Read me first!

- Please read the whole assignment first, because some questions you have while reading, might be answered later.
- Even more questions you might have will not be answered at all! The model in this assignment is *not* explained in the greatest level of detail. This leaves quite a bit of freedom to make your own modeling decisions. Feel free to make any modeling assumptions required to implement your solution, but stick to the (few) requirements given in this document.
- There are four questions, three of which require programming a discrete-event simulation. **You have to use SimPy for Question 2 and/or 3**. If you choose not to use SimPy for one of those questions, you have to use discrete-event scheduling (with a future event set) as discussed in the lecture notes Chapter 8. **Question 4 should be programmed *without* the use of SimPy.**
- In principle, you should not have to ask any questions to the problem owner (Marko Boon). All information that we want to give you, is provided in this document. Everything that is unclear to you after reading this document, is probably intended that way. You should decide for yourself how to solve these issues. If you have any *urgent* questions, please use the Canvas discussions forum, do *not* approach teachers individually. But if you use the discussions forum, please follow this rule: you cannot ask anybody for explicit solutions or provide them with source code that does not work and ask them to help you spot or fix the error!

Traffic incidents

Every day, many people need to travel from one city to another city in the Netherlands and many of them choose to travel by car. One of the disadvantages of travelling by car, is that accidents (or other unexpected incidents) might happen, which might cause a considerable amount of delay. In this assignment we will consider an extremely stylized, simplified model for determining delays caused by incidents. We will include the influence of (smart) routing algorithms on the delay.

Incident data

On the NDW (National Data Warehouse for Traffic Information) website www.ndw.nu large amounts of data are available on travel times and speeds on the Dutch highways. This data is collected through detector loops, which are triggered by passing vehicles. As you can

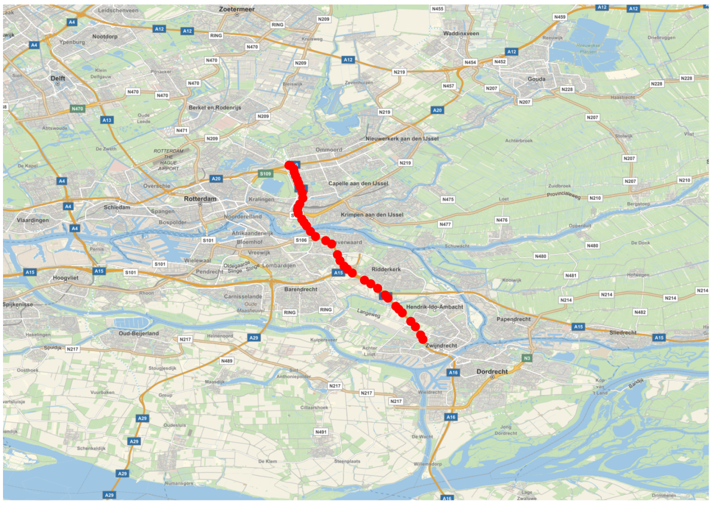


Figure 1: Detectors used to collect the incident data.

imagine, the total amount of data is huge (think of many terabytes). From this raw data we have constructed a dataset called `assignment4data.xlsx`, containing information about incidents that have taken place on the A16 highway between Zwijndrecht and Rotterdam (see Figure 1). The dataset is the result of a data analysis conducted on the raw NDW data between January and May 2017 and contains two columns. The first column contains the start (date + time) of the incident, and the second column contains the time when the incident ends. We use the following definitions;

- **Start of an incident:** The moment when the traffic flow and speed drop below a certain threshold level (not further specified here). This means that "regular" traffic jams due to overcongestion (mostly during rush hours) are also considered to be incidents, but only if traffic comes to an (almost) complete stop.
- **End of an incident:** The first moment after the start of an incident, when traffic flow and speed have grown to their normal levels again.
- **Duration of an incident:** The time (in minutes) elapsed between the start and the end of the incident.

