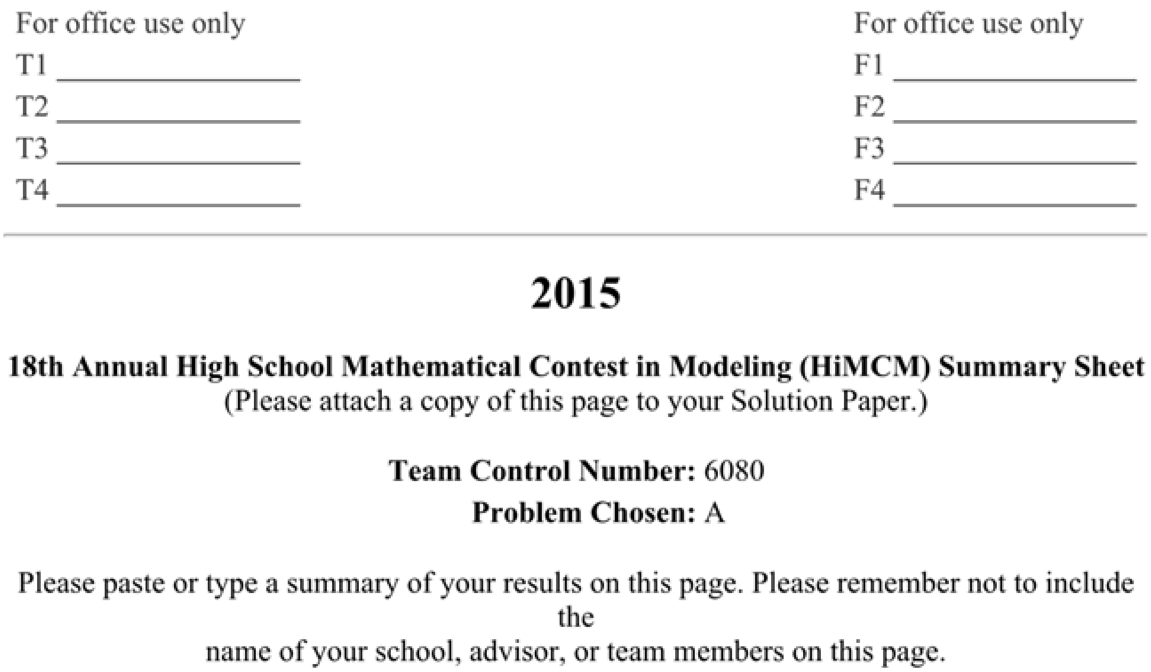


For office use only	For office use only
T1 ________________	F1 ________________
T2 ________________	F2 ________________
T3 ________________	F3 ________________
T4 ________________	F4 ________________

2015

18th Annual High School Mathematical Contest in Modeling (HiMCM) Summary Sheet
(Please attach a copy of this page to your Solution Paper.)

Team Control Number: 6080
Problem Chosen: A

Please paste or type a summary of your results on this page. Please remember not to include the
name of your school, advisor, or team members on this page.

To address the problem of roadway congestion, our team adopted an interdisciplinary approach combining theoretical mathematics, statistics, physics, computational science, and philosophy to perform a fresh analysis of driver actions and their implications in lane closure situations. Beginning with a simple model of road crossing grounded in Newtonian physics, we characterized the relationship between cars on a roadway in quantitative terms. Improving upon this physical analysis, we built computational simulations using NetLogo agent-based modeling to determine the effects of each road crossing strategy across a major highway. Using metrics of fairness and efficiency grounded in mathematical philosophy, including Rawlsian logic and rule utilitarianism, the viabilities of various strategies of road crossing and placements of lane closure were analyzed. We concluded that encouraging cars to merge out of the closing lane as soon as possible and placing the lane closure sign 1 mile away from the closure maximized both fairness and efficiency. We then performed sensitivity analyses on our recommendations to ensure their robustness. To supplement our computational simulations, we created probability density functions to estimate the distributions of cars on a roadway in lane closure situations to create an algorithm usable by policymakers to determine optimal sign placement for a major highway with $n$ lanes and $o$ open lanes, so that our model is easily customizable for public use. When applied to the sample situation of a three-lane highway with two open lanes as compared to one open lane, the model indicates that cars on the highway with n=3 and o=1 should begin to merge sooner than the cars on the highway with n=3 and o=2, so signs indicating a lane closure should be placed further from the lane closure to decrease congestion.

