Multiple Lanes Traffic Simulation

Tomer Eldor
Minerva Schools at KGI

Part 1: Traffic on a Single-Lane Circular Road

First, let's simulate a basic case of traffic jams forming on a single-lane, circular road. Even if all cars have enough space to drive in front of them, humans sometimes randomly slow down (from breaking too much, or hearing the notification sound on their phone, not to mention checking their text messages or snapchat). These slow-downs cause a chain reaction of the cars behind to slow down more, thus creating a jam out of nowhere, and only the car in the front of the jam can be freed and accelerate back to normal speed. The jam itself seems to be traveling "backwards" in the road, passing to the cars behind, like a traveling wave. This behavior was shown to exist in trials with human drivers. Can we model it? How does it depend on the density of cars?

Single Lane Model Specification and Assumptions

This model is implemented as described in the original paper pioneering this traffic flow example: Nagel, K., Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. Journal de Physique I, 2(12), 2221–2229, available at this link.

The model is a cellular automata model: a collection of cells, each cell being at a particular state, which we inspect its progression through time. The cars are randomly initialized on the road with random speeds between zero and five, and then follow four basic rules of progressing. These are, put more intuitively:

1. If a car has enough space in front of it *(gap < speed + 1)*, it will accelerate by one, until reaching maximum speed.
2. If not (*gap < speed*), it will slow down enough to not crash into the car in front of it (not reach its location by the next time step);
3. Cars may randomly slow down by one *(with p(slowdown): speed = speed – 1)*

Then, (4) cars advance according to this updated speed.

This model has many simplifying assumptions. First and foremost, it operates in a *circular, single lane road.* This doesn't happen very often in reality. Second, it assumes that all drivers and cars are the same, with the same set of logical rules for accelerating and decelerating. It doesn't consider any cars of varying sizes, average and maximal speeds, or drivers with different heuristics or behaviors than the update rules listed. It assumes that the road is smooth, and is a closed-world road, with no obstacles, turns, lights, or problems. It assumes that drivers always accelerate by 1 only or decelerate to reach where there was a car in front of them in the previous time-step, whereas in reality I would think that most drivers predict where the car in front will be next and progress in relation to their future speed. In fact, it might be a different set of rules altogether, such as keeping constant distance from the car in front of you and behind you It is indeed very limited, but it has surprising merit in modeling some aspects of real-world traffic quite well.

Single Lane Model Results: Road View

```
Traffic Simulation with density=0.3, p_slowdown=0.3
4011......3.3......3.340..424.....13........1.....
000..2.....1...3...0.000..00.....50....4......2...
00.1....3...1.....30.00.1.0.1....0.1.......4.....3
0.1..2.....3.1....00.0.1.10...2..0...2.........4.0
.1..2...3..0...2..0.1.1.10.1....2.1.....3......0.0
1..2...3..2.1....2.1.1.10.1..2..0...2......3....1.
..2..2...2.1..2..0..1.100...2..2.1.....3.......4.1
.2..2...30...2..2.1..100.1...1.0..1........4....1.
2..2...30.1...1..1..2000...2..1.1...2..........4..
..2...300...2...2..2000.1...1.0..1....2..........2
1...2.00.1.....3..20000...2.0..1...2....2.........
..2..10.1..2.....200000...0.0....2....3....3......
....20.1..2...3..00000.1...1.1......3.....4....4..
..5.00..1....3..20000.1..2..1.1........3......4...
.50.0.1..1.....20000.1..2..2.1..2..........4......
.0.1.1.1...2...0000.1..2..2.1.1...2............4..
30.0.0...2....3000.1..2..20..1..2....3............
0.1.1.1....2..0000...2..20.1...2...3.....4........
0.0..1..2....2000.1....20.1.1....2.....4.....4....
.1.1..1...2..000.1..2..0.1.1..2.....3........5....4
..1..2.1...1.00.1.1...20..1..2...3......4.......40
...1..1..2..10.1.1.1..00....2..2.....4.......5..00
1....2..2..20.10..1.1.0.1.....2...3.......5...1.0.
..2....2..200.0.1..1.10...2......3...3.......3.1.1
1....3..1.00.1.1.1.0.0.1.....3......3...3.....1.1.
..2...1.0.00..10..1.1.1.1........4....2.....4.0..1
1....3.1.100..00..0.0.0..1...........4...3...1.1..
..2...1.100.1.00...1.1.1..1.............3...3.1.1.
2....3.100.10.0.1..0.0...2..2..............30..1..
..2...1000.00.0...2.1.1...1....3...........00...1.
Finished. Density: 0.3, traffic_flow: 0.1,
 avg speed: 1.25, stopped cars 0.36
```

Figure 1 – 1 Lane traffic simulation over time.

