague.

```
for i in range(2,162):
```

5. Define variables to represent the date and time of a certain accident. AH.cell(row=i, column=13).value loads the day of an accident in row i from the in step two loaded accident data, AH.cell(row=i, column=14).value loads the month, AH.cell(row=i, column=17).value loads the hour and AH.cell(row=i, column=18).value loads the minute.

```
d = AH.cell(row=i, column=13).value
m = AH.cell(row=i, column=14).value
h = AH.cell(row=i, column=17).value
mi = AH.cell(row=i, column=18).value
```

6. Start a new loop in the length of the amount of congestions in the direction of The Hague.

```
for j in range(2,2184):
```

7. Define variables to represent the date and time of a certain congestion. A4FileH.cell(row=i, column=3).value loads the day of a congestion in row i from the in step two loaded congestion data, A4FileH.cell(row=i, column=4).value loads the month, A4FileH.cell(row=i, column=11).value loads the hour and A4FileH.cell(row=i, column=12).value loads the minute.

```
D = A4FileH.cell(row=j, column=4).value
M = A4FileH.cell(row=j, column=3).value
H = A4FileH.cell(row=j, column=11).value
Mi = A4FileH.cell(row=j, column=12).value
```

8. State an if statement which determines if a congestion developed just after the time that an accident happened. This is the case when the month, day and hour are the same in both loops and when the starting minute of the congestion is equal to or higher than the starting minute of the accident.

```
if d==D and m==M and h==H and A4FileH.cell(row=j, column=16).value and mi<=Mi:
```

9. If this is the case add the level of congestion of the congestion to the variable defined in step three.

```
FileTotalATH = FileTotalATH + A4FileH.cell(row=j, column=16).value
```

The explanation of this sheet continues on the next page.

10. Determine the percentage of the total congestion which is caused by accidents by dividing the outcome from step nine by the earlier determined.

```
PTH=FileTotalATH/FileTotalTH*100
```

11. Print the level of congestion caused by accidents and the percentage of the total congestion which is caused by accidents.

```
print(FileTotalATH, 'kmmin')
print(PTH, '%')
```

20633.644999999986 kmmin

11.478347152989658 %

XIV. PYTHON SCRIPT FOR THE DETERMINING PERCENTAGE OF THE TOTAL LEVEL OF CONGESTION CAUSED BY ACCIDENTS IN THE DIRECTION OF AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAms = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAms.get_sheet_by_name('Sheet1')
AccAms = openpyxl.load_workbook('Ongevallen richting Amsterdam_situatieberichten_00001.xlsx')
AAms = AccAms.get_sheet_by_name('Situatieberichten')

FileTotalAms = 0
for i in range(2,162):
    d = AAms.cell(row=i, column=12).value
    m = AAms.cell(row=i, column=13).value
    h = AAms.cell(row=i, column=15).value
    mi = AAms.cell(row=i, column=16).value
    for j in range(2,8000):
        D = A4FileA.cell(row=j, column=4).value
        M = A4FileA.cell(row=j, column=3).value
        H = A4FileA.cell(row=j, column=11).value
        Mi = A4FileA.cell(row=j, column=12).value
        if d==D and m==M and h==H and A4FileA.cell(row=j, column=16).value and mi<=Mi:
            FileTotalAms = FileTotalAms + A4FileA.cell(row=j, column=16).value
PAms=FileTotalAms/FileTotalAms*100
print(FileTotalAms, 'kmmin')
print(PAms, '%')
```

14507.627000000001 kmmin

22.50105431813209 %

XV. PYTHON SCRIPT FOR DETERMINING THE PERCENTAGE OF THE TOTAL LEVEL OF CONGESTION CAUSED BY ROAD MAINTENANCE IN THE DIRECTION OF THE HAGUE EXPLAINED

This sheet is very similar to the sheet to determine the percentage of the total congestion which is caused by accidents. The main difference is that road maintenance works most of the time do have an ending time. This makes it possible to define a time interval in which the starting time of a congestion must be found. This exact sheet is also used to determine the congestion caused by road blockage.

1. Import some basic Python tools with numpy and the excel reader tool with openpyxl.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
```

2. Open the excel file with congestion data in the direction of The Hague and the excel file with the data of all road maintenance works in the direction of The Hague.

```
A4FileTH = openpyxl.load_workbook('File gegevens A4 richting The Hague.xlsx')
A4FileH = A4FileTH.get_sheet_by_name('Sheet1')
WorTH = openpyxl.load_workbook('Werkzaamheden richting The Hague.xlsx')
WTH = WorTH.get_sheet_by_name('Situatieberichten')
```

3. Introduce a variable for the total level of congestion caused by road maintenance works in the direction of The Hague and set this variable to zero at the start of the loop.

```
FileTotalWTH = 0
```

4. Start a loop in the length of the amount of congestions in the direction of The Hague.

```
for j in range(2,2184):
```

5. Define variables to represent the date and time of a certain congestion. A4FileH.cell(row=i, column=4).value loads the day of a congestion in row i from the in step two loaded congestion data, A4FileH.cell(row=i, column=3).value loads the month, A4FileH.cell(row=i, column=11).value loads the hour and A4FileH.cell(row=i, column=12).value loads the minute. The last statement convert the time to a form in which it's comparable to the time of the road maintenance work.

```
D = A4FileH.cell(row=j, column=4).value
M = A4FileH.cell(row=j, column=3).value
H = A4FileH.cell(row=j, column=11).value
Mi = A4FileH.cell(row=j, column=12).value
t = M*1000000+D*10000+H*100+Mi
```

6. Start a loop in the length of the amount of road maintenance works.

```
for i in range(2,481):
```

7. State an if statement which states if the road maintenance work has an ending time.

```
if WTH.cell(row=i, column=21).value=='TRUE':
```

The explanation of this sheet continues on the next page.