Stay Out of My Lane!
A Mathematical Approach to Road Congestion
2015 High School Mathematical Contest in Modeling

Team 6080

Problem A: Preventing Road Rage

November 2015

Summary Draft

To address the problem of roadway congestion, our team adopted an interdisciplinary approach combining theoretical mathematics, statistics, physics, computational science, and philosophy to perform a fresh analysis of driver actions and their implications in lane closure situations. Beginning with a simple model of road crossing grounded in Newtonian physics, we characterized the relationship between cars on a roadway in quantitative terms. Improving upon this physical analysis, we built computational simulations using NetLogo agent-based modeling to determine the effects of each road crossing strategy across a major highway. Using metrics of fairness and efficiency grounded in mathematical philosophy, including Rawlsian logic and rule utilitarianism, the viabilities of various strategies of road crossing and placements of lane closure were analyzed. We concluded that encouraging cars to merge out of the closing lane as soon as possible and placing the lane closure sign 1 mile away from the closure maximized both fairness and efficiency. We then performed sensitivity analyses on our recommendations to ensure their robustness. To supplement our computational simulations, we created probability density functions to estimate the distributions of cars on a roadway in lane closure situations to create an algorithm usable by policymakers to determine optimal sign placement for a major highway with n lanes and o open lanes, so that our model is easily customizable for public use. When applied to the sample situation of a three-lane highway with two open lanes as compared to one open lane, the model indicates that cars on the highway with n=3 and o=1 should begin to merge sooner than the cars on the highway with n=3 and o=2, so signs indicating a lane closure should be placed further from the lane closure to decrease congestion.

Letter to the Director of the Department of Transportation

Dear Director of the Department of Transportation:

We understand that road rage is a serious issue that arises as a result of lane closures on many roadways. Therefore, we provide you with strategies to alleviate these problems. To arrive at these strategies, we developed a computer simulation and an analytical mathematical model to analyze the effects of different driver behaviors and of sign placement.

Our models suggest that drivers should merge out of the closing lane as quickly as possible. Thus, we urge that driver education materials include information about how drivers should try to merge as quickly as possible once they see the sign, and that they should pay close attention to those around them because others will be attempting to merge. Furthermore, the models suggest that the signs should be placed 1 mile back from the lane closure as cars will travel faster through the closure zone when there is a sign 1 mile back as compared to cars traveling through closure zones with a sign 500 feet away and a sign half a mile away. The increase in speeds of the cars culminates in less traffic jams and less congestion. As another precautionary measure, having a secondary sign closer to the lane closure in addition to the sign 1 mile away could help remind drivers to merge as soon as possible. It is critical to have these signs up and in place before lanes are closed, so there is no delay in response time.

If you were to adopt these measures, traffic jams and congestion that result from lane closures would be decreased in magnitude, and thus immense amounts of road rage would be minimize.

We strongly urge you to take our suggestions as road rage is extremely dangerous and is a hazard for many drivers on the road.

Best Regards,
Team 6080

1 Introduction

1.1 Problem

Road rage is commonplace in our society but has vastly detrimental impacts. Jerry Deffenbacher, Professor Emeritus in psychology at Colorado State University, performed studies on angry drivers and their actions. His studies showed that drivers with road rage engage in hostile thinking, take more risks, and behave more aggressively than drivers without road rage [1]. In a study by the AAA Foundation, a partial sample of all incidents involving aggressive driving between 1990 and 1996 were analyzed. In the sample of 10,037 incidents, 12,828 people were injured or killed [2]. The effects of road rage precipitate the deaths and injuries of a great deal of people.

When driving on a busy road, a lane closure can be a massive source of frustration. Drivers in the closing lane can either merge quickly, which causes the adjacent lane to become congested, or wait until they are closer to the lane closure to merge. Additionally, various actions frequently occur under these stressful conditions, such as a driver in the adjacent lane pulling halfway into the closing lane to prevent cars from passing, or drivers in the closing lane traveling at the same speed as a vehicle directly next to it in the adjacent lane. The objective of the problem is to analyze the different actions and behaviors of the drivers and their impacts on the fairness and the efficiency of the situation under different circumstances. From this information, steps can be taken in an attempt to strategically reduce the amount of road rage that results from lane closures.

