Parallelized Traffic Simulation

Denzel Farmer (df2817)   Andrew Yang (asy2130)

Fall 2023

1 Introduction

A nearly universal experience as a driver is getting stuck in traffic. Congested highways often slow to a near stand-still while advertising speed limits of 60-80 mph. These bottlenecks can appear for seemingly no reason, and can prove highly frustrating to students returning from a relaxing Thanksgiving vacation.

With this in mind, we plan to create a highway traffic simulator, which will hopefully help to demonstrate some of the counterintuitive chain reactions that can occur on the highway. With some simplifying assumptions, and by taking advantage of existing driver behavior models, we believe this simulator can be highly parallelized to allow fast simulation of scenarios involving potentially hundreds of vehicles.

2 Objective

We plan to build a highly parallelizable traffic simulator, to observe the flow of highway traffic as it changes over some time period, based on configurable parameters.

We will simulate highway traffic microscopically, considering the behavior of each individual driver based on models meant to emulate human behavior. Based on this simulation, we will be able to observe how traffic flow responds to random stimuli (for example, a car briefly slowing down to avoid an obstacle).

To build our simulator incrementally, we will begin with a simple scenario and increase complexity, as time allows:

1. A single lane of highway traffic
2. Multiple lanes of traffic, including a lane-switching algorithm
3. Multiple lanes with lane-switching and an on-ramp for secondary inflow
4. Time permitting, a comparison between scenario 3 and one with an on-ramp merging into the leftmost 'fast' lane rather than the rightmost lane[1]

[1] This scenario is based on a particularly frustrating real-world intersection, where Wilbur Cross Highway merges onto I-91 southbound near Hartford, CT.

3 Proposed Algorithm

3.1 Overview

Our simulator will run based on some initial configuration (including, for example, the car inflow rate, a set of driver configurations to assign randomly to each newly generated car, and a number of time steps to run). After the simulation concludes it will dump its output to a CSV file, containing position information for each car at each time step. This information will then be processed by code that parses the CSV and converts the simulation data to a video using basic graphics.

To perform the simulation, we will iterate sequentially over each time step. Within each time step, we might generate new cars based on the inflow rate, remove any cars passing out of the simulation bounds, and compute the next state for each current car.

We will use the car-following and lane-switching algorithms described in the next section to calculate the desired acceleration for each car and whether or not it should initiate a lane switch during any time step.

We will also allow a configuration that will temporarily slow down a random car during a particular time step, and will introduce random variation into the speeds and driver configuration during the generation of each car. Because of these randomized characteristics, we expect different behavior after running each simulation even in the one lane scenario.

3.2 Car Following Algorithm

We plan to use a car-following driver model that takes in position and speed information about a car and the car directly in front of it, and decides how much to accelerate/decelerate at each time step to maintain a safe following distance.

In particular, we plan to implement the Intelligent Driver Model described in chapter 11 of Traffic Flow Dynamics [1].

3.3 Lane Switching Algorithm

We would also like to extend our model to include the MOBIL lane-switching algorithm implemented in the linked JavaScript simulator [3], described by the referenced paper [2].

This model takes in information about neighboring cars and decides based on their position and the driver's configuration parameters whether or not to switch lanes. This includes parameters tuning how 'polite' a driver should be, for example, and whether they prefer a particular lane.

4 Parallelization

At each time step, we calculate the instantaneous acceleration of each vehicle based only on the one in front (using the IDM model), and whether or not a lane switch should be initiated based on the neighboring vehicles (using the MOBIL model).

Since each car's next state can be calculated independently (based only on the previous state of surrounding cars), and since we believe the bulk of computation time will be spent calculating each car's next state, we should be able to significantly speed up the simulation by running these calculations in parallel.

Even with this optimization, we will still need to run each time step sequentially; we can only move on to the next time step after all the calculations for the previous time step are complete. However, since each next-state calculation should take roughly the same time, we hope that this will result in relatively equal utilization for each core.

It might be possible to run certain calculations within the next time step in parallel with those from the previous, if we can be certain that there is no dependency between the calculation (i.e. the cars are far apart). However, the overhead and complexity of checking dependencies might outweigh the speedup gained by running these calculations in parallel.

5 Evaluation

We can evaluate the correctness of our simulation by comparing the results to existing simulations which use the same driver models [3]. We can also configure the simulation to demonstrate traffic patterns well documented in the real world.

We will evaluate the effectiveness of our parallelization by comparing runtime of a parallelized simulation over a large number of time steps to the same simulation (with the same random seeding) run purely sequentially.

References

[1] Treiber, Martin, et al. "Chapter 11: Car-Following Models Based on Driving Strategies." *Traffic Flow Dynamics: Data, Models and Simulation*, Springer, Berlin, 2013.

[2] Kesting, Arne, et al. "General Lane-changing model Mobil for car-following models." *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, 2007, pp. 86–94, https://doi.org/10.3141/1999-10.

[3] `https://github.com/movsim/traffic-simulation-de/tree/master`