For a model with this extent of simplicity, the road is surprisingly able to mimic the behavior in reality, where jams of cars stopping (at speed '0') seem to be "traveling backwards". This happens in reality and in the simulation by cars randomly slowing down excessively and having cars slowing down more from behind, small traffic jams are formed. Then, the first cars are released from the front sequentially, whereas new cars are joining the back of the traffic jam. You can see that demonstrated in Figure 1, simulated at car density of 0.3, meaning 30% full road. In this model, we initialized cars randomly with this probability, so the exact number of cars ranged *around* 0.3. Empty spaces are represented by dots ("."), and the numbers represent the speed of the car, from 0 (stopped car) to 5 (maximum speed).

How does the road perform when it gets more packed with cars?

First, to test performance we need to define what is the metric we care about. There are a number of interesting metrics, like the average speed of cars on the road, or the number of jams; but first let us focus at a measure that combines both of these and expresses the outcome of the efficiency of the road as an instrument to transport as many cars as possible through it as quickly as possible. We measure this with "Car Flow" – defined as the number of cars passed a certain location (we check at the boundary) *in average per time-step*. As more cars get on the road, the more cars *can* pass through it and thus increase its efficiency; but at some stage, there would be *too many* cars on the road such that traffic jams would be formed (assuming some cars might slow down and create jams) so that the overall speed of traversing the entire road per car is inhibited and resulting traffic flow is reduced. The real question is, *what density of cars is the optimal, beyond which total car flow would decrease?*

Single Lane Traffic Flow for Varying Densities

To test this, I simulated the road for each density between zero and one in increments of 0.01, repeated 100 trials for each such value, for a probability of slowing down of 0.5, road length of 50, and over 200 time-steps per each run. These results show a peak traffic flow around densities of 0.10 – 0.15. **Meaning, we can have optimal flowing number of cars through the road if the road is between 10% – 15% full of cars.** These results resemble the results of the original article by Nagel and Schreckenberg (1992), who found the optimal density to be around 0.1 reaching a traffic flow around 0.3, and compared to observations from real-world traffic data where optimal flow peaked at about 0.2 and has a similar sharp rise and slow decrease shape to it (Fig. 5 of Nagel and Schreckenberg). The peak in this location is not surprising, since it means the maximal number of cars that can travel at maximum speed in the road, before more cars are added would cause slowing down. That number of cars could be calculated by dividing the length of the road by max_speed + 1, since a car needs to be speed+1 cells away from the next car for maintaining full speed. For example, in a CA with 100 cells, that would be 100/6=16.67.

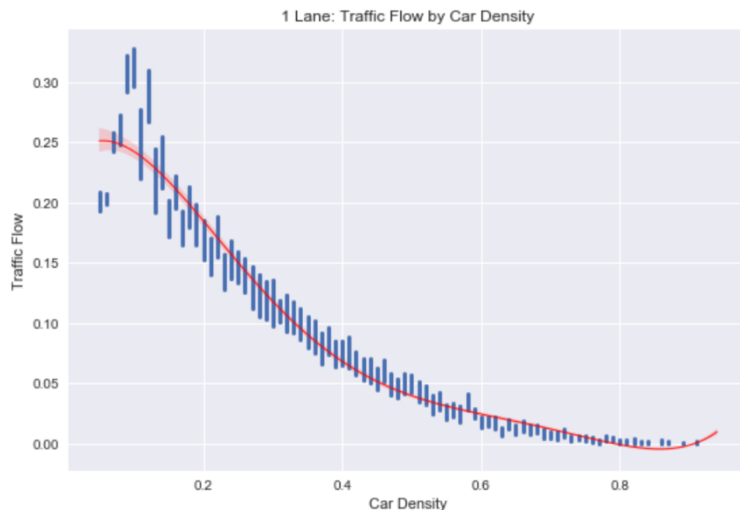


Figure 2 – 1 Lane: Traffic Flow per Density, p_slowdown = 0.5

What if our drivers drive better with fewer random slowdowns?

I would assume that in many countries (at least European or western developed countries), most drivers don't randomly slow down as much as half of the time (p_slowdown = 0.5), the values inspected earlier here and potentially in Nagel and Schreckenberg (1992). Let's inspect what happens when the slowdowns are limited to about 0.1-0.3 percent of the time. This could be viewed as better driver behavior or even closer to that of computer-assisted or self-driving cars (which would probably have close to 0 probability of randomly slowing down). Will this improve our results? Simulating and plotting with p_slowdown = 0.1, the peak increases by 0.05 to be around ~0.016 and increases in height, *which resembles the real-world data much more closely*. This could also hint on one already clear advantage of self-driving vehicles which may have probability of slowing down of 0: this would further improve the traffic flow per for any density level.