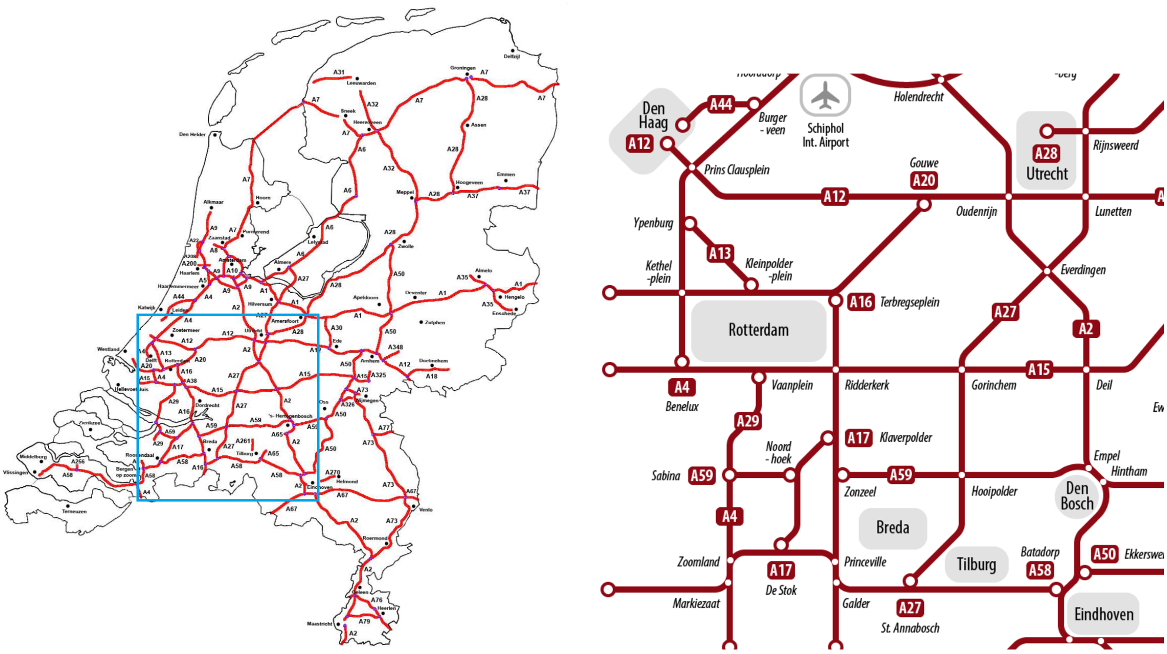
Question 1. Analyse the data. Use summary statistics, plots, histograms, etc. to study and provide insight in:

- the duration of an incident,
- the start time of an incident,
- the number of incidents per day,
- ...

Try to fit one (or more) suitable distributions to the duration of an incident. Interpret all your results.

Travel times

In the remaining part of the assignment, we will focus on the effect of incidents on *travel times*. To this end, we will focus on the Dutch highway network between the following cities: Eindhoven, Tilburg, Breda, 's Hertogenbosch (Den Bosch), Rotterdam, The Hague (Den Haag), and Utrecht. In Figure 2, a schematic representation is given of the highway network between these cities. In this part of the assignment, we are interested in travel times of people traveling from city A to city B (where A and B are different for each group, see Canvas). In more detail: assume that λ vehicles per time unit travel from A to B, using the highway network depicted in Figure 2. We assume that incidents happen randomly on each of the links in this network. A *link* is the stretch of highway between two major junctions. The durations of these incidents $I_1, I_2, \ldots$ are independent identically distributed random variables. Whenever an incident occurs on a highway link, vehicles will be delayed during the duration of this incident.



(a) Highways in the Netherlands

(b) Schematic representation of the network.

Figure 2: The Dutch highways. The blue rectangle indicates the part discussed in this assignment.

Question 2. In this question we ask you to write a discrete event simulation to estimate the travel time *distribution* for vehicles travelling from A to B. You are fully free to develop and implement your own mathematical model, but here are some assumptions, rules and/or guidelines:

- Focus on the highway network between the cities Eindhoven, Tilburg, Breda, 's Hertogenbosch (Den Bosch), Rotterdam, The Hague (Den Haag), and Utrecht. In fact, you are allowed to simplify this network, if you find the whole network too large (for example: remove the A13, A27, A29, A59, ...), but be sure to keep all the seven major cities and at least twenty links. Represent this network as a graph where vertices correspond to the cities/highway intersections and the edges correspond to the stretches of highway between these vertices.
- In this simulation you will need to have (at least) a network of servers/server pools. Hint: the number of servers might be infinite, depending on the model you choose.
- We assume that vehicles do *not* leave the highway until they reach their destination: they stay on the highway network.
- In this question, we consider the simplest form of vehicle routing: take the shortest path, i.e. the route with the smallest expected travel time. This is the travel time *without* taking incidents into account! You can regard this as users having an old-fashioned navigation device which does not receive any real-time traffic updates, so all vehicles take the same route from A to B. You should consult Google Maps (or a similar tool) to find realistic travel times for all of the links, under the assumption that there are no delays. Hint: in Google you can specify that you want to leave at midnight on a Sunday. You do not have to find real data about *variations* in these travel times, but we do expect you to add some randomness to the travel times.
- You can use the data from Question 1 to model the process of incidents on all of these links. Although the data from Question 1 was obtained from a short stretch of highway between Zwijndrecht and Rotterdam (more precisely: Klaverpolder and Terbregseplein in Figure 2(b)), you may assume that this data is representative for *all* links in the network.
- You are free to choose yourself how an incident translates to delay of individual vehicles. The simplest (but still acceptable) way to do this, is to add the remaining incident duration to the travel time of a link, for those vehicles that *start* driving on this link. Vehicles already present on that link remain unaffected.

