

Multi-Agent Systems

B. Nebel, F. Lindner, T. Engesser
Summer Semester 2016

University of Freiburg
Department of Computer Science

Exercise Sheet 1

Due: May 2, 2016, 10:00

Please note: Submissions are to be made to the folder `ex01` of your group's repository. While all theoretical questions have to be answered in one single pdf file, you're allowed to submit multiple Python source files (just make clear in the header of each source file, to which exercise it refers to). A brief explanation of the simulation framework is given in the file `README.md`, which you can find in the repository. For more detailed information on how the framework's components work, you can look directly into the documented source files.

Exercise 1.1 (Nagel-Schreckenberg Model, 2+2+2)

Nagel and Schreckenberg employ cellular automata to model traffic flow.¹ Your task is to represent a concrete traffic situation within the Nagel-Schreckenberg model and to predict the system's future behavior based on the model.

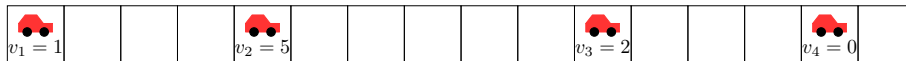


Figure 1: Some traffic situation in the Nagel-Schreckenberg Model

- Represent the situation depicted in Figure 1 as a cellular automaton.
- According to the model, what are the possible values for the velocity v_3 of car 3 in the next iteration? Consider the cases $p = 0$, $p = 0.25$ and $p = 1$ (given $v_{max} = 5$).
- Illustrate the most probable successor state for the next two iterations, given that $p = 0.1$, $v_{max} = 5$, and that car 4 (due to randomization) remains on its cell for the first iteration.

Exercise 1.2 (Nagel-Schreckenberg Model, 3+3+3)

Your task for this exercise is to implement the Nagel-Schreckenberg model within our multi-agent framework. You can find a documented implementation of a simpler traffic simulation in your repository (`ex01/trafficexample.py`).

- Implement the cars' behavior by implementing the four steps each car performs in each iteration. Your simulation is supposed to take the values of v_{max} and p , as well as the number N of cars (which are to be placed randomly on the track), as command line arguments.
- Run the simulation with different values for N , v_{max} , and p . Compute the fundamental diagrams of traffic flow (for at least 9 different combinations of v_{max} and p) and interpret/compare what you see. For plotting, you can for example use `PYTHON` or `GNUPLOT`. Add the computed diagrams and your discussion to your pdf submission file.
- Implement Velocity Dependent Randomization (VDR) with two different probabilities of random deceleration, p_0 for $v = 0$ and p_+ for $v > 0$. If the car stops it takes longer to re-accelerate, hence the higher randomization factor. What differences can be observed w.r.t. the simulation without VDR?

¹Check out the original article: http://www.pd.infn.it/~agarfa/didattica/met_comp/lab_20140108/1992_origca.pdf