
An Analysis of the Impact of Self Driving Cars on Traffic Conditions

Written by

Vinit Ranjan
Junmo Ryang
Kelly Zhang

Duke University, United States of America

Sponsor: David Kraines, Duke University, Department of Mathematics, dkrain@math.duke.edu

Abstract:

The goal of this paper was to assess the effect of self driving cars on traffic conditions in the Greater Seattle Area using data from the Consortium for Mathematics and Its Applications in the 2017 Mathematical Contest in Modeling. We built a computational, agent based, mathematical model by which we could vary parameters such as number of lanes and capacity of cars. After polling a sufficient sample space, we ran our model and used curve fitting techniques to create functions to model the system. We used the model to calculate average speeds for various highways in Seattle. After creating our model, we adapted the computational model to allow for self driving cars to show increase in average car speed in various conditions. Our results show the increase in average speed with an increasing percentage of self driving cars in terms of increased average speed. The advantages of our model are the agent based aspects, which allow us to observe and model the system's behavior. Using this data to interpolate surfaces allows for more analytic techniques as well. The computational model is also flexible enough to poll data for different traffic conditions in other cities.

Authors' Note:

This paper was initially written for the Consortium for Mathematics and Its Applications' (COMAP) 2017 Mathematical Contest in Modeling and received recognition as a Finalist paper - top 10 out of 1527 teams working on the same problem. For reference, this problem was problem C and our team's control number was 70320. Permission was obtained to publish this paper provided that proper credit is given to COMAP. As such, the entirety of the problem and all preliminary data is property of COMAP.

E-mail contact: vinit.ranjan@duke.edu, junmo.ryang@duke.edu, k.zhang@duke.edu

Introduction and Background Information

As stated in the authors' note, all data and motivation for this problem comes from the Consortium for Mathematics and Its Applications in the 2017 Mathematical Contest in Modeling Problem C [2].

Self driving cars have emerged as a solution to the long standing problem of limited traffic capacity. In a study by Columbia University, it was found that cars only take up 5% of total available road space, and the implementation of cooperating self driving cars could increase road capacity by up to 273%. This is due to the large distances that drivers follow other cars, which is around 150 feet on average [1]. With the implementation of self driving cars (SDCs), we can not only effectively lower this following distance to approximately 20 feet but also reduce traffic congestion and prevent accidents that also contribute to low traffic capacity [5].

These self driving cars have various features that allow them to increase traffic capacity. Being able to travel at a constant speed would help prevent traffic "shockwaves" that occur when a leading car slows down and causes a ripple effect causing cars further back in the lane to slow down. This phenomenon has been shown previously in studies on circular tracks with the shockwave traveling at approximately of 12 miles per hour backwards [4]. By keeping a constant speed and being able to communicate with other cars, these backups and shockwaves can potentially be eliminated.

There is also an issue of bottlenecks where the number of lanes on a highway decreases. This causes some cars to switch lanes, and overall slows the traffic speed. Self driving cars have accurate road maps and know where these bottlenecks occur allowing for them to switch into other lanes earlier. When the number of lanes increases, there are typically no traffic congestion issues.

Finally, intersections of interstate highways and other state highways can also cause traffic issues when large influxes of cars join or leave the highway system. Exits and entrance ramps do not pose as big of an issue.

Restatement of the Problem

Given such a broad and complex question in traffic flow, we set out to clearly define the goals of our study. We interpret the problem to charge us with the following three main goals:

1. To understand the current traffic situation in the Greater Seattle Area (0% self driving cars)
2. To develop a means to predict and assess future traffic situations given the adoption of self driving cars
 - Consider varying levels of SDC saturation in the car market (e.g. 10%, 50%, 90%)
 - Search for equilibrium or threshold points in the traffic behavior
3. To identify the optimum achievable traffic situation through the introduction of self driving cars

In developing the specific steps above to answer the problem, we often made assumptions and decisions on complicated phenomena to make the problem tractable. The following section outlines these assumptions and decisions.

- **Assumption 1: Both sides of a highway contribute equally to Average Daily Traffic Volume.**

Because the given Seattle traffic data only provides the conglomerate count of all cars on both sides of a segment of highway during a 24 hour period, we must find a way to partition this count between the two sides of a road.

- **Assumption 2: Drivers that travel on one side of the road on a particular day will also travel the other side of the road on the same day.**

Most traffic, especially in urban centers such as Seattle, consists of employees commuting to and from work. Therefore, it is feasible that any given car traveling on a road will have to take the same road in the other direction for their commute. This supports the previous assumption.

- **Assumption 3: Traffic on one side of the road does not affect the other side of the road.**

There are very few unlikely reasons why traffic on opposite sides of the highway would interact (e.g. large accidents, construction zones, etc), none of which we can feasibly model (see assumption 4). This allows us to model both sides of a road by focusing on the dynamics within one side.

- **Assumption 4: Accidents are erratic by nature, so we cannot effectively model their occurrence.**

While we might be able to model the effects of such events such as accidents, the scope of our model does not extend to accurately predicting their cause or occurrence.

- **Assumption 5: The amount of cars per day can be uniformly distributed throughout the day.**

The purpose of this simplifying assumption is twofold:

1. We were not provided sufficient data such that a proper distribution could be derived, and we could not locate additional sources of data specific to Seattle that could aid us in deriving a feasible distribution.
2. Assuming a steady constant value improves the simplicity of our model and the underlying computations when compared to the alternative of using a random variable sampling from a distribution.

Using the mean number of cars across the entire day as the constant value is as reasonable as any other value. If ever the occasion arises that a maximum value, mode value, or some other constant value becomes available to use, the analysis can simply be repeated with the simple replacement of the value.

- **Assumption 6: Non self driving cars go up to 10 miles per hour over the speed limit.**

Realistically, most drivers seldom strictly keep the speed limit. Additionally, drivers commonly speed up while switching lanes, feasibly traveling up to 10 miles over the limit. In this problem, the nominal speed limit was given as 60 mph, so that is the value used throughout this paper.

- **Assumption 7: Non self driving cars grow "impatient" and attempt to change lanes when slowed by cars in front of them, while self driving cars do not.**

Humans are subject to impatience, especially when driving slower than their preferred speed. They will seek ways to relieve their impatience, such as by changing lanes (or honking). SDCs have a pre-calculated route and try not to excessively change lanes.

- **Assumption 8: Non self driving cars are concerned about keeping safe following distances proportional to speed.**

Humans have non-negligible reaction times that inflate their stopping distances proportionally to their speed. This forces them to keep longer following distances for higher speeds.

- **Assumption 9: Self driving cars strictly follow the speed limit.**

SDCs are pre-programmed by manufacturers, and manufacturers will be discouraged by legislators to produce, advertise, and sell an illegal product. Since traveling over the speed limit is technically illegal, SDCs will be forced to follow the speed limit when cruising (switching lanes allows for a momentary increase).

- **Assumption 10: Non self driving cars (human manned) use Gipp's car following model.**

Cars will speed up if there are no cars within it's following distance and will slow down when there is a car in it's following distance. This also ties into the idea of patience and has been seen in human car driving studies [6].

- **Assumption 11: Self driving cars do not have to keep as long following distances.**

SDCs have fairly negligible reaction times due to their high-tech sensory and communications devices. They will begin braking near instantaneously as the car in front starts to brake. As a result, their following distances can afford to be much shorter with the minimum being 20 feet [1].

Please note that while only 3 digits are displayed after the decimal in our values, all significant figures were kept in calculations and in the data.

Part I: The Standard Traffic Model

We attempt to address the first goal of understanding Seattle's current traffic situation through the following steps:

1. Establish average (arithmetic mean) speed as the metric by which we judge any given traffic situation along a segment of highway, hereafter denoted as z .
2. Develop an agent-based model in Netlogo that models cars driving along one side of a highway with variable car density (x_1), or cars per second, and variable number of lanes (x_2); we can call this the Standard Traffic Model.
3. Poll the metric z (average speed) for regular intervals along the x_1 (car density) and x_2 (number of lanes) axes, ensuring that the full range of Seattle traffic data lies within the rectangular region sampled.
4. Fit a surface to the set of (x_1, x_2, z) points generated by the polling. The function $z = f(x_1, x_2)$ describing this surface closely approximates the Standard Traffic Model for the rectangular region in $x_1 \times x_2$ space encapsulating the Seattle traffic data; we can call this the Seattle Function f_0 .
5. Interpret what the Seattle function says about the traffic and an overall metric, Z , for the map of Seattle.

For the rest of this paper, we shall refer to many of the specific parameters and values with the variables introduced above. To reiterate: z represents the mean speed of cars; x_1 represents the car density or cars per second; x_2 represents the number of lanes. Z is a number used to represent the average speed of cars across all of Seattle. Its eventual derivation is the main bulk of Part I, and its extensions are discussed in Parts II and III.