Figure 3 –1 Lane: Traffic Flow per Density, P_slowdown = 0.1 in green, 0.5 in blue

Measuring Average Speed

Let us also look other metrics of performance. Looking at the average speed of all cars on the road (over 100 time-steps) by cars density, it naturally begins close to maximum speed as car density is low and cars almost always have much space to drive at full speed without seeing a car in front of them within 5 spaces, and then gradually decreases as the road gets denser with cars. When the random slowdown probability is 0.1 (in green in Fig. 4), the cars begin around maximum speed of 5, staying high almost constantly until ~0.15, then plunge down most sharply until density of about ~0.3 when they reach average speed of 0.1. Conversely, in the more stochastic model when p_slowdown is 0.5, cars begin at an average speed closer to 4 because of that random decrease by 1 (and its chain reactions), and plunge down more quickly around 0.08, most sharply until reaching an average speed of 1 at density of 0.2. When there is a high probably for random slowdowns, cars decrease their speeds more and create more chain reactions quicker, so they affect the total average speed of the network faster, thus the quicker plunge. This suggests that increasing random slowdowns impact average speed (and traffic flow, less directly) negatively and nonlinearly (with decreasing marginal utility), like increasing car density does. More specifically, by removing most but even not all of the random slow-downs (for example by computationally assisting human drivers or improving driver behavior), we could *have an additional 10% of road occupancy before the average speed plunges to 1*.

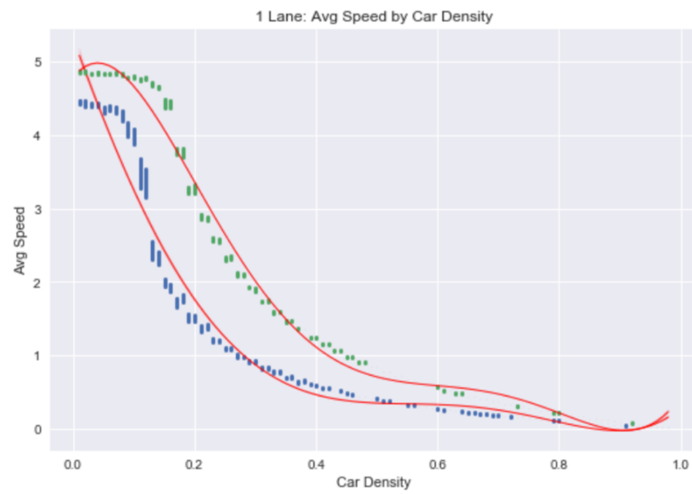


Figure 4 – 1 Lane: Avereage Speed by Density. p_slowdown: Green = 0.1, Blue = 0.5

This model showed interesting results, but it neglects many aspects of real roads. One important aspect is having *multiple lanes*. Let us examine what happens with multiple lanes.

Multiple Lanes Traffic Flow Simulation

Model Definition and Assumptions

The multiple lanes model bases of the single-lane model, where each lane acts according to the same rules and definitions, but with some extensions. First, there are multiple lanes. Each car might want to switch lanes, if it can’t accelerate because of a car in front of it. The car looks at the *gap* between itself and the car in front, defined as the number of empty spaces between them (so, 0 if there is a car right in front, in contrast to the single-lane article which defined distance as 1 for that base case originally). For switching lanes, (after seeing a car in front of it), the car will check the following conditions are met:

1. Look ahead in the other lane to see if there is a car that would block it from accelerating (speed + 1) cells. If there is a blocking car, don't switch.
2. Look back in the other lane. If there is a car in any cell up to [maximum speed] cells behind it, don't switch. This is a highly cautious behavior, assuming that a car might accelerate up to maximum speed in the next time-step and you wouldn't want it to crash into you!
3. With probability p_switch, cars switch onto the other lane.

I implement a symmetrical version of this model, where cars might switch to lanes at either of their sides. If the first lane checked (randomly chosen right or left) isn't available, the car will inspect the other lane and will switch into it if possible. This model uses the same parameters as the single lane, with the addition of: number of lanes (n_lanes) and probability of switching into them (p_switch). This symmetrical version assumes that there is no preference as for which lane to switch to, which might only be true in certain countries and cases, as many countries have a preference or even a law for cars to stick to the right and only use the left lane to bypass (in a right-side driving country).