1.2 Analysis of the Problem

Our problem-solving strategy was to break down the problem into its component parts and construct a solution to the problem in its entirety from there. The first step involves understanding the basic Newtonian physics behind the scenario

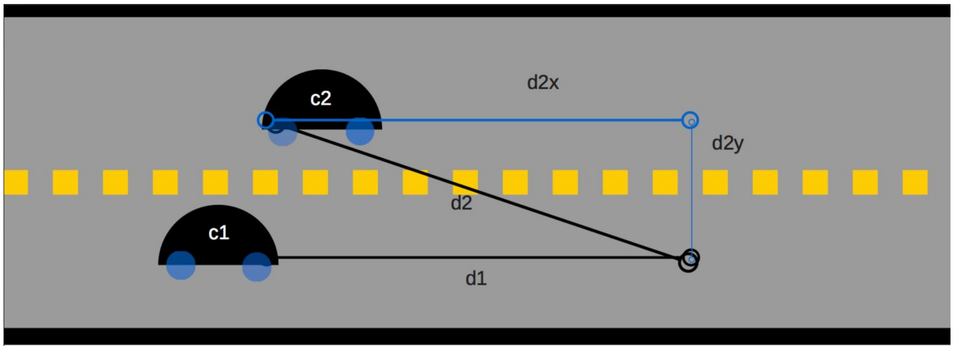


Figure 1: Schematic of simple merging dynamics

A simple version of the decision to merge to an adjacent lane can be modeled by Newtonian physics. When a car (C_2) is attempting to merge into a lane occupied by another car (C_1), it should merge when, for any value of d_{2x}:

$$\frac{d_2}{s_2} < \frac{d_1}{s_1} \text{ where } d_2 = \sqrt{d_{2x}^2 + d_{2y}^2} \text{ and } s_1 \text{ and } s_2 \text{ are the speeds of Car 1 and Car 2}$$

This occurs when the merging car (C_2) will arrive at the other lane before the other car (C_1) arrives at the same point.

The above equation assumes that all parties are travelling at a constant speed throughout the entire encounter, which is unrealistic. In reality, the cars will accelerate or decelerate in reaction to their surroundings. If a_1 and a_2 are the accelerations of Car 1 and Car 2 respectively (values which can be positive or negative to represent acceleration or deceleration) and t_{a1} and t_{a2} are the times over which the respective cars accelerate, the second car should merge while:

$$\frac{d_2}{s_2 + \frac{a_2}{t_{a2}}} - \frac{d_1}{s_1 + \frac{a_1}{t_{a1}}} > 0$$

The maximum acceleration was also determined using Newtonian physics. Assuming a coefficient of static friction of 0.7, the maximum magnitude of the acceleration of the car is approximately 15 mph/s, by the following equations:

$$F = ma = \mu_s mg$$
$$a = \mu_s g$$

Building on these simple physical principles, we constructed an agent-based NetLogo model of cars interacting on a major highway, to ensure that our model was grounded in reality. This model was expanded to simulate various driving actions (shown in Table 1) and analyze their impact on roadway traffic and congestion.

Table 1: Cases and Actions

	Action of Left Lane Car	Action of Right Lane Car
Case 1: “Immediate”	Maintain speed	Merge as soon as possible after seeing sign
Case 2: “Wait”	Maintain speed	Maintain speed and merge at the end near the lane closure
Case 3: “Uniform”	Maintain speed	Merge at a point inbetween the sign and the lane closure
Case 4: “Sit”	Drive between the two lanes to prevent others from passing	May swerve probability of swerving is p probability of not swerving is 1-p
Case 5: “Match”	Maintain speed	Match speed of car in left lane for increased “fairness”

This analysis of cases enhances the understanding of the problem. From here, we build a computational and an analytic model of the scenario, solving various sub-problems along the way. But first, we will create metrics that allow us to analyze the outputs of our models and rank them in terms of fairness and efficiency.

2 Metrics: Mathematical Definitions for Fairness and Efficiency

To better approach this question, we constructed a metric for fairness and a metric for efficiency.