Data Consideration

The given data contains total counts of cars that cover any given stretch of a road per day. The stretches of road are delimited by mile markers. We are also given information such as the type of road and number of lanes. In order to run our model to obtain meaningful results for the data, we observe that the number of lanes for every scenario is between 2 and 5, inclusive. Upon further inspection, we also observe that the number of cars on each road is given as a total per day. Via our assumption 5, we treat the number of cars at any time as essentially constant. So, we define our 'car density' (x_1) as the number of cars that pass through any arbitrary line across all of the lanes per second. We are given the total number of cars per day, which we will call C . Assumptions 1 and 2 let us separate the two sides of the road and also allow the two sides of the road to contribute to the car density equally. So, we divide the initial value C by 2 to account for the two sides of the road. We then divide by the number of hours per day (24) and the number of seconds per hour (60*60). The resultant value gives car density in cars/sec. This is summarized by equation (1).

$$x_1 = \frac{C}{2 * (24 * 60 * 60)} \tag{1}$$

Using the excel sheet given, we calculate car density for each road segment given and conclude that the maximum value for car density is approximately 1.4 cars/sec. This gives us a range of values to aim to sample over. Note that in the original development of our model, we do not actually use the data. We create the model independently of the data then apply the data to the resultant equations. This preliminary calculation simply gives us an idea of the sample polling space to use for the final application of the model.

Development

Before starting development of the model, we need to establish a metric by which we can measure the quality of a given traffic scenario. In order to judge the highway in each of its segments, we decide to define a metric z as the arithmetic mean of the speeds of all cars after an equilibrium has been reached. This is a fair metric due to the fact that the time to travel across each specific segment is longer than the time for the Netlogo system to reach equilibrium. To judge the system as a whole, we use the weighted arithmetic average of these z speeds based on the distance of that respective segment for the combined northbound and eastbound highway sides (increasing mile markers) and for the combined southbound and westbound highway sides (decreasing mile markers). We choose this metric because it gives an overall sense as to how the entire system of cars is moving. We denote this weighted average Z .

Our Standard Traffic Model is developed in Netlogo [8], an agent-based modeling software designed for studying complex systems. Our current problem of traffic flow strongly lends itself to this sort of model, as each car can be modeled as an independent agent with distinct parameters, taking in information from its surroundings and neighbors and making decisions based on pre-defined criteria. Netlogo gives us the freedom to place these any number of these agents in a simulation with user-defined initial conditions. In Figure 1, we present the logic flowchart for a non-SDC agent within our Standard Traffic Model and a view of the Netlogo interface for our model. The non-SDC agents are represented as blue cars, whereas the SDC agents are represented as orange cars.

For the base model, we start with Uri Wilensky's 'Traffic 2-Lane' model and modify the code [7]. Modifications include a change to the creation logic in order to allow multiple numbers of lanes. In addition, we add in SDC agents, which follow a slightly different logic algorithm. The different logic for these agents will be discussed in Part II.

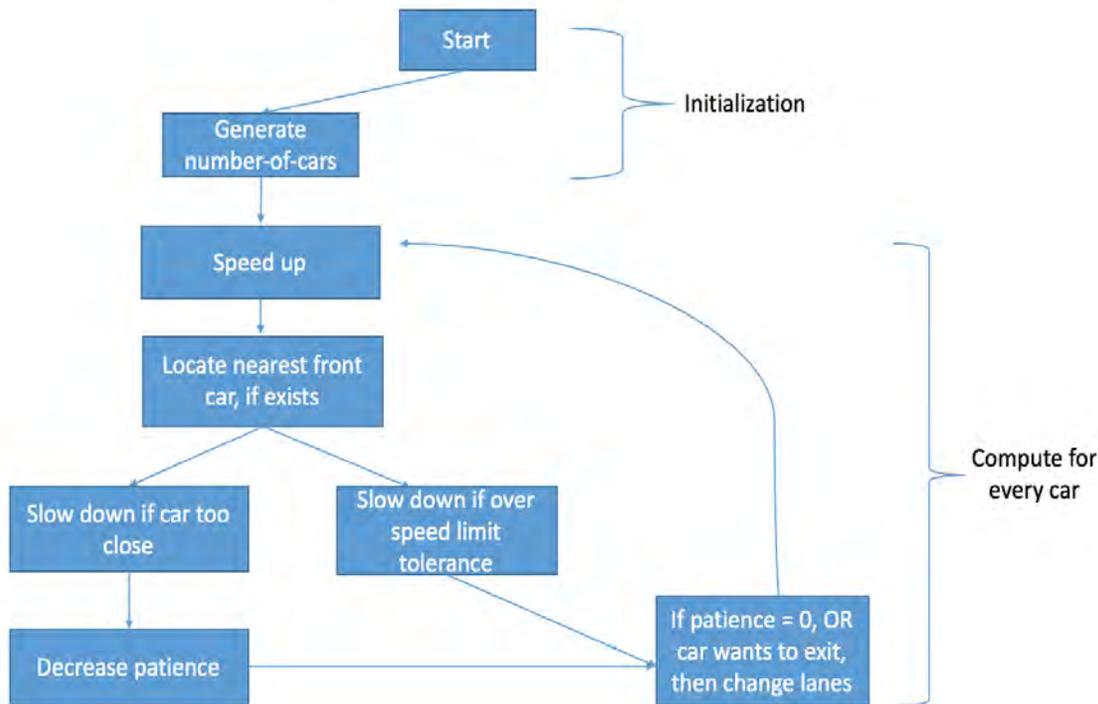


Figure 1: Non-SDC Agent Logic

As stated in the assumptions, the nominal speed limit for the entire problem was given as 60 mph, and each individual agent has its own randomly generated tolerance value. This accounts for variation in driver

tendencies to drive over the speed limit. In addition, each car has a randomly generated patience value, which is on a scale from 1 to 100. The change lane procedure resets the driver's patience. Every tick, with a very small probability, (.002), a car makes a decision to change to the bottom lane - a method by which we chose to model the desire of a car to change lanes because it needs to exit soon. This probability was simply chosen to impart this lane change characteristic upon the system. Lastly, a leading car is determined to be 'too close' if the car is within 150 feet (about 2 patches) at 60 mph because this is a close the recommended following distance for cars [1]. Although, this following distance decreases in direct proportion with the car's speed since going slower requires a smaller following distance.

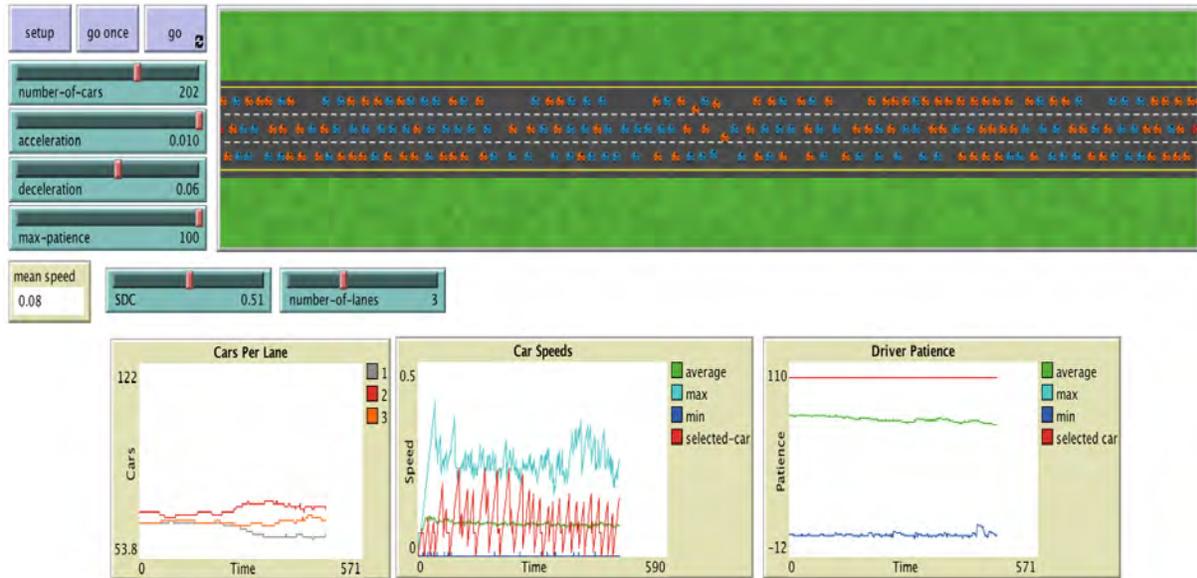


Figure 2: NetLogo Interface

As shown in the view of the interface above, in addition to agents representing non-SDCs, our Standard Traffic Model takes car density and number of lanes parameters. The output is given as a graph of min, max, and avg speeds. By allowing each simulation to run until the average speed reaches an equilibrium, we can map each pair of car density and number of lanes parameters to a single average speed value. Note that the speed limit in the program is set such that 1 patch per iteration in NetLogo represents 60 mph.

Results

By running simulations in NetLogo using given ranges and intervals for each parameter, we obtained data for the average speeds z in an Excel spreadsheet. To decide the parameters, we looked at the data and determined the ranges to iterate over; we ran the simulations with the following ranges: number of lanes from 2 to 5, and car density from 0.05 to 1.5. These ranges was chosen from the data, as discussed in the 'Data Consideration' section. Figure 3 gives the resultant 3D scatter plot from MATLAB to help discover any trends in the wide set of average speeds for highway segments.