This model still has many simplifying assumptions

What are the assumptions, parameters, and update rules of the model

Visualization of Model States Over Time

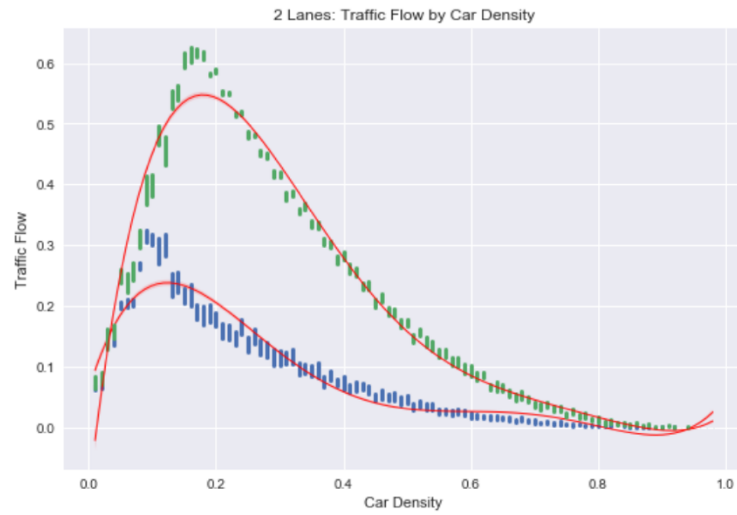
The model states are visualized like the single lane model, but with two lanes and a black line representing a one time-step progress. Here I mark when a car is switching lanes with a declaration and printing out the previous and next locations of the car switching lanes.

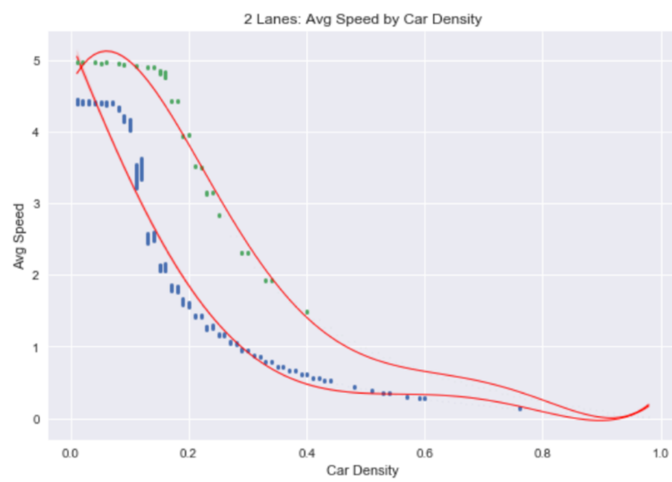
```
Traffic Flow 2 lanes Simulation with car_density=0.3, p_slowdown=0.5, p_switch=0.8)

.0....2.1.....4...1...3..2..10.1.1.......1..10.1..2...20....
..3..0.1.......000......3...100.....5..0.00....4...30.1.....

..1...0...2......3.1....2.1.0.1.1.1.......1.00...2..2.00....
...1..1..2.....00.1........3000......1.0.0.1......30.1..2...

....2.0......3...0..1....1.10.0..1.1.......10.1....20.00....
....1..1...2...0.1.1.......0000......0..10...2....00..1...2.

....0..1........3.1..1....10.1.1.0..1......0.1..2..00.0.1...
.3...1...2...2..10..1......0000.......1.0.1.....3.0.1..1....

.....1..1.......0..1..1...0.10.0..1..1......1.1..1.00.0...2.
...2..1....2..1.00....2....0000........1.1.1....0..1..2..2..

.3....1..1.......1..1..1..0.00..1...2.1.....0..1..100..1....
3...1...2...1.0.00......2..0000........0..1..2..0....2..2...

....3..1..1.......1...2..2.100....2..1..2....1...200.1...2..
..2...2....30..100.......1.0000.........1...2..2.1.....2..2.

.....1...2..2.......2...2.100.1....1..1....3..1..000...2...2
.3...3..2..00..000........1000.1.........1....2.1..2.....2..

..3...1....2...3.......3.100.1..2....2..2....2.1.00.1....2..
....3..2..20.1.000........000.1.1..........2..0..1...2.....2
_______________________
Car Switched Lanes!
original lane: 0/1, cell: 37/59, speed: 2, other_lane: 1/1

..3...1....2...3.......3.100.1..2.......2....2.1.00.1....2..
....3..2..20.1.000........000.1.1....2.....2..0..1...2.....2
_______________

....2..1.....2....3.....1000..1....3......2..0.0.00...2....2
.2....2.1.00..1000........00.10...2....2.....2.1...2....3...

Finished. Density: 0.3, traffic_flow: 0.3, avg_speed: 0.986437908497, stopped_cars 0.0
```

Performance of 2-lane Model and Ideal Deterministic Drivers

Here I wanted to inspect the performance of the model while comparing relatively bad driver behavior with random slowdowns of p=0.5, compared to a deterministic NO random slowdown model (which could be thought of as ideal, or self-driving vehicles ideal, form of driving). From here on, the blue plots represent the human stochastic p_slowdown = 0.5, and the green plot represents the perfect deterministic model with p_slowdown = 0.0.