Metric 1 for Fairness: Maximin
The maximum hardship faced by any one car is minimized.

Under this metric, fairness for the least well off in the situation is emphasized, where the maximum hardship for any one car is minimized such that the most unfair situation is as fair as possible. This involves minimizing the maximum amount of time any one car must wait and minimizing the maximum amount of lanes any one car must cross. This is defined by:

$$hardship_c = at_c + bl_c$$

where t represents the wait time for that car, l represents the number of lanes the car must change, and a and b are adjustable constants that can be changed in the model to reflect different perspectives on the impact of wait time versus lane changes.

Justification: A metric for fairness must capture the intersectional nature of mathematics and philosophy that can be applied to public policy in an interdisciplinary approach. To ground these principles, we based this first conception of fairness on the theories posited by Harvard professor of philosophy John Rawls, who studies the application of philosophy to public policy. Rawls conducted a thought experiment known as the “veil of ignorance,” concluding that when determining fairness for society as a whole one must adopt the “maximin” principle, choosing to help the least well off in society to establish security so that the amount of hardship any one individual experiences is minimized [3]. This also minimizes the maximum amount of road rage, and since high levels of road rage are the ones most associated with dangerous situations, the metric minimizes intensity of predicted casualty. However, although this metric is very fair towards the individuals, it is not the most efficient solution for the system as a whole.

Metric 2 for Efficiency
Minimize congestion.

From a societal perspective, minimization of congestion is the way to ensure smooth functioning of the roadways. Congestion can be measured by the amount of extra time a car must spend on the road on average due to other cars, defined as:

$$congestion = \mu_{t.reality} - \frac{d}{s}$$

where $\mu_{t.reality}$ is the average time a car spends travelling over a distance d, and s is the speed limit. Using this equation, congestion is the time taken in reality – the time taken if there were no other cars.

In situations that hold distance and speed limit constant, the $\frac{d}{s}$ does not change, so comparison of congestion can be determined by comparison of $\mu_{t.reality}$.

Justification:This metric can also be justified from a fairness perspective based on rule utilitarian philosophy. As described in the Stanford Encyclopedia of Philsophy, many philosophers advocate for a form of consequentialism known as rule utilitarianism, where the rules for society must be formed to maximize overall wellbeing and minimize hardship, and where individual agents must then adhere to these rules to ensure fairness[4]. This metric creates a universal rule that will minimize the amount of hardship experienced for each car (agent), and is thus supported from both a mathematical and a philosophical perspective. In this manner, our metric for efficiency also takes fairness into account.

3 Computational Model

3.1 Assumptions

Assumption: Cars accelerate or decelerate in relation to cars in front of them.
Justification: Cars attempt to match the speed of the car directly in front of them and accelerate or decelerate in accordance with that car to maintain a space between them and avoid crashes. Additionally, this ability to accelerate or decelerate is observed in cars due to the presence of an accelerator pedal and a brake.

Assumption: Signs indicating a lane closure can be placed at places including 500 feet, 2640 feet, and 5280 feet from the lane closure.
Justification: The three distances of 500 feet, a half-mile, and a mile were given as distances to test in the problem description.

Assumption: Cars in the left lane do not slow down to allow for cars from the right lane to merge.
Justification: Cars in the left lane have right of way, and thus assume that it is the responsibility of the driver in the right lane to determine when it is safe to merge.

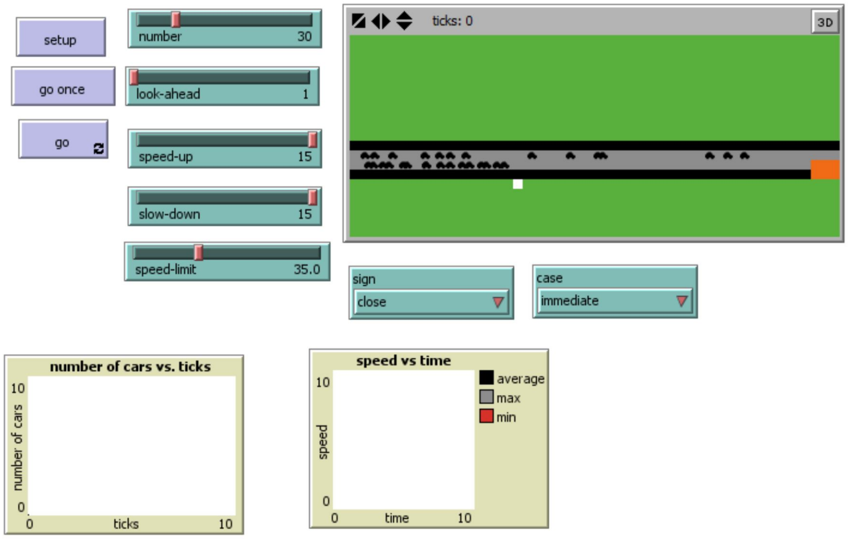
Assumption: Cars adhere to the speed limit.
Justification: This is a reasonable assumption to make because most cars will do so; however, even if some cars are speeding, this can be taken into account by the easily adjustable speed limit user input.

