Modeling and simulating traffic congestion

Steven Raaijmakers, 10804242

February 27, 2017

- **Rule 184: [1, 0, 1, 1, 1, 0, 0, 0]**
- r = 1
- k = 2

1. Because cars can only move when there is an empty spot next to them. The amount of cars will also stay the same, and since they will all move in the same direction, there will not be any congestion.

2. Figure 1: evolution of rule 184 with init_row.density = 0.4

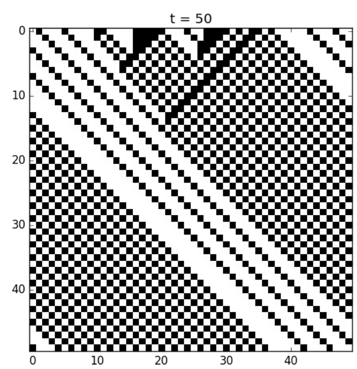
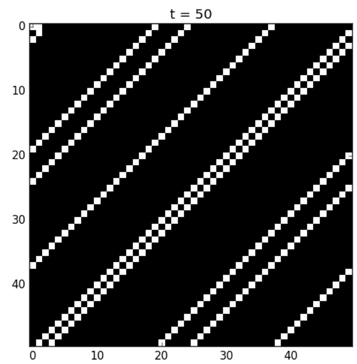


Figure 2: evolution of rule 184 with init_row.density = 0.9

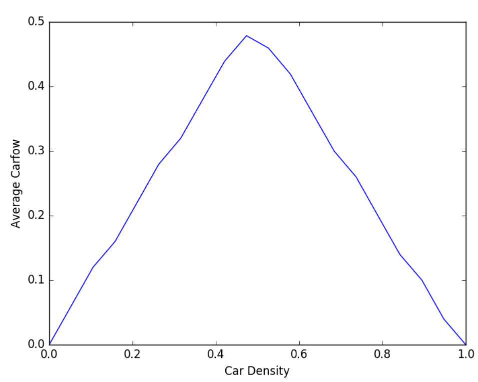


The evolution of the figure with density 0.4 has the cars (represented as the black dots) moving to the right, while the density 0.9 results in the cars moving left.

3. (a) Money
   (b) Time
   (c) No accidents
   (d) Less environmental impact

4. For 20 different densities between 0 and 1, with height (T) = 1000, width (N) = 50, R = 5.

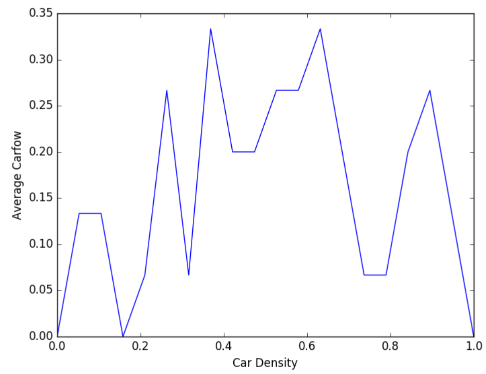
   Figure 3: Average carflow / density initial row



   There is a clear phase transition around density 0.5. The function increases before this density, and decreases after the same density.

5. For 20 different densities between 0 and 1, with T = 5, N = 50, R = 3.

   Figure 4: Average carflow / density initial row



The resulting plot shows a function which is much more angular. This is probably due to the transient length not being greater than T. In other words, the CA has not filtered out the randomness of the initial state, so there is no clear pattern yet.

6. (a) Code to find max of [x] and [y]:

```
def find_max(x, y, n_samples=100):
    samples = np.linspace(np.amin(x), np.amax(x), num=n_samples)
    return np.amax(np.interp(samples, x, y))
```

   This function finds the max of a function defined by two numpy arrays: [x] and [y]. It finds the min and max value in x and takes n_samples between these two values. After this the function interpolates between [x] and [y] and gets the corresponding values for each of the sample points. Of all these corresponding values it returns the max value.

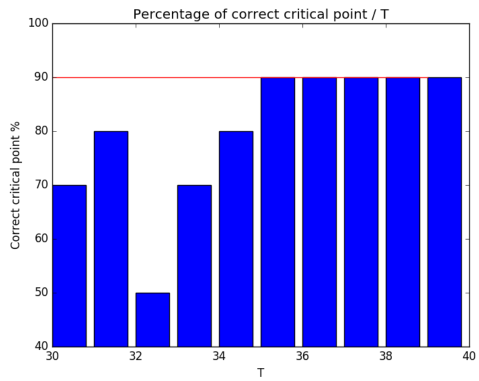
   For the input of question 4, it returns $density = 0.478$ for the max found car-flow and thus the point where the transition happens.

   (b) The suggested algorithm is implemented, but since it takes forever to run the script, I started from t=30, because while debugging I noticed none of t ¡ 30 had an significant correctness.

   The algorithm is in $O(N^5)$: There has to be looped in T; inside T the experiment has to be repeated; each experiment has multiple densities where there has to be looped over; for each density the experiment has to be repeated; for each of these experiments the carflow has to be found. Of course this could be written different so there experiment will be much faster.

   $T_{min} = 35$

   Figure:



This experiment was with R = 10 and R2 = 10. R2 is the number of repeated experiments for T.

7. Not much untill you have runned some experiments which shows the impact of gender on this phase transition. Same goes for carsize etc.