The traffic flow of the 2 lanes shows similar trends to the single lane model, but with much higher traffic flows. The peak which previously was around similar density levels around 0.1, reaching traffic flow of 0.075, yet now reaching a traffic flow of 0.3 for the p_slowdown = 0.5. We see significant improvement for the perfect, non-slowing-down drivers with the green plot, reaching a peak height of 0.6 traffic flow around a density of 0.15 – 0.18! This suggests that the 2-lane model works similarly, with increasing density initially increasing flow but then sharply decreasing it, but the advantage of being able to switch lanes helps cars move faster through the entire road overall. The improvement of the deterministic model is significantly better than the more stochastic driver behavior.

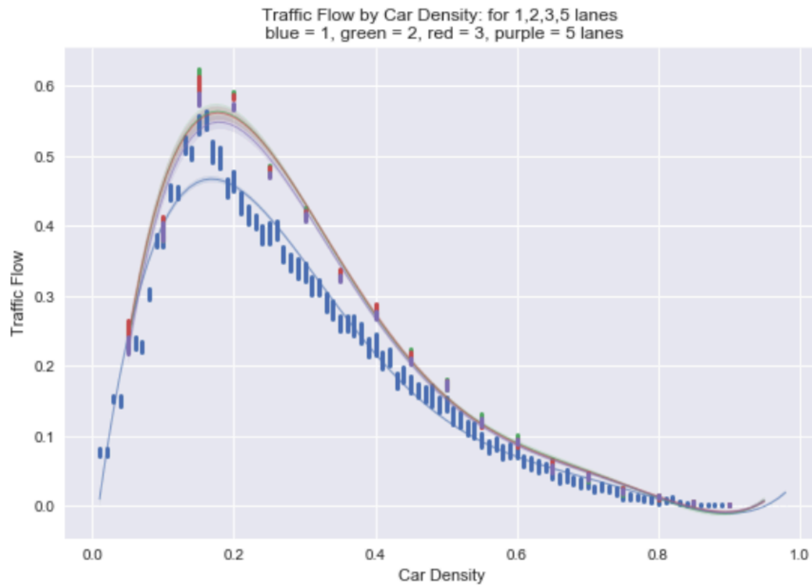




In addition to traffic flow, the average speed is again starting significantly higher for the less stochastic model and plunging down more slowly and starting only later. As expected, the perfect model's performance is as good as one could hope, starting at speed = 5 and only decreasing at density of 0.18.

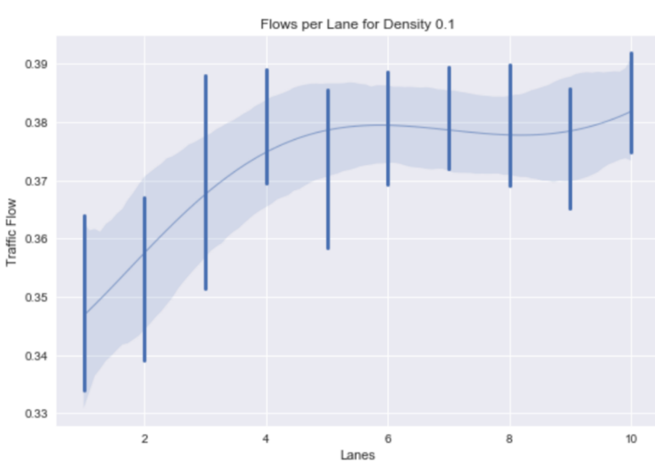
How Much More Traffic Can Flow Through Two or More Lanes?

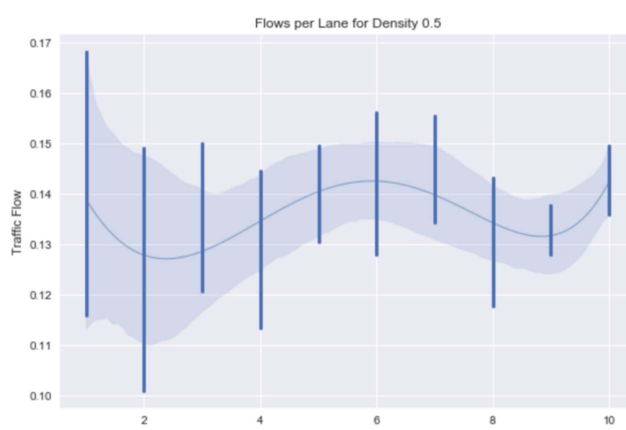
What is the effect of adding an additional third lane? Would it be as significant as adding the second lane? And how about having 5 lanes? One might expect that adding more lanes would increase traffic flow, potentially with some slight, and because decreasing marginal utility. However, the model's results show that 3-lane model results are not visibly better than those of 2-lane model. They are both better than the single lane model, but it seems that adding another lane didn't add much. I suspect several reasons for that: first, the majority of the effect would be applied by allowing *some* cars to bypass blocking cars for the first time, rather than being completely stuck behind them without any choice. Once you had an option to bypass, the chances that you could go into your left lane are about the same as the chances you can go into your right lane. The restrictions before switching are quite strict, and therefore the number of switches isn't high, and seems to be relatively