Furthermore, since the model was used to compare the effects of various merging "strategies", the effects of these assumptions should not drastically affect the relationships determined, ex. strategy "wait" taking a longer time for all the cars to clear than strategy "immediate".

3.2 Design

To compare the various driver strategies, a computational agent-based model founded in the principles of Newtonian physics and individual driver decisions was constructed, using NetLogo software. In this model, cars are assigned random speeds under a speed limit that is adjustable via a slider, with the maximum speed limit value being 100. There are two lanes which impact each other, a right (bottom) and a left (top) lane. Cars in the top lane have the right of way and thus maintain their speed, while cars in the bottom lane look for an opportunity to merge onto the top lane according to the set strategy. Cars in the right lane (which must merge) have three possible strategies, as described in the problem analysis. Cars accelerate if they are given the chance to because there is no car directly in front of them. Otherwise, if there is a car directly in front of them, they match the speed of the car in front and decelerate to allow for cushion time between cars. Cars can accelerate until they reach the speed limit, which they may not exceed. Cars can accelerate and decelerate below the speed limit to adjust to other cars in front of them at rate of up to 15 mph per second as determined by a slider. The maximum acceleration magnitude of 15 mph is determined using the equations determined in the Newtonian model, and with a coefficient of static friction of .7, we find that acceleration = .7*9.8 $\frac{m}{s^2}$=15 $\frac{mph}{s}$

In short, car motion is affected by car strategy, number of cars, maximum acceleration and deceleration ability, speed limit, and distance to the lane closure sign, and these factors are all adjustable in the user interface to test a wide variety of situations.



number 30

look-ahead 1

speed-up 15

slow-down 15

speed-limit 35.0

sign close

case immediate

Figure 2: Base case NetLogo simulation with two lanes and one lane closure. In this case the strategy chosen is “immediate”, so cars in the bottom (right) lane attempt to merge into the top (left) lane as soon as possible after seeing the lane closure sign, shown here as a white patch.

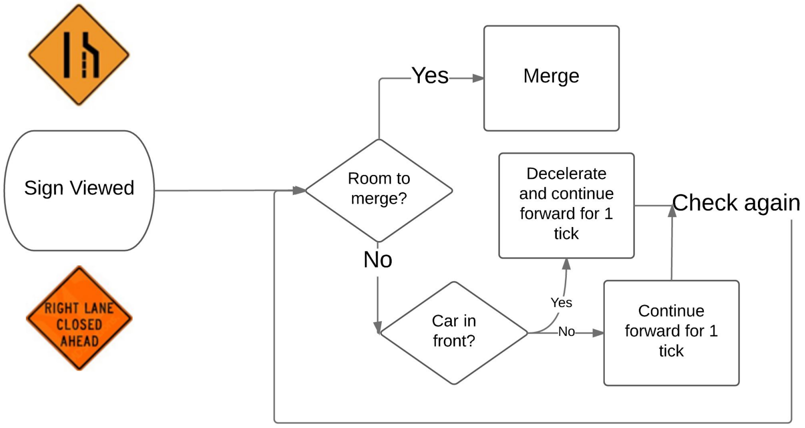
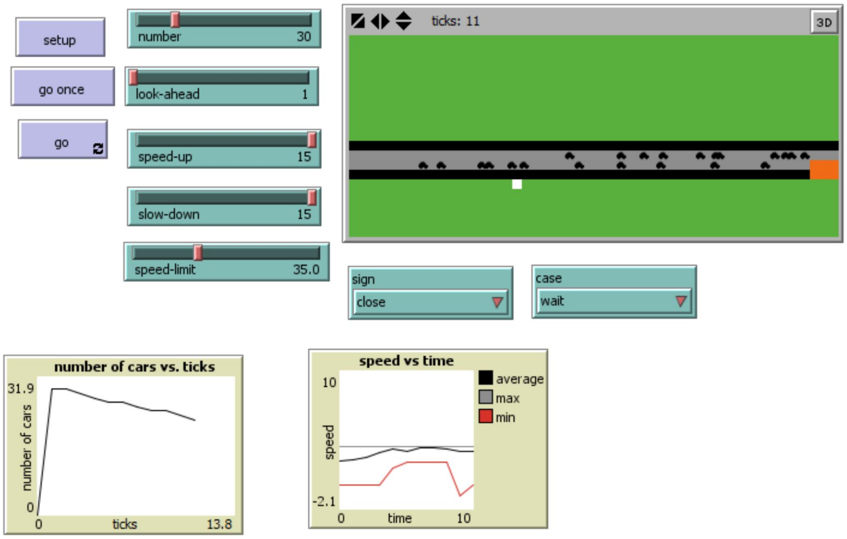