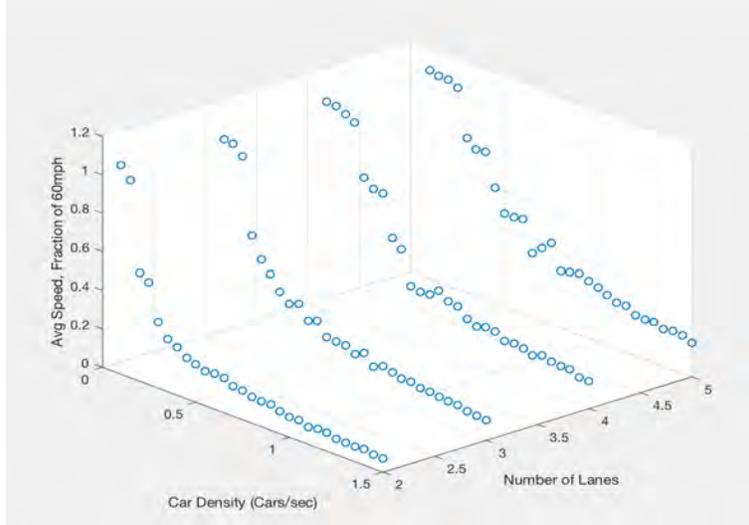


Figure 3: Average Speed vs Car Density vs Lanes

Note that the data appears to have two phases since it levels out for very low density values. This is simply because at lower speeds, average speed is capped by the speed limit instead of traffic patterns. In order to fit this data with a surface, we first tried to find the relationship between average speed (z) and car density (x_1). We took a section of the data with number of lanes set constant at 2, and attempted to best fit the data by linearizing. Eventually, we determined that a plot of $\ln x_1$ vs $\ln z$ produced the most linear fit, leading us to believe that a power law fit was correct, of the form $z = \alpha x_1^\beta$.

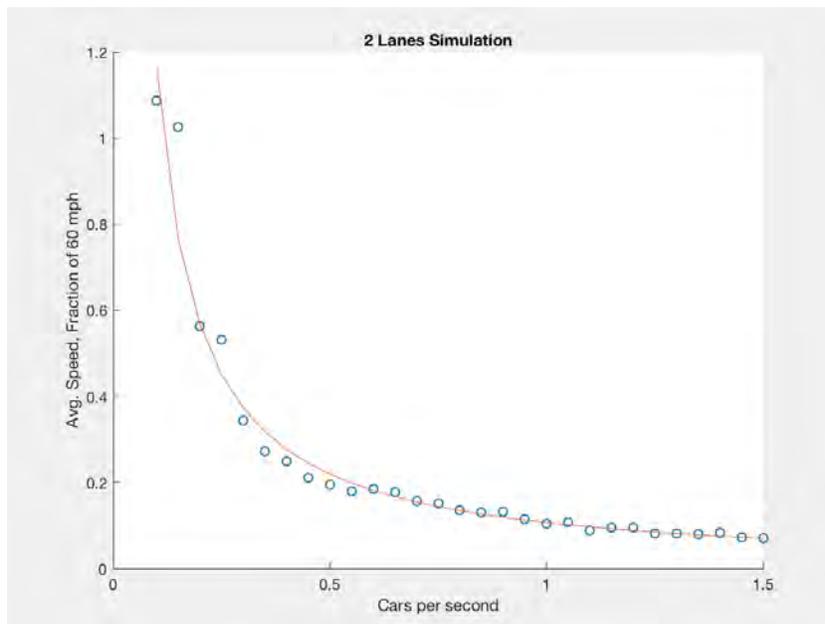


Figure 4: Car Density vs Average Speed, Lanes = 2

The power law fit yielded by fitting to the curve:

$$z = 0.107(x_1^{-1.037})$$

To show support for our fit choice, we produced the residuals for this plot, and the fairly random distribution of the residuals support our choice strongly. The spreading to the left hand side of the graph can be explained by the erratic behavior when the car density is low (people driving at night with low visibility, etc).

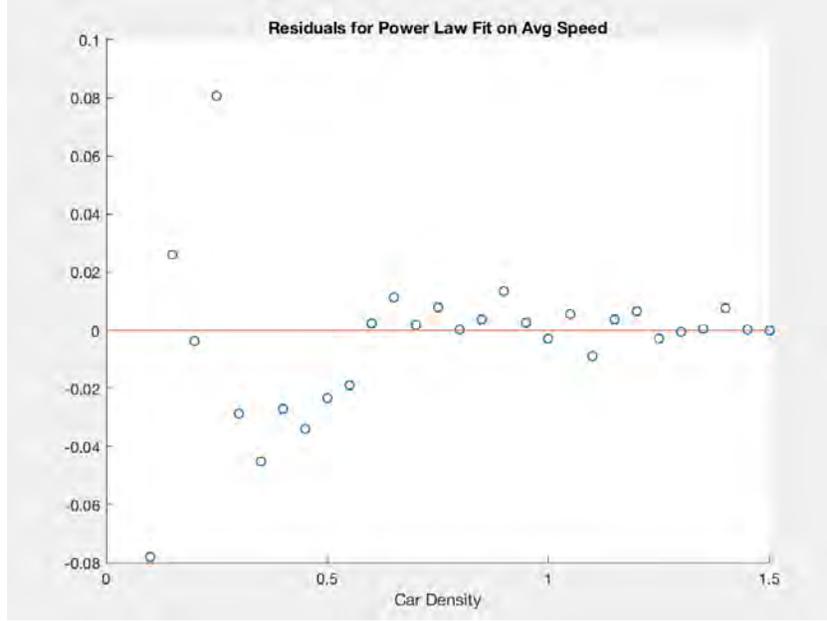


Figure 5: Residuals: Car Density vs Average Speed, Lanes = 2

Doing the same for all of the curves corresponding to lanes = 3, 4, 5, we arrive at 4 strong power law fits with variable α and β constants. Table 1 gives the constants for the power laws of the form $z = \alpha x_1^\beta$.

Number of Lanes	α	β
2	.107	-1.037
3	.166	-.930
4	.215	-.914
5	.274	-.803

Table 1: Constants Alpha and Beta for Different Numbers of Lanes

From here we attempted to find the relationship between these constants and the number of lanes by plotting alpha versus the number of lanes (Fig. 6) and beta versus the number of lanes (Fig. 7). Both α and β were found to have a fairly linear relationship to the number of lanes. So, replacing α and β by general linear equations (of the form $y = mx + b$), we can get the form of the multivariable equation for avg. speed, z .

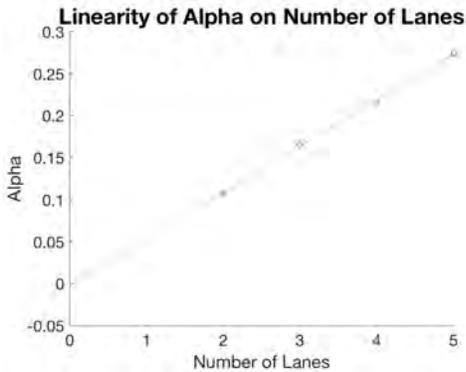


Figure 6: Alpha vs Number of Lanes

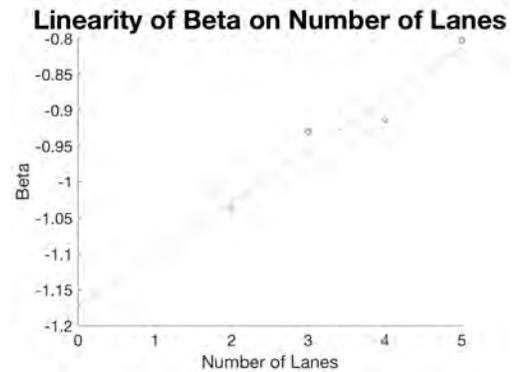


Figure 7: Beta vs Number of Lanes

To reiterate, we began by searching for a relationship just between z and x_1 , and found strong evidence for a power law relationship of the form $z = \alpha x_1^\beta$. To introduce the x_2 into the equation, we studied how α and β

varied for different values of x_2 , and found strong evidence for a linear relationships of the form $\alpha = ax_2 + b$ and $\beta = cx_2 + d$. Substituting, we arrive at equation (2).

$$z = \alpha x^\beta = (ax_2 + b)x_1^{cx_2+d} \quad (2)$$

Using MATLAB's curve fitting tool, we fit a surface of this form to the data, and the result is shown in Figure 8.