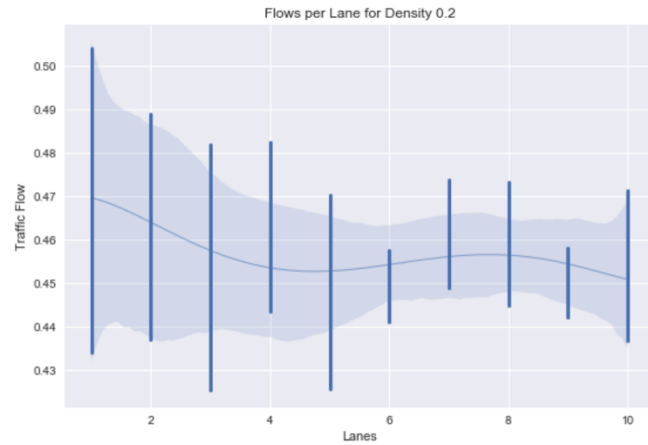


consistent when divided per lane while adding arbitrarily many lanes. This is because not only the road should be clear ahead, but the looking-back rule, where there shouldn't be a car any cell up to maximal speed backwards, is highly restrictive. Then, the random probability of not switching further decreases the occurrences of switching (although I used p_switch = 0.8 for all results here, seeing that the other rules are rarely met anyway). The reason switches are so rare is basically that, depending on density: either the density is very low, where you could probably switch if you wanted to, but you probably don't want to switch because your lane is clear; or that density is too high, meaning you have blocking cars ahead of you but also too many cars around in the other lane. Thus, a more viable model for switching might be an *asymmetrical model with varying densities per lane*, which is more like real world countries where the right lane (assuming driving side is right) is the primary and slow lane, and the other lanes are meant for bypassing, and thus by definition contain fewer vehicles and only of higher speeds.

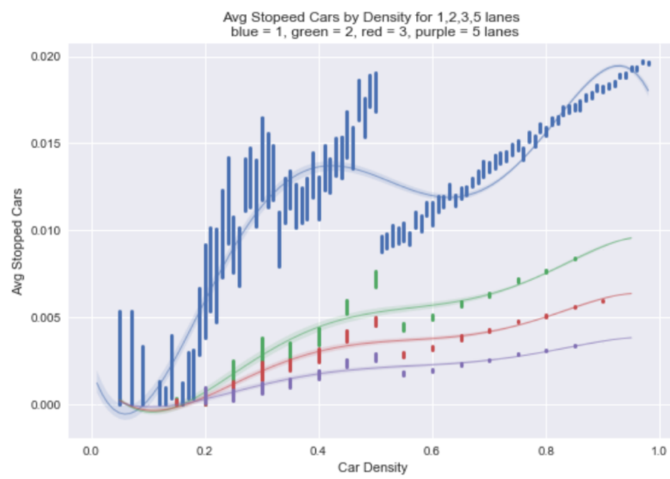
The differences in performance are more visible when the p_slowdown is low and the model is closer to deterministic. There, the gap shows that **traffic flow may be up to 0.2 higher on a multilane road; but not significantly higher on any more lanes than 2-lane road.** When the probability of slowing down is high, randomness ruins the advantage in adding lanes and the chances for cars to switch lanes, so that their performance is very similar. Thus, you can see in the following figure that the traffic flow of the blue single lane model is the only distinctively lower one from the rest, whereas all the other traffic flows of the 2, 3, and 5 lanes models are quite the same (when divided by number of lanes).



