Traffic Modelling – Real Road Networks

Description:

It is commonly observed that traffic jams and other similar traffic issues occur 'for no obvious reason'. These effects have also been observed in simple, one-lane models of traffic, where cars 'look ahead' to determine whether to speed up or slow down. The Perthshire Road Agency are responsible for scheduling necessary road and lane closures, for activities such as road repair and installing services such as fibre broadband. They have seen some results of simple models and are interested in understanding if and how they can be applied in more realistic situations. In particular, they are hoping to determine how such models work on real road networks, and how such modelling can influence their decisions on road/lane closures to minimise the overall impact on the road network.

Aims:

The Perthshire Road Agency wish to understand some/all of the following:

- Can the model treat roads with multiple lanes?
- If so, what does it tell us about the formation of traffic jams on single- vs multi-lane roads?
- How can the model be extended to(wards) real road networks, e.g., including junctions, merging lanes?
- Again, what does the model tell us when applied to a real road network?
- Can the model predict the effects of road and/or lane closures on the formation of traffic jams? Can it be used to determine an 'optimal' strategy if roads need to be closed?

An initial model:

Environment:

A one-dimensional road (array) of L sites with open (or periodic?) boundary conditions;

Agents:

Vehicles, with integer velocities between zero and $v_{\max}$.
Each road site may be occupied by one vehicle, or be empty.

Algorithm:

The motivation for the algorithm is that vehicles will accelerate towards a maximum speed limit, as long as this will not lead to them crashing into the vehicle in front. If they would do so then they decelerate. Also, drivers are not perfect at doing this, so some random deceleration is included in the model.

For an arbitrary state, one update of the system consists of the following four steps, performed in parallel for all vehicles:

1. **Acceleration:** if the velocity v of a vehicle is lower than $v_{\max}$ and if the distance to the car ahead is larger than $v+1$, increase the speed by one $[v \mapsto v+1]$.

2. **Deceleration:** if a vehicle at site i sees the next vehicle at site $i+j$ with $j \leq v$, it reduces its speed to $j-1$ $[v \mapsto j-1]$.

3. **Randomisation:** with probability p, the vehicle's velocity is decreased by one to a minimum of zero $[v \mapsto \max(v-1, 0)]$.

4. **Motion:** each vehicle is advanced v sites.

You'll need to decide what to do at the boundaries of your domain (i.e., how to add/remove vehicles).

Reference:

'A cellular automaton model for freeway traffic', Kai Nagel and Michael Schreckenberg, Journal de Physique 1, **2**(12), 221-2229, 1992