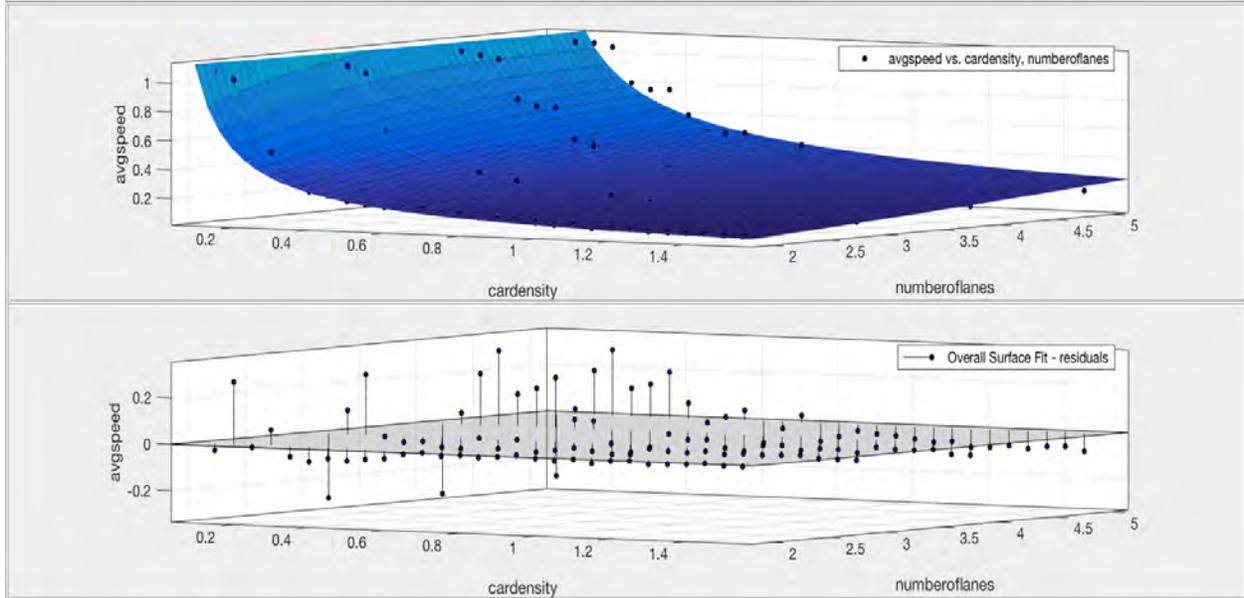


Figure 8: Surface Fit: Seattle Function f_0 with Residuals

The equation of the surface in Figure 8 is given by the next equation.

$$f(x_1, x_2) = (.068x_2 - .013)x_1^{.114x_2-1.188}$$

The residuals show that the function accurately models most of the space sampled, but has a tendency to underestimate the average speeds in low x_1 and high x_2 situations.

Note that the values for z seem to be on a scale of approximately 0 to 1, but this result was simply from the fact that the average speeds were measured in patches per iteration in Netlogo. Since we set up the model for this unit of speed to represent 60 mph, then we allow these z values to represent fractions of 60 mph. As a result, we write the overall Standard Traffic Model Seattle function as:

$$f_0(x_1, x_2) = 60(.068x_2 - .013)x_1^{.114x_2-1.188}$$

There are many advantages to creating the Seattle Function f_0 . While our NetLogo model is powerful, it is not very portable. When seeking the average speed across a stretch of highway with a specific car density and number of lanes, we must set up the model with these specific configurations and run a full simulation. In comparison, the f_0 model has enormous utility because, in the same situation, all we must do to find the average speed is evaluate $f_0(x_1, x_2)$, where x_1 and x_2 are the specific car density and number of lanes, respectively.

Now using the average of speeds (z from the f_0 function), we can calculate the weighted average of speeds (Z) on the highways as a whole based on distance. For the north and eastbound highways, $Z = 16.262$ mph. For the south and westbound highways, $Z = 16.300$ mph. These do not vary much from each other and can be combined to form $Z_0 = 16.281$ mph (averaged because they have the same car flow based on assumption 1) for when there are 0% self driving cars.

Interpretation

Figure 9 features the results of our NetLogo model in a visual form. The parameters for each road segment in the Seattle data were fed into our Seattle Function. The colors indicate the average speed of each segment. Potential bottlenecks can occur for any contiguous stretch of road that decreases in average speed. Each side of the road is color coded based on the speed of that direction of traffic flow (on the right side of the road is the north or eastbound traffic flow and on the left side of the roads is the south or westbound traffic flow). The base map was provided by COMAP as part of the problem itself [2].

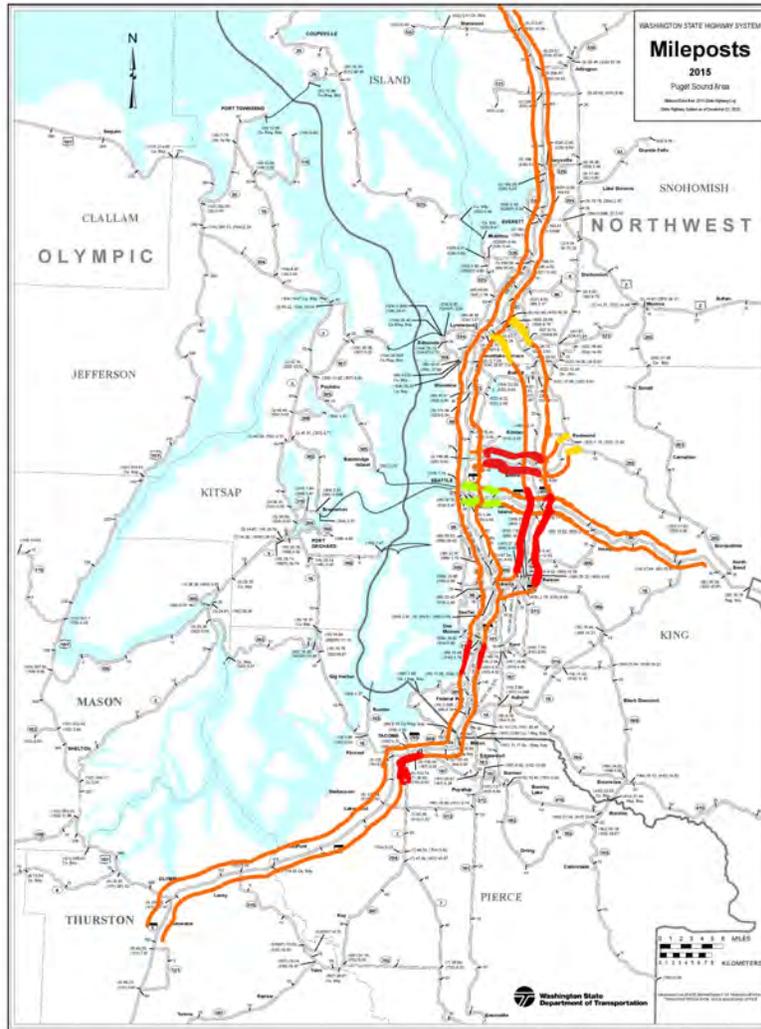


Figure 9: Color Map For Current Seattle Conditions, $Z_0 = 16.28$ mph

Color	Avg. Speed Range
Red	0-10 mph
Orange	10-30 mph
Yellow	30-50 mph
Green	50-60 mph

Table 2: Legend for Color Coding

Bottlenecks

As lanes end on highways, cars will need to switch lanes to continue driving, potentially causing cars to change speeds. On top of this, when the car density is fairly large, the impact of these lane drops is magnified. Lower speeds allow for smaller following distances which also can prevent some cars from having opportunities to switch lanes. This is mainly an issue with non self driving cars. Unfortunately, this is also erratic by nature and hard to model due to the large number of extra parameters needed. Thus, the equilibrium speeds will just be compared between sections with different numbers of lanes, knowing that the worst case scenario is much worse and could have traffic potentially stopped. This cannot be helped unless infrastructure changes occur (adding lanes/changing the numbers of lanes) or policy changes are made (designating lanes for specific cars).

On the other hand, with a larger percentage of self driving cars implemented with smart road mapping technologies, this effect can be mitigated with early detection and switching.

Intersections

Similar to bottlenecks, these large influxes of cars can cause issues, but our model is not built to deal with these as they are also quite unpredictable. The time between when the average speeds have not yet reached equilibrium are hard to measure and act similarly to bottlenecks.

Part II: The Self Driving Car Traffic Model

We attempted to answer the second goal of predicting the impact of self driving cars with the following steps:

1. Establish a specific list of behavioral differences between SDCs and non-SDCs based on the features and limitations of modern SDCs.
2. Adapt the Standard Traffic model to implement SDCs and inspect for any qualitative differences in behavior. Allow for variations in the proportion of SDCs out of the entire car population; we can label these the SDC-10 Traffic Model, SDC-50 Traffic Model, and so on.
3. For certain SDC traffic models, poll the metric for the same rectangular region in $x_1 - x_2$ space, and again create Seattle Functions: f_{10} , f_{50} , f_{90} and so on.
4. Apply various analytical tools to the Seattle Functions in order to identify the overall effect of introducing SDCs to Seattle traffic.
5. Interpret what the multiple Seattle Functions say about traffic speed overall (Z_x) on the Greater Seattle area.