We are interested in the following performance measures:

1. For individual vehicles: distribution of the travel time (mean, standard deviation, plot a histogram). Indicate which part of the travel time is caused by the delay. Present results on the number of incidents they encounter on average (again: mean, standard deviation, distribution).

2. For the network: distribution of the number of delayed vehicles at any arbitrary epoch. Formulated differently, we would like to see a table where you simulate for $k = 0, 1, 2, \ldots$ the fraction of time that exactly k vehicles are being delayed (consider it as being stuck in a traffic jam).

3. For incidents: distribution of the number of incidents in the whole network, at any arbitrary epoch. This is similar to the previous question, but now for the fraction of time that exactly k incidents are taking place simultaneously in the network.

Remark 1. *Note that, in order to determine the second performance measure (number of delayed vehicles in the network), it actually matters* how many *vehicles are driving in the* whole *network. So now you need to add random vehicles in the network that drive between the highway junctions. This part does not have to be implemented very realistically. You may even assume that vehicles enter the network randomly at each of the nodes, travel one link, and then leave the network again. For these vehicles,* no *origin-destination routing is required. Only vehicles travelling from A to B have a fixed route. Our advice is to keep it simple: do not introduce too many model parameters, like number of vehicles per hour travelling on each link. You are allowed to assume that this rate is equal for all links.*

Question 3. Extend the model from Question 2 by including the following feature in the model:

Assume that there is *one* roadside assistance company, located near a highway junction in the center of your network (for example Hooipolder or Gorinchem, see Figure 2). This company has K tow trucks that can be sent to the location of an incident to tow away the vehicle(s) causing the traffic jam. Assume the following:

- This company receives information about incidents instantaneously when they happen: the location and even the exact duration of the incident.
- If the duration of the incident is longer than the travel time from the home-location of the company to the incident, and a tow truck is available, the company sends one of its tow trucks to the incident.
- This tow truck is *not* affected by the traffic jam, because it passes the traffic on the emergency lane.
- As soon as the tow truck arrives at the incident, the incident is immediately cleared (this takes zero time). The vehicles on the highway are unblocked and continue driving towards their destination. The tow truck returns to its home location.

Determine, for several (positive, integer) values of K, how much the congestion decreases by recomputing the three performance measures listed in Question 2.

Question 4. We know that the aforementioned assumptions are highly restrictive and not very realistic. Obviously, in reality, shorter links may have less incidents and the delay of an individual vehicle would depend on how long after the start of the incident the vehicle joins the queue. In the next question we ask you to build a more realistic model.

Make the basic model of Question 2 (so without the roadside assistance extension) more realistic by adding *queueing dynamics*. In the previous question, there was no real queueing dynamics: all vehicles are delayed, but no actual queue is formed and when the incident dissolves, all vehicles continue their trip immediately. In reality, the duration of an incident consists of two phases when an accident has happened: the first phase, when one or more lanes are blocked due to the accident. During this part, the queue builds up. The second phase starts as soon as all lanes are cleared and traffic can flow regularly again. During this phase, the traffic jam (the queue) dissolves. The phase ends when traffic flows normally again and the traffic jam has disappeared completely. We expect you to use a queueing system to model the dynamics of the traffic jam when an incident occurs. Recompute all the performance measures mentioned in Question 2. Comment on how they changed and try to explain the changes. Note that Remark 1 applies here too: the lengths of the queues is significantly affected by the vehicles in the *whole* network.

Hint: in reality, traffic queues have a special type of dynamics (compare it to a traffic-light queue). The front of the queue might be driving, the middle part is standing still, and at the end of the queues vehicles are joining. You do *not* have to implement this level or realism, with acceleration and deceleration. Instead, use the regular queues that we have discussed: vehicles at the front of the queue are being served and depart from the queue when their "service" is complete. Vehicles at the end of the queue arrive according to a regular arrival process. The whole network becomes similar to the queueing networks we have discussed in Chapter 8 of the lecture notes, but the incidents (which can be modelled as a server that becomes unavailable for a while) make it more challenging.

More detailed guidelines can be found in Canvas, so check the rubric for more information. Urgent questions can be sent to m.a.a.boon@tue.nl. Upload your report in PDF format before the deadline specified in Canvas. **This is a hard deadline**! Please include your source code in a separate `.txt` file.