8. If it has an ending time, define variables to represent the starting and ending date and time of each road maintenance work. WTH.cell(row=i, column=13).value loads the month in which a road maintenance work started in row i from the in step two loaded road maintenance work, WTHFileH.cell(row=i, column=12).value loads the starting day, WTHFileH.cell(row=i, column=15).value loads the starting hour, WTHFileH.cell(row=i, column=16).value loads the starting minute, WTHFileH.cell(row=i, column=18).value loads the ending month, WTHFileH.cell(row=i, column=19).value loads the ending day, WTHFileH.cell(row=i, column=18).value loads the ending hour and WTHFileH.cell(row=i, column=18).value loads the ending minute. The last two statements convert the time to a form in which it's comparable to the time of the congestions.

```
Ms = WTH.cell(row=i, column=13).value
Ds = WTH.cell(row=i, column=12).value
Hs = WTH.cell(row=i, column=15).value
Mis = WTH.cell(row=i, column=16).value
Me = WTH.cell(row=i, column=18).value
De = WTH.cell(row=i, column=17).value
He = WTH.cell(row=i, column=19).value
Mie = WTH.cell(row=i, column=20).value
ts = Ms*1000000+Ds*10000+Hs*100+Mis
te = Me*1000000+De*10000+He*100+Mie
```

9. State an if statement which states if the congestion starts during the road maintenance work.

```
if t>=ts and t<=te:
```

10. If this is the case add the level of congestion of the congestion to the variable in defined in step three.

```
FileTotalWTH = FileTotalWTH + A4FileH.cell(row=j, column=16).value
```

11. If the answer to the statement of step seven is not the true, the road maintenance work doesn't have a defined ending. Which means that only a starting can be defined. This can be stated by an else statement.

```
else:
```

12. Define variables to represent the starting and date and time of each road maintenance work. WTH.cell(row=i, column=13).value loads the month in which a road maintenance work started in row i from the in step two loaded road maintenance work, WTHFileH.cell(row=i, column=12).value loads the starting day, WTHFileH.cell(row=i, column=15).value loads the starting hour and WTHFileH.cell(row=i, column=15).value loads the starting minute.

```
Ms = WTH.cell(row=i, column=13).value
Ds = WTH.cell(row=i, column=12).value
Hs = WTH.cell(row=i, column=15).value
Mis = WTH.cell(row=i, column=16).value
ts = Ms*1000000+Ds*10000+Hs*100+Mis
```

13. State an if statement which states if the congestion starts between the start of the road maintenance work and one hour after this. This one hour is an scattered duration of the work.

```
if t>=ts and t<=ts+100:
```

14. If this is the case add the level of congestion of the congestion to the variable in defined in step three.

```
FileTotalWTH = FileTotalWTH + A4FileH.cell(row=j, column=16).value
```

The explanation of this sheet continues on the next page.

15. Print the total level of congestion caused by road maintenance work and determine and print the percentage of the total level of congestion caused by road maintenance.

```
print(FileTotalWTH, 'kmmin')
PTH=FileTotalWTH/FileTotalTH*100
print(PTH, '%')
```

```
179578.59800000003 kmmin
1693.35 kmmin
0.9429575789426754 %
```

XVI. PYTHON SCRIPT FOR DETERMINING THE PERCENTAGE OF THE TOTAL LEVEL OF CONGESTION CAUSED BY ROAD MAINTENANCE IN THE DIRECTION OF AMSTERDAM OVERVIEW

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import openpyxl
A4FileAms = openpyxl.load_workbook('File gegevens A4 richting Amsterdam.xlsx')
A4FileA = A4FileAms.get_sheet_by_name('Sheet1')
WorAms = openpyxl.load_workbook('Werkzaamheden richting Amsterdam.xlsx')
WAms = WorAms.get_sheet_by_name('Situatieberichten')
FileTotalWAms = 0
for j in range(2,629):
    D = A4FileA.cell(row=j, column=4).value
    M = A4FileA.cell(row=j, column=3).value
    H = A4FileA.cell(row=j, column=11).value
    Mi = A4FileA.cell(row=j, column=12).value
    t = M*1000000+D*10000+H*100+Mi
    #print(d,m,h)
    for i in range(2,375):
        if WAms.cell(row=i, column=21).value=='TRUE':
            Ms = WAms.cell(row=i, column=13).value
            Ds = WAms.cell(row=i, column=12).value
            Hs = WAms.cell(row=i, column=15).value
            Mis = WAms.cell(row=i, column=16).value
            Me = WAms.cell(row=i, column=18).value
            De = WAms.cell(row=i, column=17).value
            He = WAms.cell(row=i, column=19).value
            Mie = WAms.cell(row=i, column=20).value
            ts = Ms*1000000+Ds*10000+Hs*100+Mis
            te = Me*1000000+De*10000+He*100+Mie
            if t>=ts and t<=te:
                FileTotalWAms = FileTotalWAms + A4FileA.cell(row=j, column=16).value
        else:
            Ms = WAms.cell(row=i, column=13).value
            Ds = WAms.cell(row=i, column=12).value
            Hs = WAms.cell(row=i, column=15).value
            Mis = WAms.cell(row=i, column=16).value
            ts = Ms*1000000+Ds*10000+Hs*100+Mis
            if t>=ts and t<=ts+100:
                FileTotalWAms = FileTotalWAms + A4FileA.cell(row=j, column=16).value
print(FileTotalWAms, 'kmmin')
PAms=FileTotalWAms/FileTotalAms*100
print(PAms, '%')
```

```
64697.28799999997 kmmin
778.5400000000001 kmmin
1.203358014017528 %
```