Development

To extend our Standard Traffic Model to an SDC Traffic Model that incorporates self driving cars, we had to introduce a new type of agents. Keeping in mind our assumptions about SDC's, we added the slider to the NetLogo interface shown in Figure 2 called 'SDC', intended to represent the percentage of SDC's. Within the simulation, we represented SDCs with orange cars to distinguish them from the blue non-SDCs. Using the definition from COMAP concerning self-driving, cooperating cars, we have the following:

- Self-driving cars do not have drivers; rather, they are guided by various sensors that allow for extended vision ahead and 360 degree coverage around the car [3].
- Cooperating cars can identify, communicate and exchange data with other cooperating cars, assisting in route planning and decision making.

With these definitions, combined with other assumptions we made, the logic for the SDC agents was subject to the following changes:

- SDC's have a hard set speed tolerance at 60 mph, the speed limit.

- If nearest front car is also an SDC, then decrease following distance significantly [1].
- Never decrease the patience counter, as SDC's are not subject to this issue.

Results

Within the confines of the problem statement, we were specifically tasked with determining the effects of 10%, 50%, and 90 % SDC's, called the SDC-10 Traffic Model, SDC-50 Traffic Model, and SDC-90 Traffic Model, respectively. We chose to also add in the scenario where we have 100% SDC's, called the SDC-100 model. We repeated our NetLogo simulations over the sample space to find our f_{10} , f_{50} , f_{90} , and f_{100} functions. However, as the percentage of SDC's, we noticed a new behavior as a result of the hard speed limit set on the agents. The data for the SDC-50 Traffic Model is given in Figure 10.

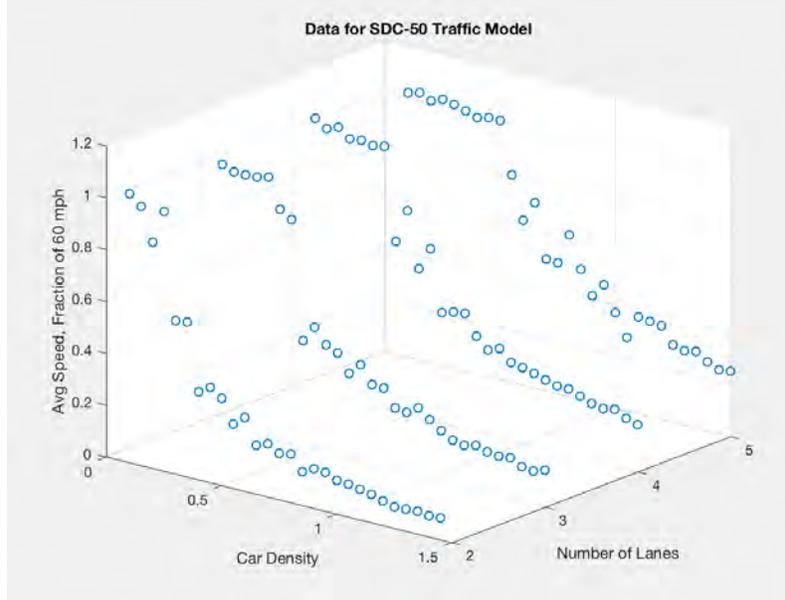


Figure 10: Average Speed vs Car Density vs Number of Lanes, SDC = 50%

The effect of the speed limit is especially pronounced in the data subset that has number of lanes = 5. As discussed earlier, for smaller car densities, the average speeds tend to be cut off slightly above 60 mph due to the speed limit. So, this part of the data is not simply able to be approximated via the same surface as the rest of the data, which still tends to follow the power law distributions. Here, a decision needed to be made as to how to modify our functions because the effect of the speed limit cap was too pronounced. In order to simplify the model, yet still accurately represent the phenomena at hand, we made two decisions. First, the data were looked at, and those values that produced z values at or above 1 (in terms of patches in the Netlogo model) were removed from the set. With the remaining data, the MATLAB curve fitting tool was used to find a surface of the same form as f_0 . Lastly, the new functions were set as the minimum of 60 and 60 times the new function. We know that this will work since the surface will tend to overestimate for these small values of car density for what the maximum average speed can be, but the minimum function will bring these values to the speed limit. The results are as follows for the SDC-10, SDC-50, SDC-90, and SDC-100 models, respectively.

$$\begin{aligned}
 f_{10}(x_1, x_2) &= \min(60, 60(.065x_2 - .002)x_1^{.071x_2 - 1.054}) \\
 f_{50}(x_1, x_2) &= \min(60, 60(.090x_2 - .062)x_1^{.089x_2 - 1.621}) \\
 f_{90}(x_1, x_2) &= \min(60, 60(.169x_2 - .172)x_1^{-.027x_2 - 1.253}) \\
 f_{100}(x_1, x_2) &= \min(60, 60(.195x_2 - .211)x_1^{.156x_2 - 1.914})
 \end{aligned}$$

The fits are given in Figures 11 through 14. Note that they are only plotted on top of the data that were in the valid range of the distribution.

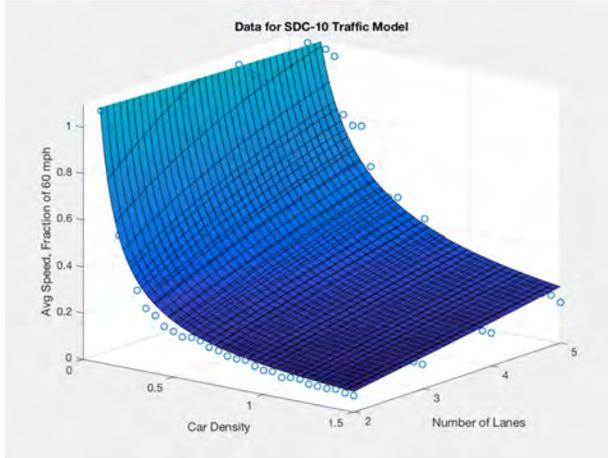


Figure 11: SDC = 10%, Surface Fit

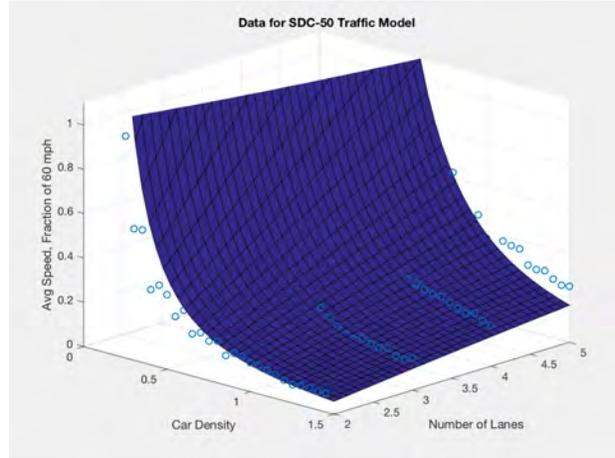


Figure 12: SDC = 50%, Surface Fit

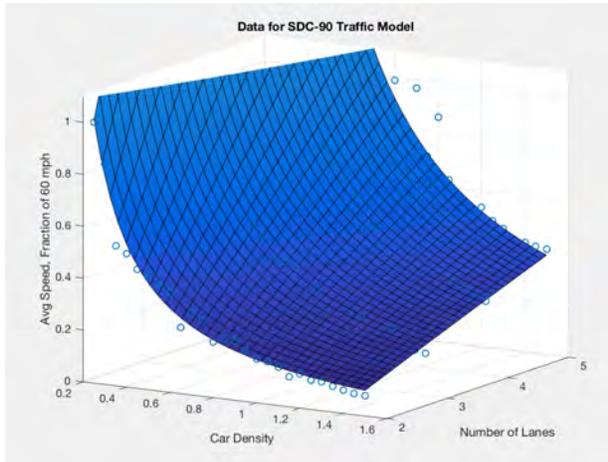


Figure 13: SDC = 90%, Surface Fit

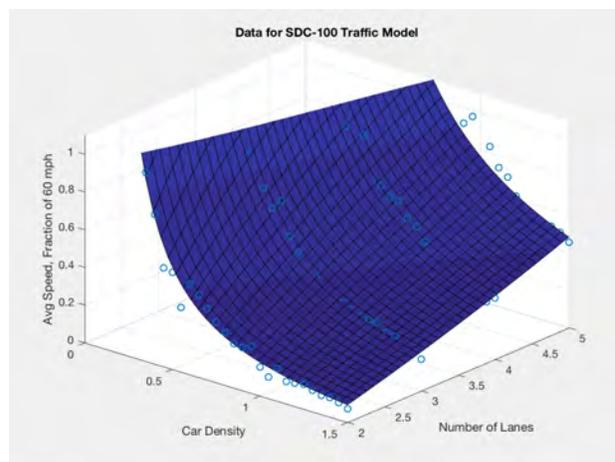


Figure 14: SDC = 100%, Surface Fit

We calculated the Z values (weighted average speed) for the overall highway sides and accumulated results in Table 3.