Figure 3: Logic diagram of the “immediate” strategy: Car in the closing lane merges as soon as possible.



number 30

look-ahead 1

speed-up 15

slow-down 15

speed-limit 35.0

sign close

case wait

Figure 4: Base case NetLogo simulation with two lanes and one lane closure. With the “wait” strategy selected, cars wait until they see the actual lane closure to merge.

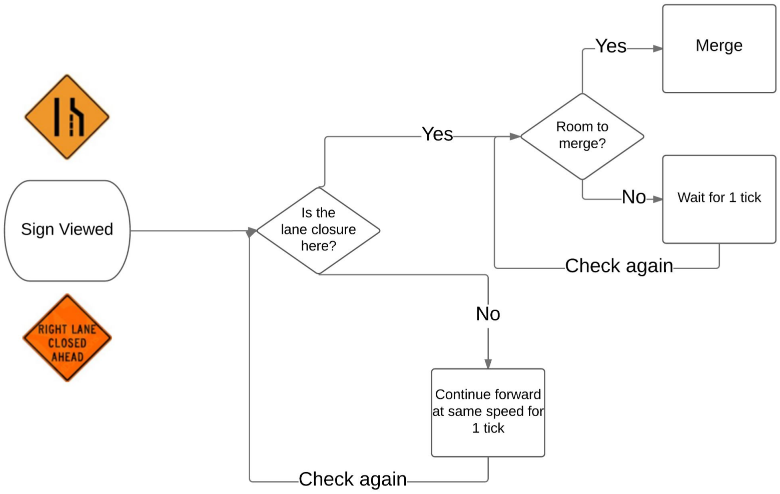
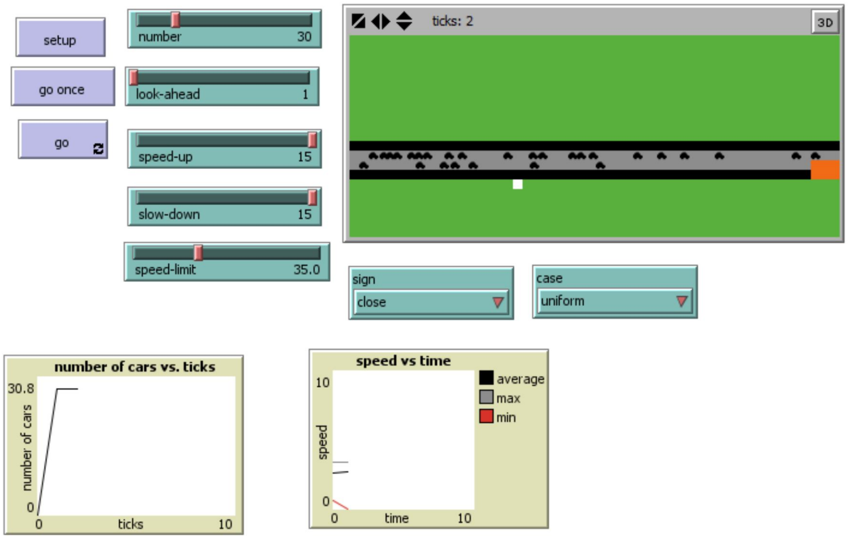


