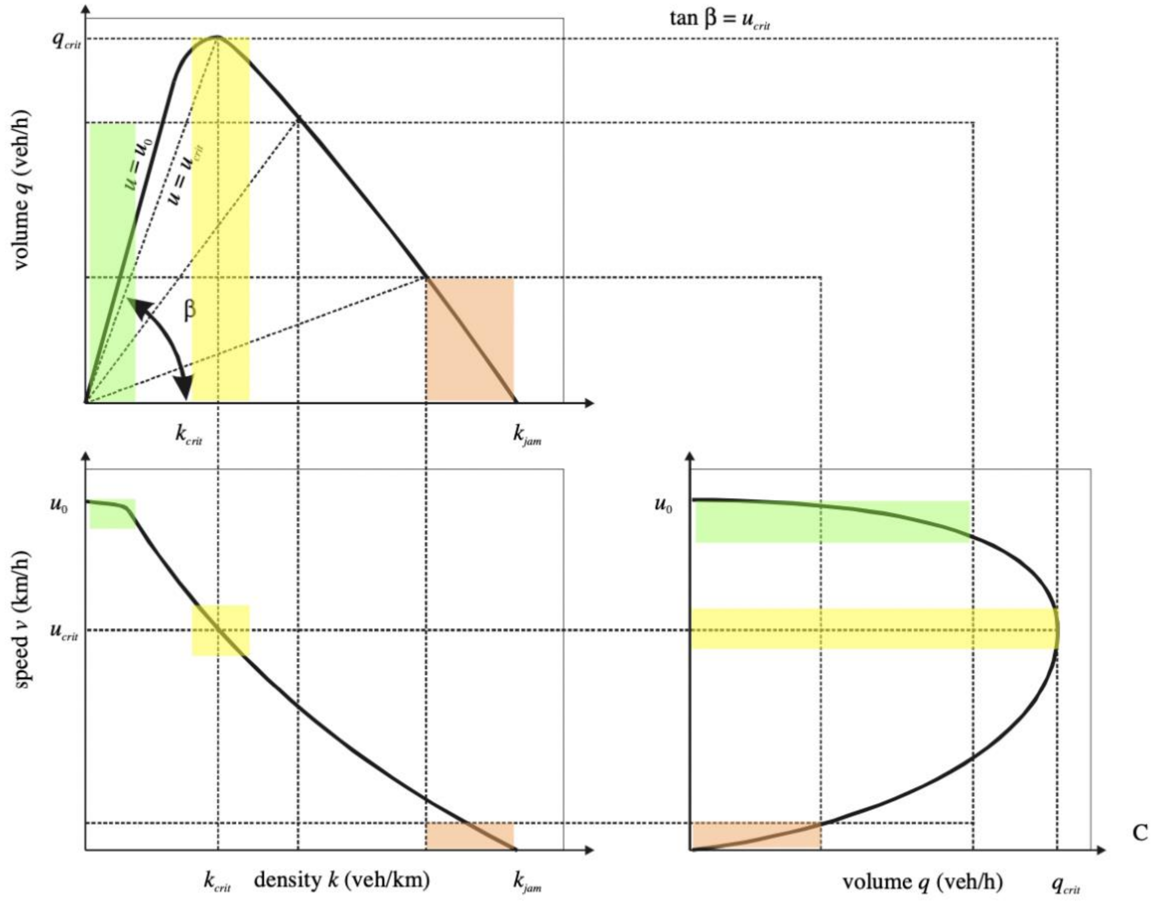
FAQ MSO CA Simulation Project

What is the meaning of optimum throughput?

There are three parameters that have an influence on the objective and subjectively perceived throughput on a motorway:

- The **density** of traffic (k), i.e. the number of vehicles per km of road.
- The **volume** or intensity of traffic (q), i.e. the number of vehicles passing per h.
- The **mean speed** of traffic (u) for vehicles, measured in km/h.

Density and volume typically refer to a single lane. The relation between these parameters is described by the so called fundamental diagram (source: [Traffic Flow Theory and Simulation, TUDelft](), chapter 4).



Obviously, density divided by volume equals mean speed. Under free flowing (green) conditions higher speed means higher volume of traffic with virtual no impact on density. The density of traffic determines the average free headway in front of a car and hence the theoretical maximum speed. When the density increases it starts impacting on the effective speed until it reaches the critical region (yellow) of maximum traffic volume. Any further increase of density results in a reduction of speed and hence the traffic volume until we reach traffic jam conditions (red).

We use the term **throughput** as cumulative volume across all the lanes going in one direction measured and averaged over a prolonged period of time. In the context of the CA the figure could be measured by averaging the number vehicles entering the 3 lane motorway section or equally the number of cars leaving the 2 lane section.