Z	10% SDC	50% SDC	90% SDC
North and Eastbound	16.463 mph	22.398 mph	34.967 mph
South and Westbound	16.496 mph	22.393mph	34.894 mph
Average (Z_x)	16.479 mph	22.396 mph	34.931 mph

Table 3: Weighted Average Speed for Different SDC%

Interpretation

Similarly to the results shown in part I, we present the results of our NetLogo model in a visual form. Using the same colors and average speed values, we produced:

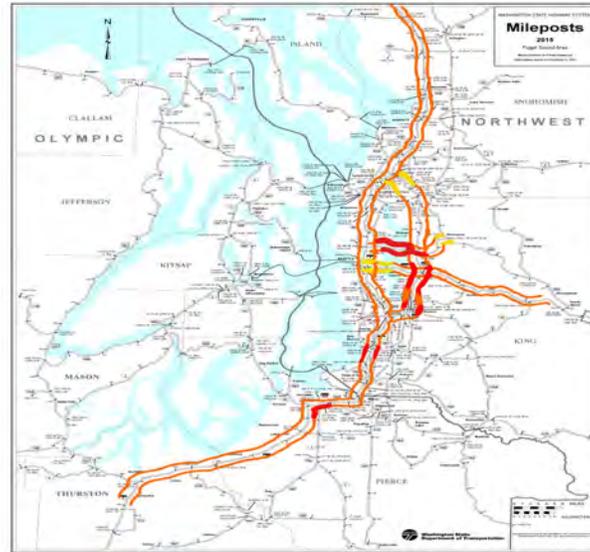


Figure 15: Projected Color Map for 10% SDC's

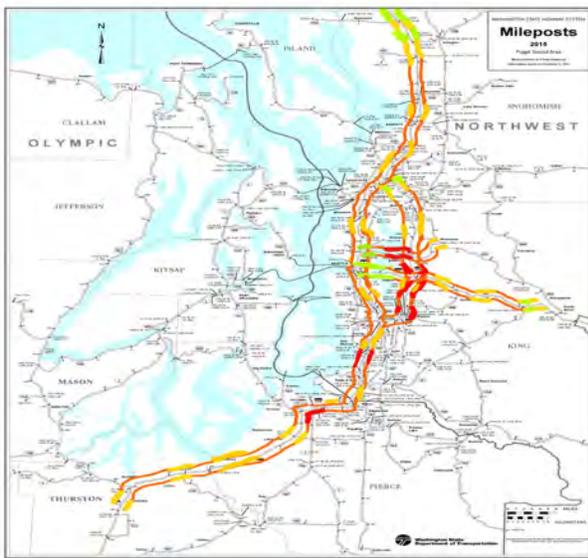


Figure 16: Projected Color Map for 50% SDC's

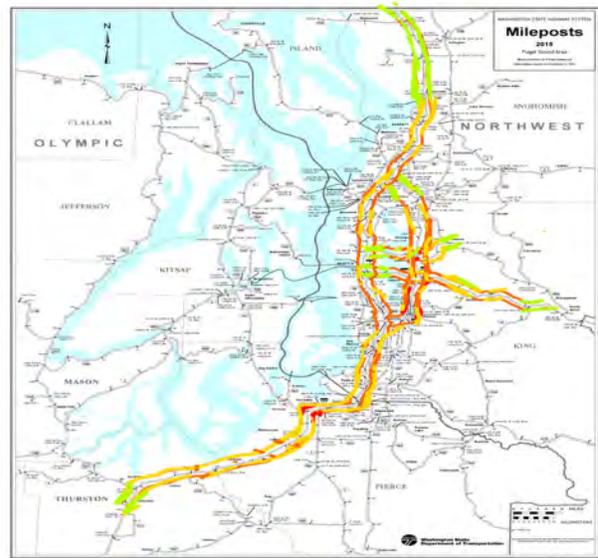


Figure 17: Projected Color Map for 90% SDC's

As the percentage of SDC increases, it is clear to see from the maps that the amount of red and orange has decreased and changed into the yellow and green that represent faster average speeds for those respective highway segments. Many of the green sections are towards the ends of the highway segments where traffic is less dense. On the other hand, the red sections tend to be at bottle necks and around the center of the map where Seattle itself is located. This suggested to us that many people travel to and from work (work in Seattle where the congestion is the heaviest and speeds are slower) as we stated in assumption 2.

As seen in Table 3 above the Z_{10} , Z_{50} , and Z_{90} values increase as the percent of self driving cars increases. These Z_x values with the one from part 1 (Z_0) and Z_{100} (which was calculated in addition to the rest of the data) allows us to try to find a fit for how the overall system of highways' speed changes as the percent of SDCs changes.

Part III: Optimizing Seattle Traffic

We attempted to fulfill the third goal of optimizing traffic flow by the following steps:

1. Approach the optimization through 3 distinct methods
 - (a) Consider the 3-dimensional surface graphs of $f_0, f_{10}, f_{50}, f_{90}$, etc. as level surfaces for an overall Seattle Function $f^*(x_1, x_2, x_3)$, where x_3 is the percentage of self driving cars.
 - (b) Visually study the surface graphs of the functions $z = g_{x_2}(x_1, x_3)$ for $x_2 = 2, 3, 4, 5$ that yield average speed, keeping number of lanes constant, and observe at which values of x_3 lie maxima.
 - (c) Plot the broader metric Z against x_3 , find a strong fit, and maximize this single variable function $Z = h(x_3)$.

Connecting Level Sets

If we define a function f^* such that $f^*(x_1, x_2, x_3) = f_{x_3 \cdot 100}(x_1, x_2)$, then all of the surfaces that we found in the previous section are level sets of the 3-variable Seattle Function. We can attempt to visualize the 4-dimensional graph of this function by viewing the overlay of the surfaces.

In Part I, we developed the form of the 2-variable f_0 function by plotting fit variables for its traces against the number of lanes. We attempted to find an equation in three variables for this equation by the same method:

We started with the form and fits for the level sets.

$$f_{x_3 \cdot 100}(x_1, x_2) = (ax_2 + b)x_1^{cx_2 + d}$$

x_3	a	b	c	d
0	.068	-.013	.114	-1.188
10	.065	-.002	.071	-1.054
50	.090	-.062	.090	-1.621
90	.169	-.172	-.027	-1.253
100	.195	-.211	.156	-1.914

Table 4: Fit Parameter Values for Different Level Sets

Next, we plot and fit the relationship between each of the fit parameters in the equations above and x_3 . However, we quickly run into a problem. The fits for a and b are strongly quadratic, but no definitive fits for c and d can be found. This is shown in Figure 18. We have some suspicions that the exponential β term in Part I is not actually linear against x_2 , but no stronger fit was found than the linear fit given earlier.

At this point, we decided to defer to our other two methods. Even if strong fits were found above, we would only be left with the form of f^* . In Part I, MATLAB was used to find the exact constants to fill out the form of the function, but the software is limited to 3-dimensional surface fits. Additionally, we would be unable to visually inspect the function as it is in three variables. The following two methods seek to reduce the number of variables to complement an analytical solution with a visual representation.

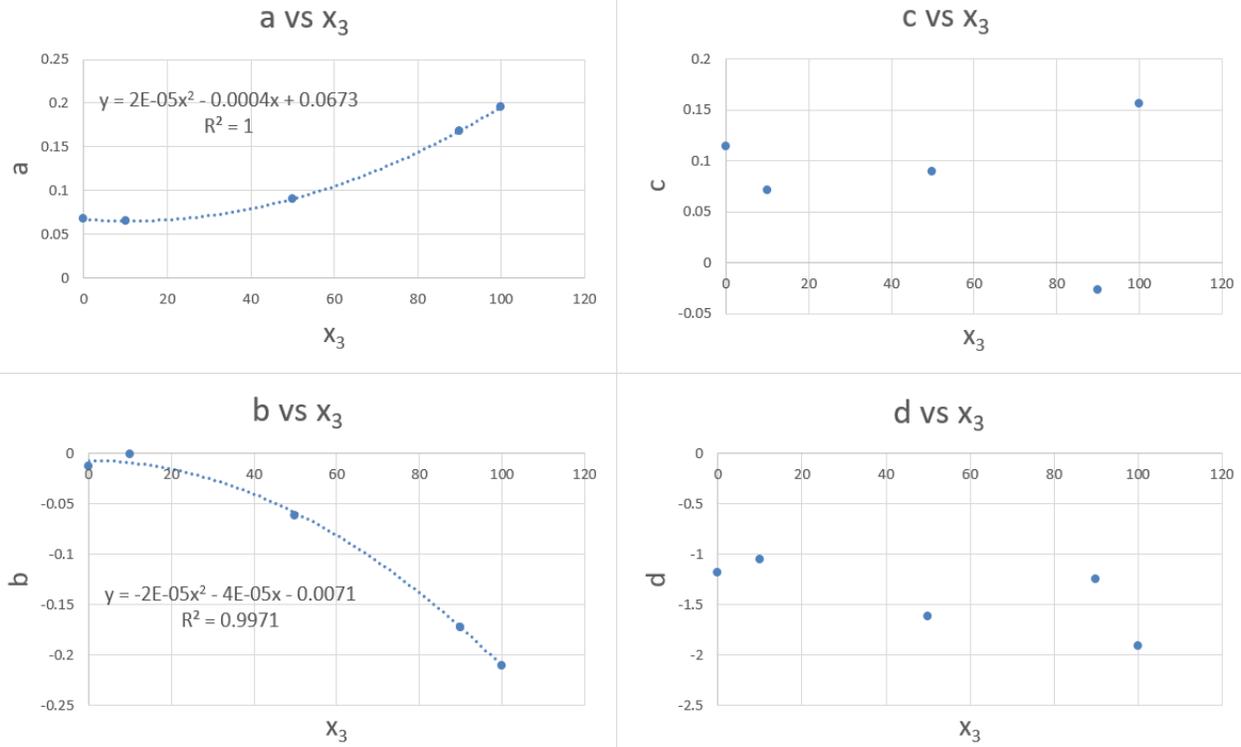


Figure 18: Fit Variables vs x_3

Viewing Alternate Level Sets

We currently have defined $z = f_{x_3}(x_1, x_2)$ as level sets of f^* . So we define alternate level sets $z = g_{x_2}(x_1, x_3)$. By rearranging our data, surfaces can be fit to these level sets, and equations for these functions found:

$$g_2(x_1, x_3) = (.086x_3 + .097)x_1^{-.336x_3-1.047}$$

$$g_3(x_1, x_3) = (.205x_3 + .149)x_1^{-.044x_3-.967}$$

$$g_4(x_1, x_3) = (.277x_3 + .185)x_1^{-.259x_3-1.019}$$

$$g_5(x_1, x_3) = (.361x_3 + .234)x_1^{-.152x_3-1.075}$$

From here, following the same path as above would only lead to the same result. However, a new path has opened up, as the functions for these level sets include the parameter x_3 , which we hope to optimize z against. In addition, these 2 variable functions can also be graphed in 3-space, giving us the option of visual confirmation.

We will begin with g_3 , considering that three lanes are the most common occurrence in Seattle. Since, we cannot change the car densities in Seattle and lanes are held constant, we are only optimizing with respect to the proportion of SDCs on the road. So, we find the partial derivative of g_3 with respect to x_3 in order to find the critical points. Then, we can compare the value of g_3 at the endpoints ($x_3 = 0, 1$) against

the values at the critical points to find the global maximum.

$$g_3(x_1, x_3) = (.205x_3 + .149)x_1^{-.044x_3-.967}$$

$$\frac{\partial}{\partial x_3}g_3(x_1, x_3) = x_1^{-.044x_3} [.205x_1^{-.967} - .044x_3 \ln x_3 - .007x_1^{-.967} \ln x_3]$$

$$\frac{\partial}{\partial x_3}g_3(x_1, c) = x_1^{-.044c} [.205x_1^{-.967} - .044c \ln c - .007x_1^{-.967} \ln c] = 0$$

$$x_1 \neq 0, \quad x_1^{-.044c} \neq 0$$

$$.205x_1^{-.967} - .044c \ln c - .007x_1^{-.967} \ln c = 0$$

$$.205x_1^{-.967} = \ln c [.044c - .007x_1^{-.967}]$$

We again run into a problem. Unfortunately, the partial derivative is very complex. The resultant form above can be represented $A = c \cdot e^{Bc+C}$, where A and C vary with x_1 . From this form, we cannot solve for c , except by introducing the Lambert-W function, which does not give us any meaningful interpretations.

Alternatively, we can take a simple, cursory glance at the graphs given in Figures 19 through 22 and note that the value of the function strictly increases as we move in the direction of increasing SDCs. In particular, at middling car densities, we see a greater change. This make sense intuitively, as SDCs would not make much of a difference in situations with too few or too many cars.

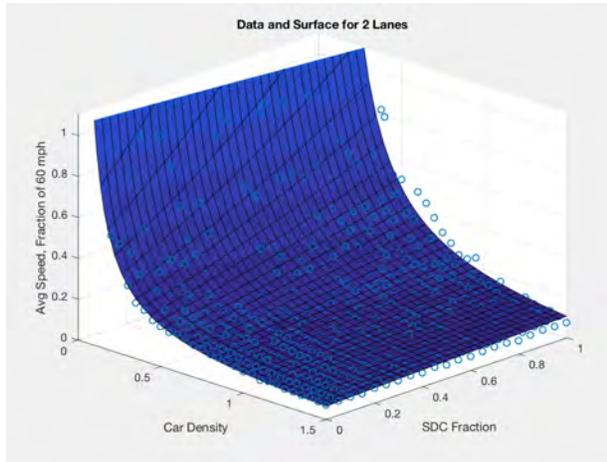


Figure 19: 2 Lanes, Surface Fit

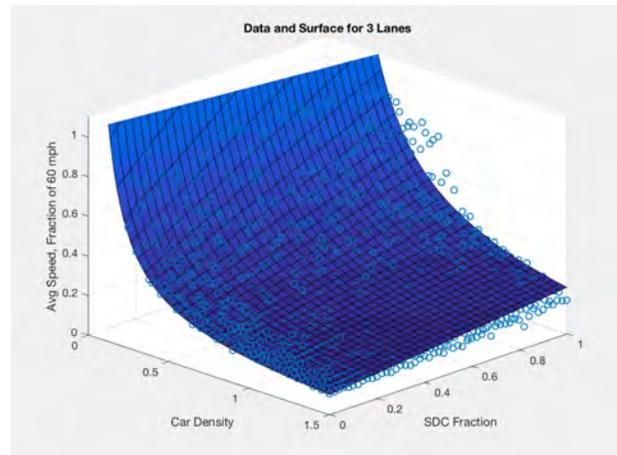


Figure 20: 3 Lanes, Surface Fit

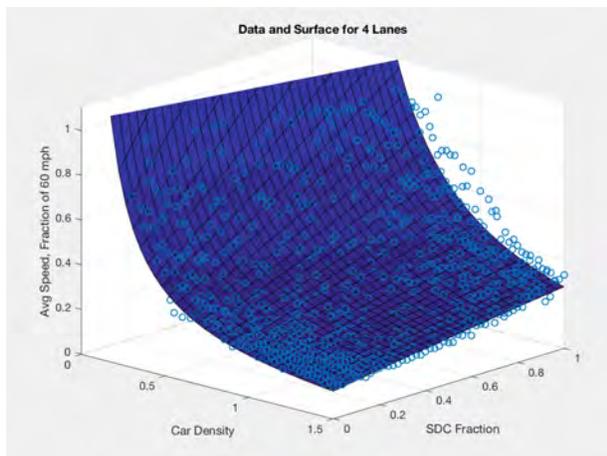


Figure 21: 4 Lanes, Surface Fit

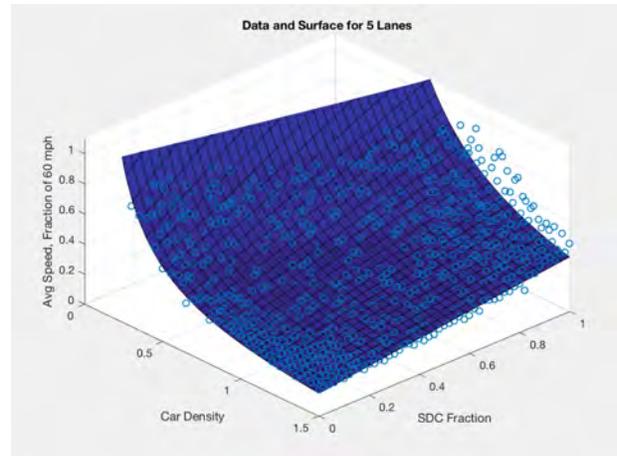


Figure 22: 5 Lanes, Surface Fit

So, while the previous method failed to provide both an analytical and graphical solution, the current method at least yielded a graphical one. The next method gives an even simpler method that succeeds even in giving an analytical solution.

A Simpler Method

Our last method features an even simpler method of optimizing a single variable function. Since we have different values for Z_x for each of f_0, f_{10}, f_{50} , etc., we can plot these points and fit a single variable function $Z_x = h(x_3)$. As Z_x represents a conglomerate metric for the entire Greater Seattle area, it effectively collapses the z, x_1 , and x_2 variables all into one. As such, h functions as Seattle's SDC Potential Function.