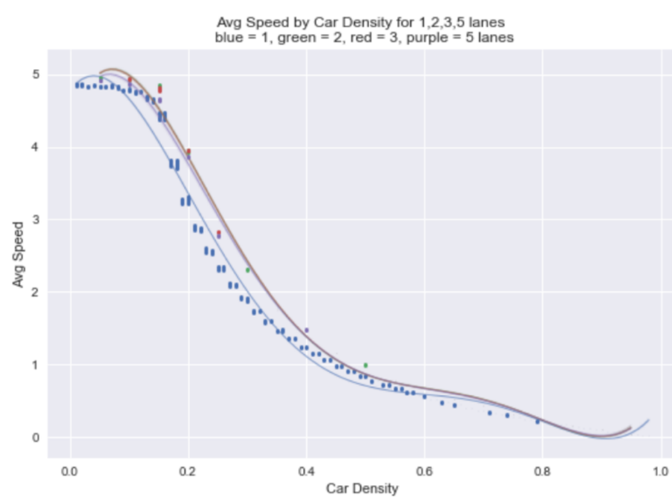




When increasing the number of lanes for up to 10 lanes, the resulting flows improved when the density was low, but when the density was higher than peak, the resulting flows did not improve consistently or significantly. Mainly, the distribution for resulting flows per density tested shrank. This makes sense simply because the larger number of cars, lanes, and thus samples we get. Notices in the figures that only for density 0.1 there is a consistent increase of flow by number of lanes. At 0.2, around peak density, the value of flow just converges to its lower end. **This suggests that if planners are after increasing traffic flow, it would be more impactful to reduce the density (occupancy) of cars on the road, than to increase the number of lanes while having the same number of cars per lane** (since this traffic flow is measured per lane)**.** Of course, if we will expand an existing road, number of cars per road is not likely to increase as much as we increased the road capacity, so **expanding the road could thereby reduce density and serve as the method for reducing occupancy and improving traffic flow.**



An example metric which does seem to have a significant effect of increasing the number of lanes is not so much the average and overall speeds, but the *proportion of stopped cars at a given timestep*. That was significantly reduced by adding more lanes. This is important to know, since while the effect of the average cars flow and average speed concerns the macro, outside planner scale, the number of stopped cars concerns the micro-scale – how many completely frustrated drivers do I have on my road? Drivers who are at a complete halt would get much more irritated than merely slowing down, and the longer they stay, the more irritated they get. These situations might lead to frustration, and in some countries (like Israel) might lead more to road-violence. If city planners care also for



their citizens' mental health (and potentially physical health), they should also consider minimizing the number of cars completely at halt.[1]

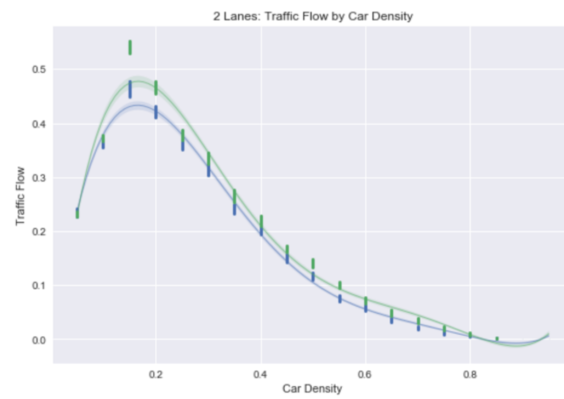
---

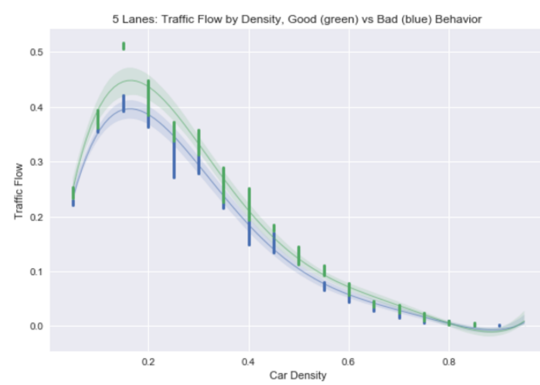
[1] The leap down at density = 0.5 was due to some malfunctioning of the dataset formed from 2 different iterations, which I didn't have time to fix before submission. The true model looks more linearly and continuously ascending, but I couldn't replicate those graphs in time before submission.

Bad Driver Behavior

Previously, we've tested how bad drivers (e.g., with more distractions such as reading more SMS messages, or simply driving inconsistently) with higher probability of randomly slowing down decrease the flow rate and average speed. Let's see what is the effect of worse driving behaviors.

I modeled the behavior of lanes switching more akin to the local culture in Hyderabad: *no looking backwards before switching lanes.* In this case, we would expect cars coming from behind to break more suddenly, more strongly, and more times. I therefore measured the same metrics of the good and bad models for a model with 2 lanes, same parameters as before[2], but changing only the bad driver behavior.





The results show **that bad driver behavior impacts negatively (around 10%) on traffic flow around peak density,** but when density is too low or high, the effect is reduced or not significant. We can see that for 2, 3, 5, and 10 lanes models examined. I attach here some of the figures. **This further shows us that improving the density of cars on**

---

[2] Parameters were: `(densities=densities, n_lanes=2, p_switch=0.8, road_length=50, p_slowdown=0.1, repeats_per_dens = 30, timesteps = 100,  max_speed=5, show = False, bad_switching=False/True)`

road would generally have a more significant impact on traffic flow than improving driver behavior or than adding more lanes if they will not reduce the total density (i.e., if many more cars choose to drive in that new and expanded road). With more lanes, **the average speed was harmed more, average stopped cars and the number of breaks was increased significantly with the bad driver behavior when increasing the number of lanes.** Average size of breaks wasn't changed significantly. This means that the impact of bad driver behavior is noticeable on the general speed and flow, and also on the number of stopped cars, and that we should prevent bad driver behavior if we want a better, smoother road driving.

