Traffic Jam

Figure 1: Südosttangente, March 20th, 2019 3:30 pm at the Gürtel - Landstraßer Hauptstraße junction. (source: asfinag.at)

Motivation

Traffic jams during rush hour test the patience of drivers in Vienna, as in any major city. But how many vehicles need to be on the road to cause such congestions?

Modellbeschreibung

One possible model to implement is the Nagel-Schreckenberg model, which is based on the principle of a cellular automaton. For this purpose, the length of the road is divided into cells, each of which can accommodate a single vehicle. In each round, all vehicles first determine their intended direction, and then they are all moved simultaneously. Each vehicle can occupy only one cell at a time, so speeds are defined in fixed increments: 1 cell per time step, 2 cells per time step, etc.

In the one-dimensional model — a single-lane road — there are four update rules:

1. If a vehicle has not yet reached its maximum speed, its speed is increased by one (acceleration).
2. If the gap (in cells) to the next vehicle is smaller than the speed (in cells per round), the vehicle's speed is reduced to match the gap size. (collision avoidance)
3. The speed of a vehicle is reduced by one with probability p (dawdling factor), provided it is not already stationary. (dawdling)
4. All vehicles are moved forward according to their current speed.

In the multidimensional model, there are additional rules to consider:

1. If a vehicle is forced to brake, it can switch to the left lane if it is free. (Overtaking)
2. If there is enough space to the right of a vehicle, it should change lanes. (Merging)

Assignment

Task 1

Analyze the model analytically: What unit does the maximum speed have in the model, and how is it related to the speed of a car in reality? What speeds are actually achievable in the model? How would the model need to be extended to achieve finer granularity? For your implementation of the simulation experiment, define reasonable values for cell length, time step, and model maximum speed.

Task 2

Implement the single-lane model. The simulation should dynamically generate vehicles for the model based on the two parameters 'lag parameter' and 'entry rate', place them at the edge of the grid, move them to the other edge using the given rules, and remove them from the model there.

Task 3

Experiment with different values of the input parameters and various visualization options. Implement the model in such a way that for each vehicle removed from the edge, the time it took to complete the segment is recorded.

Task 4

Extend the single-lane model to arbitrary lanes. For overtaking and merging, first consider which surroundings of the vehicle need to be observed and choose appropriate checks.

Task 5

Choose a reasonable (what is reasonable in this context?) section of road in reality and, for example, search the internet for real data that describes average travel times through this section depending on the time of day. How does the data compare with your model results?

Task 6

Write an algorithm that calibrates the two parameters of the multi-lane model so that the average travel time in the model matches a given travel time. Use the results to parameterize your simulation with the queried real data. That is, the simulation should be able to dynamically change the vehicle density and thus produce the same average travel times as the queried dataset.

Task 7

Perform a proper face-validation of the model. That is, compare a dynamic visualization of your road with real recordings, for example, from traffic cameras.

Task 8

Use your fully parameterized model for scenario tests. How do the travel times change in the case of a (temporary) lane closure?

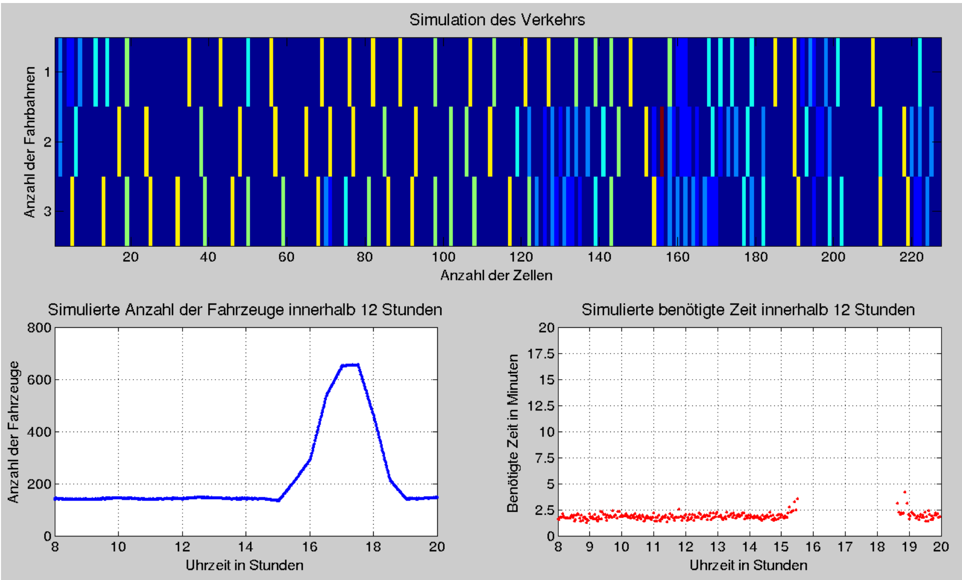

Figure 2: example picture: simulation of traffic, number of vehicles and travel time of a three lane road in a timespan of 12 hours.