The plot and fit of the various values for Z are shown below:

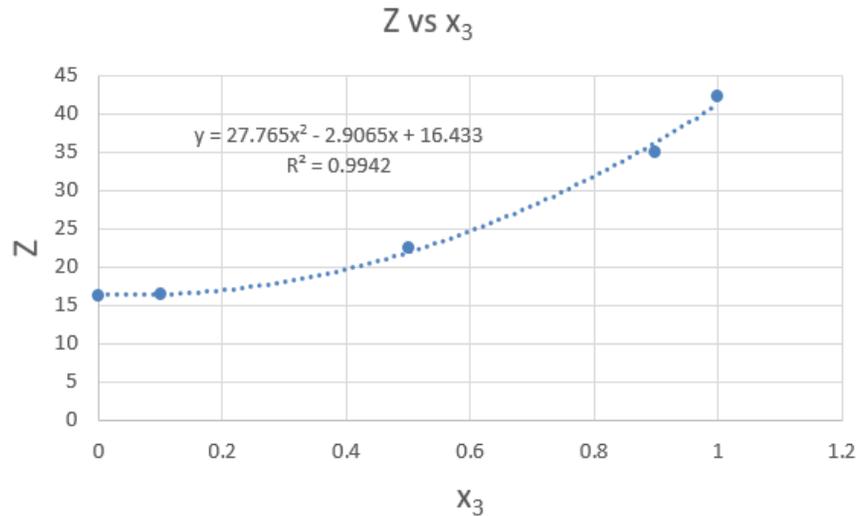


Figure 23: Z vs x_3

Defining $h(x_3) = 27.765(x_3)^2 - 2.9065(x_3) + 16.433$, we can again use our standard calculus optimization techniques.

$$h(x_3) = 27.765(x_3)^2 - 2.9065(x_3) + 16.433$$

$$h'(x_3) = 55.530x_3 - 2.9065$$

$$h'(c) = 55.530c - 2.9065 = 0$$

$$c = \frac{2.9065}{55.530}$$

$$c = 0.05234$$

$x_3 \cdot 100$	$h(x_3)$
0	16.433
5.234	16.357
100	41.292

Table 5: Weighted Average Speed for the Extrema of the SDC Potential Function

The calculus shows us that the maximum is, as seen on the graph, when $x_3 = 1$. In addition, we see that the global maximum is not at $x_3 = 0$, but near $x_3 = 5$. Taking Z as a proxy for overall traffic performance, this

indicates that adding a small number of self-driving cars actually reduces the traffic performance of Seattle highways. However, as soon as we pass a threshold around 10%, performance will drastically increase from the 0% case.

Evaluation of the Model

Sensitivity Analysis

A major component of our model was showing the dependence of average speed on car density to be a power law distribution. One way to check whether the data really do follow a power law distribution is to model the first half of the data as a power law distribution and to see if the curve successfully models the rest of the data. So, we took the data from the f_0 model with $x_2 = 2$ lanes. Taking the data for values of car density from 0.05 to 0.75 produced the following power law fit:

$$y = .101x^{-1.085}$$

Overlaying this fit over the subset of data produces the following graph, which shows the data, the original fit, and our new fit for the purposes of sensitivity analysis.

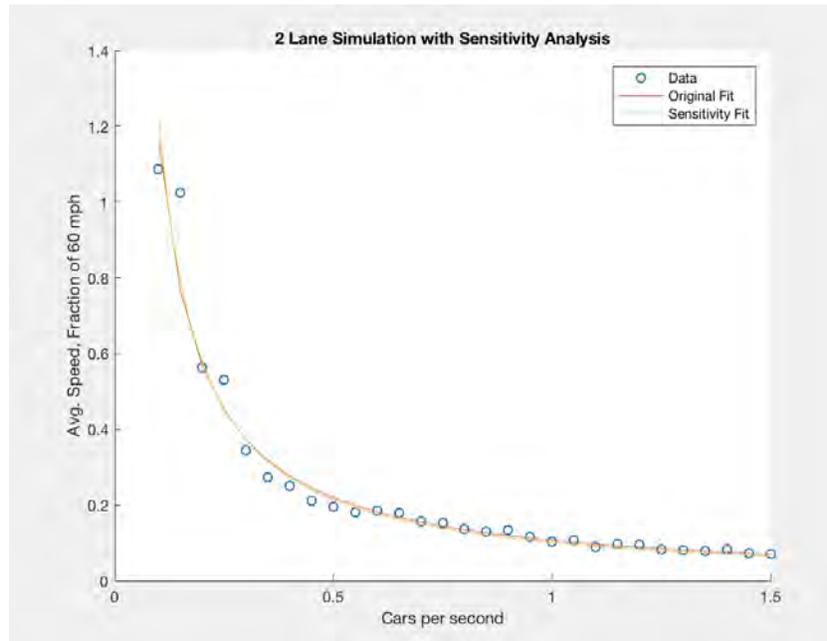


Figure 24: 2DPlot: Data, Original Fit, Modified Fit

Visually, there is almost no difference in the two fits, which shows that the power law fit was the correct choice for this aspect of the model.

Strengths

- The model has implemented a robust traffic rating metric related to the average speed of cars.
- Agent-based modeling allows us to see how initial conditions and limited set of agent actions affect the system's behavior, which directly affects our metric. Additionally, the Netlogo interface is distributable and reasonably user friendly.
- By creating a function approximation of the model, we have allowed for some analytics and tangible interpretations of the model. The function also made our model more portable and easy to use.

-
- The function approximation of our model is optimized to a specific range of traffic data, giving a more accurate picture of the Seattle traffic behavior than a catch-all mean field approximation.
 - This paper has both examined individual highway segments and the highway as a whole, providing both a microscopic and a macroscopic interpretation of the traffic situation in Seattle.
 - Our model is easily expandable to other congested highways and urban centers beyond Seattle, as long as we have data on car density and number lanes. We simply poll our Standard Traffic Model over a region encapsulating City X's traffic data and develop City X Functions.

Weaknesses

- We have not developed a fully analytical model or function offering a catch-all solution to all possible initial conditions.
- Our analytical extensions lend themselves to interpolation, but are not guaranteed to hold for extrapolation.
- Our model does not deal with the absolute worst case scenario at peak traffic times due to the assumption that traffic flow is uniformly distributed throughout the day.
- The constant highway speed of 60 mph is limiting in our model and does not necessarily let us capture all the emergent behavior from the simulations.
- There was limited cooperation between SDC vehicles implemented (only knowing each other's locations). This reduces the effectiveness of our solution due to cars not being able to communicate their future intentions (switching lanes, etc).

Conclusion

Our solution models the traffic conditions given various parameters, including number of lanes, density of cars, and percentage of self-driving, cooperating cars.

First, we created an agent based, computational model for the traffic conditions through which we could run various simulations of varying traffic conditions. We established average speed of cars in equilibrium moving conditions as the metric for the system. After creating the model and running simulations, we fit a function through which we could model the current traffic conditions in Seattle. Afterwards, to judge the system as a whole, we used the weighted arithmetic average of the speeds based on amount of distance the road segment covered.

After this, we modified the model to allow for varying percentages of self-driving cars and repeated the process used before to fit functions to each situation. Using these new functions, we recalculated the average speed values using the Seattle data to predict the change in traffic conditions given varying amounts of self-driving cars. We specifically looked at 10%, 50%, 90%, and 100%. This allowed us to note the general upward trend which showed that self driving cars generally improved traffic conditions.

Lastly, we attempted to put all of the data together to fit a function of three variables to the data, but this proved to be difficult. As a result, we used different techniques including analysis of level surfaces and best fit analysis on the weighted average of speeds to obtain simpler functions to differentiate. Overall, aside from a small decrease in conditions for very low amounts of self-driving cars, once the percentage went over 10% and all the way up to 100%, we saw that increasing the percentage of self driving cars only improved driving conditions.

Overall, our agent based model lent itself very nicely to this problem due to the nature of travelling cars and traffic. Curve fitting techniques allowed us to create functions that were easier to work with and analyze. Limitations include the fact that our techniques lend themselves to interpolation over extrapolation and

averaging traffic volume throughout the day instead of using peak and average times. We did end up seeing that self-driving cars, on the whole, increased quality of traffic conditions in Seattle.

Future Work

To make our model more accurate, we can investigate the distribution of car flow and car density throughout a day (which we assumed to be uniformly distributed). Therefore, we could also investigate peak time (which was given as 8% of the total traffic in a day). We could also try to find a method to implement what was described in the section "Connecting Level Sets". Lane restrictions for SDCs could also be implemented, and various other policy changes could be investigated.

References

- [1] Ackerman, Evan. "Study: Intelligent Cars Could Boost Highway Capacity By 273%". IEEE Spectrum. N.p., 2017. Web. 23 Jan. 2017.
- [2] Consortium for Mathematics and Its Applications, 2017 Mathematical Contest in Modeling, Finalist Solution. <https://www.comap.com/undergraduate/contests/mcm/>
- [3] "Model X Design". Tesla Motors. N.p., 2017. Web. 23 Jan. 2017.
- [4] Sugiyama, Yuki et al. "Traffic Jams Without Bottlenecks—Experimental Evidence For The Physical Mechanism Of The Formation Of A Jam". *New Journal of Physics* 10.3 (2008): 033001. Web. 23 Jan. 2017.
- [5] Tientrakool, Patcharinee, Ya-Chi Ho, and Nicholas F. Maxemchuk. "Highway Capacity Benefits From Using Vehicle-To-Vehicle Communication And Sensors For Collision Avoidance". 2011 IEEE Vehicular Technology Conference (VTC Fall) (2011): n. pag. Web. 23 Jan. 2017.
- [6] Vasconcelos, Luís et al. "Calibration Of The Gipps Car-Following Model Using Trajectory Data". *Transportation Research Procedia* 3 (2014): 952-961. Web. 23 Jan. 2017.
- [7] Wilensky, U. (1998). NetLogo Traffic 2 Lanes model. <http://ccl.northwestern.edu/netlogo/models/Traffic2Lanes>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- [8] Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.