Optimal throughput is the largest sustainable throughput that can be achieved. It is ultimately an optimisation problem as it is unclear if that highest throughput can be achieved under the standard motorway speed limit of 120km/h or with a reduced speed limit. Common wisdom suggests a reduced speed limit, as the more homogenous driving style results in less disturbance and hence allowing a much higher density of cars per km. On the other side if the proportion of semi-autonomous cars using adaptive speed control increases, this would results in less speed variations and higher car density at higher speeds. Connected cars could also operate at a reduced safety distance because they could accelerate and decelerate 'in sync' or using a more global picture of the traffic situation hence reducing the speed slightly if there is a risk of congestion ahead and so keeping the overall traffic fluid without ever congesting.

The curves depend from many conditions, like technical parameters of the vehicles, the traffic mix, road surface and weather conditions, motorway speed limits and enforcement measures.
The general technical progress has resulted in an increase of the achievable maximum traffic volume per lane by 0.4 to 1% per year. In the 1950ies it was 2,000 vehicles/hour is now between 2,200 and 2,400 vehicles/hour. With regards to the critical speed the numbers in the literature vary, but there is an increasing body of knowledge that suggests a range between 60km/h and 100km/h.
Depending on your simulation parameters and driving strategies implemented, your curves will have slightly different shapes. For a practical application one needs to strive during the validation phase to match real world measurements. For the purpose of the CA it is sufficient that your values are in the same order of magnitude.

Overall this is an optimisation problem. There are only a few parameters that can be controlled. For example the proportion of heavy goods traffic (which could be banned during the rush hour), or speed limits at different times of the day. Essentially you will have to run for each of the different parameter values you can or want to control a number of simulations with different traffic volumes (i.e. different inter-arrival times) to determine the optimal throughput. Based on these simulation results you can determine what the optimal parameter setting is.

What different car types should be considered?

The simulation as developed in class is based on a number of system wide constants, like car length, maximum acceleration and deceleration, safety distance or reaction time, and uses a single driving strategy encoded in the common subroutines, like 'update', 'process' etc. Some of these parameters are specific for a particular type of vehicles (like HGV, light trucks or family cars, reflecting different engine power and torque, vehicle load, and driver support systems implemented). Other parameters seem more related to driver behaviour (like maximum velocity, safety distance, reaction time).

For this simulation you need to capture the effect of different types of vehicles and different driver behaviour.

You could try to capture aspects related to different 'types of vehicles' by using different subclasses of vehicles and using different initialisation constants. This would also allow to use different control strategies through changes in the code for 'update' or 'process'.

Aspects related to driver behaviour are probably better represented by instance of the vehicle, and passed in as parameter to the constructor.

You could also put both aspects together and maintain a single class 'Vehicle' with many more internal variables and hence parameters in the constructor. You could for example pass car length, acceleration and deceleration together with Boolean values for different types of intelligent adaptive speed control or types of driver behaviour as parameters into the constructor.

You create then for example a data frame 'X' of *randomly generated* input parameters giving inter arrival time (taking into account speed and safety distance), all sorts of vehicle parameters, together with driver behaviour parameters and drive the simulation by this single data frame.

The project description asks for studying the impact of fully electric vehicles (with about double acceleration and deceleration behaviour), and vehicles with autonomous or partially automated cars (with different driving strategies). If you only consider family cars in the morning rush hour, you may come to certain answers, which may differ depending on the speed limit set (still remember the attempt to set variable speed limits?).
However when you consider that between the family cars used in the commute, there is a large number of trucks, the outcome may well be different.
You choose the scenario you want to investigate, and clearly document the various cases by distinguishing fixed parameters and varying parameters (given as proportion of different vehicle types).

Some tips for organising your work

Don’t implement all these features at once. You might lose too much time chasing nice features. Instead make an initial project plan (estimating your available working time and compute time but leaving a bit of room at the end). Then make an initial decision what types of vehicles you want to support (may be just one or two different types), build your system, go through the verification and validation stages (don’t be afraid of reusing code from the class), collect the simulation data and write a draft report.
If this takes a bit longer than expected, this is no problem, as you have scheduled some reserve time.
If you manged through this stage and you still have time then analyse your simulation results and speculate what additional vehicle types or driver behaviour would give you the most interesting observations. Implement one or two more types (now you start to write serious code), go through the full simulation life cycle (you need to test your new code), collect more data and spend the rest of the time on polishing your report.
Keep in mind if you have two or more different types, you may need to run separate simulation runs for different proportions between different types of vehicles/drivers.
If you still have time, and are burning full of ideas, iterate again. But keep in mind: at this stage you have already something ready to submit, just in case your final attempt doesn’t work out in time.