Data:

```
2 Lanes:
Avg Breaks Size difference with bad lanes switching:  -0.000441142460694
Avg Speed difference with bad lanes switching:  -0.0482477036023
Avg Stopped Cars difference with bad lanes switching:  -0.000228893914819
Num Breaks per Step difference with bad lanes switching:  0.0
Traffic Flow difference with bad lanes switching:  -0.016798245614

5 Lanes:
Avg Breaks Size difference with bad lanes switching:  -0.00718493678693
Avg Speed difference with bad lanes switching:  -0.101724761329
Avg Stopped Cars difference with bad lanes switching:  -8.26099771112
Num Breaks per Step difference with bad lanes switching:  0.144
Traffic Flow difference with bad lanes switching:  -0.0207578947368
```

Could this model fit to traffic in Hyderabad, India?

Unfortunately, I don't think this model is very applicable to traffic in India. Traffic in India works not only on an entirely different set of rules from this model, but also different from the rest of the western developed world. Traffic in India, firstly, does not really consider lanes. Even if they are drawn on the road, vehicles do not adhere to them. Second, cars do not slow-down in order to keep appropriate distance if they see a car in front. If they can, they will bypass the car while honking very quickly even going into the lane with the opposite side of traffic. Cars seem to never look back into the other lane before switching lanes. There are many extremely different kinds of vehicles and objects on the road which all have extremely varying speeds (slow tuktuks, speedy motorbikes and standing cows), paths of movement and behavior (like tuktuks driving only on the edge of the road, very slowly, and stopping very often), directions of movement (like pedestrians or cars crossing the road at any place). Moreover –vehicles (especially motorbikes and tuktuks) drive *against the direction of traffic* many times.

Generally, I would define Hyderabad's driving model extremely differently. It would look more like free-flowing movement of agents in an open, but narrowly defined space. Looked upon from above, it would probably look more like flow of water, or particles, or a swarm, through a vessel. The rules governing this behavior seem to be much more agent based as a complex system rather than an orderly cellular automata model. The rules in a model might be actually very similar to those of movements of swarms or animal groups. An abstraction of the rules that seem to govern the real-world traffic situation here, as I discussed upon with several local residents, might translsate to this *specific Hyderabad driving model update rules*:

1. Maintain at least ~1-meter space away from each other vehicle around you on any side
2. Drive as fast as you can until ~1 meter from the vehicle in front of you
3. If you are stuck behind a vehicle, shift sideways 1 car width (while honking) to where it's clearer in front, unless it is completely blocked by larger vehicles than you
4. When you hear a horn, look around, calculate the movements of other vehicles, and shift sideways 1 car-width to allow them to pass their route if they are larger than you
5. If there are obstacles on the way, try to bypass them from the side while keeping 1 meter distance from any other vehicle or slow down gradually to enable them to pass and you to bypass them

Such a set of rules (or a simpler classic swarm/flock behavior model) might produce an emergent behavior of collective movement of the system, in a free-flowing space, potentially mimicking better the driving behavior of the Hyderabad system.

Future Work

These models were not the most realistic, and there are many ways to make them more so. First, let's speak about the car and driver behavior. Feasible suggestions include storing unique features per car: variable maximum and average speeds per car, probabilities of slowing down, and probabilities of switching. This will help accounting for the various types of vehicles and drivers in the real world, and is technically not too complicated to do using an extra dimension to the scipy /

numpy array that could hold the number of features to include uniquely per car such as mentioned above, then accessing them of that car instead of the general class parameter.

Regarding the road, this is a closed world model with a circular lane-only road. It would be more interesting and realistic to include turns, lights, intersections, and what happens in them. One cold also include bad parts of the road itself which make cars go slower when driving over them. Finally, we should consider road blocks. Especially speaking of India, we should consider coding standing cows in the middle of the road as holy roadblocks. These could be also implemented b another dimension for the CA, indicating the feature of the road cell itself.

Regarding drivers and driver behavior, I would be interested to encode varying behaviors per driver, not uniform as we have now, and make them more realistic with different driving styles, different default distance kept, slowing down or accelerating at varying amounts (not just +/- 1), and exhibiting bad behavior such as not looking back when switching lanes ("cutting").

Finally, it would be interesting to encode ideal machine type drivers, who could perfectly predict the behavior of other cars on the road, possibly faux-communicate with them to act simultaneously to exit a traffic jam, in order to model and see how could we model, program, and expect self-driving cars to behave as a system and perform on the roads. Such a research may yield timely and useful conclusions.

References

Nagel, K., Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. Journal de Physique I, 2(12), 2221–2229.

Rickert, M., et al. (1996). Two Lane Traffic Simulations using Cellular Automata. Physica A: Statistical Mechanics and its Applications, 231(4), 534–550.

Scheffler, C. (2018). Code given in Class. Minerva Schools at KGI.