Figure 5: Logic diagram of the “wait” strategy: Car in the closing lane waits until the lane closure to merge.



number 30

look-ahead 1

speed-up 15

slow-down 15

speed-limit 35.0

sign: close

case: uniform

Figure 6: Base case NetLogo simulation with two lanes and one lane closure. The "uniform" strategy selected allows to cars to merge into the upper lane at a randomly selected point inbetween the sign and the lane closure, modeling individual decisions of drivers deciding to merge at different times.

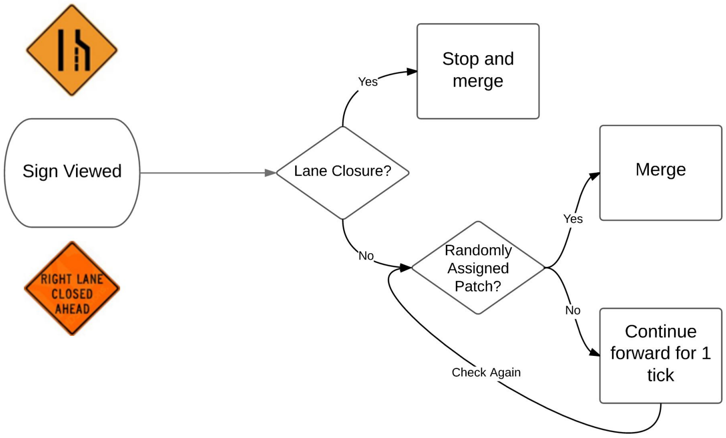


Figure 7: Logic diagram of the "uniform" strategy: Car in the closing lane merges at a randomly selected point between the sign and the lane closure.

3.3 Simulation 1: Vary Car Strategy at 65 mph for "Highway"

We ran a simulation that compared the implications of three merging strategies on congestion on a highway. In strategy 1, cars in the right lane immediately try to merge after the sign signifying lane closure. In strategy 2, cars pass the sign and do not try to merge until the actual lane closure is directly in front of them. In strategy 3, cars merge at some point between viewing the sign and the actual lane closure, in a random manner on the agent level wherein each car has an equal probability of merging at any given point in the road, creating a uniform distribution. In all instances, once the car decides it wants to merge, it begins checking the left lane for a free spot and merges as soon as there is an opening in the left lane to merge. This simulation involved 50 runs for each strategy (150 runs total), controlling the number of cars (30 cars distributed across three lanes), closure (the right-most lane being closed at the end), speed limit (65 miles per hour) and sign (placed 500 feet from the lane closure) so we could compare the agent strategies. Results were measured in terms of total congestion time, measured by how many seconds (ticks) it took for all the cars to pass through the botteneck and then analyzed in comparison to fairness (via maximin) and efficiency (via total congestion).

3.3.1 Results of Simulation 1

After 150 runs, the following data was collected, as summarized below.

Table 2: Data Summary of Varying Strategy at 65 mph

Strategy	**Mean Total Time**	**Standard Deviation**
Strategy 1 ("Immediate")	10.44 seconds	1.2
Strategy 2 ("Wait")	12.98 seconds	1.9
Strategy 3 ("Uniform")	10.72 seconds	1.4

Qualitatively, it is clear strategy 2 took longer than the other two strategies to clear up congestion.

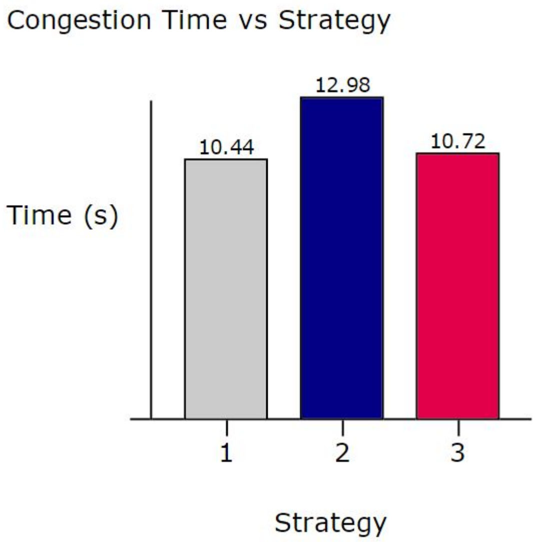


Figure 8: Congestion Times at 65 mph

The next step is significance determination. A Tukey HSD test was conducted, with HSD[.05] = 0.73 and HSD[.01] = 0.91. The following table summarizes the results:

Table 3: Tukey HSD Results for 65 mph

Strategy Comparison	**Significance**
Strategy 1 vs. Strategy 2 ("Immediate" vs "Wait")	p < .01 is significant
Strategy 1 vs. Strategy 3 ("Immediate" vs "Uniform")	p is not significant
Strategy 2 vs. Strategy 3 ("Wait" vs "Uniform")	p < .01 is significant

Although the effects of driving immediately and driving at some random (uniformly distributed) time were not significantly different, the effect of waiting until the lane closure was significant enough in terms of congestion time. This was also qualitatively observed in the visual NetLogo, with strategy 2 ("wait") resulting in more obvious bottlenecks. The data indicate that in similar circumstances, encouraging drivers to merge as soon as possible rather than waiting until the last second is likely to lead to shorter wait times for them and lower congestion overall, satisfying both fairness and efficiency.

3.4 Simulation 2: Vary Car Strategy at 35 mph for "Secondary Road"

We ran another simulation that compared the same three driving strategies, but this time at a speed of 35 miles per hour. In strategy 1, cars in the right lane immediately try to merge after the sign signifying lane closure. In strategy 2, cars pass the sign and do not try to merge until the actual lane closure is directly in front of them. In strategy 3, cars merge at some point between viewing the sign and the actual lane closure, in a random manner on the agent level where each car has an equal probability of merging at any given point in the road, creating a uniform distribution. In all instances, once the car decides it wants to merge, it begins checking the left lane for a free spot and merges as soon as there is an opening in the left lane to merge. This simulation controlled the number of cars (30 cars distributed across three lanes), closure (the right-most lane being closed at the end), speed limit (35 miles per hour) and sign (placed 500 feet from the lane closure) so we could compare the agent strategy. Results were measured in terms of total congestion time, measured by how many seconds (ticks) it took for all the cars to pass through the botteneck. This allowed us to compare efficiency as based on metric 2, from a societal perspective. This also satisfies metric 1, by comparing maximum times taken across cars, seeking to minimize this time. In this manner, both the fairness and efficiency were compared using one dependent variable.

3.4.1 Results of Simulation 2

The following data was collected, as summarized below.

Table 4: Data Summary of Varying Strategy at 35 mph

Strategy	**Mean Total Time**	**Standard Deviation**
Strategy 1 ("Immediate")	27.8 seconds	2.7
Strategy 2 ("Wait")	24.9 seconds	2.9
Strategy 3 ("Uniform")	27.6 seconds	4.8

Qualitatively, the "wait" strategy took slightly less time, but all three values are extremely close.

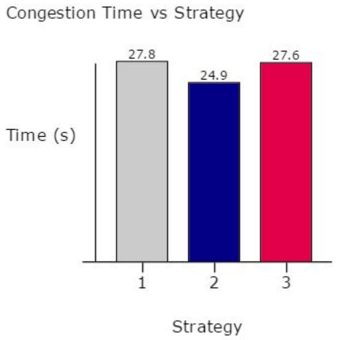


Figure 9: Congestion Times at 35 mph

The next step is significance determination. A Tukey HSD test was conducted, with HSD[.05] = 2.98 and HSD[.01] = 3.78. The following table summarizes the results:

Table 5: Tukey HSD Results for 35 mph

Strategy Comparison	**Significance**
Strategy 1 vs. Strategy 2 ("Immediate" vs "Wait")	p is not significant
Strategy 1 vs. Strategy 3 ("Immediate" vs "Uniform")	p is not significant
Strategy 2 vs. Strategy 3 ("Wait" vs "Uniform")	p is not significant

Thus, we are unable to find a difference between the driving strategies that is statistically significant enough to make a policy recommendation, because any one of the recommended strategies equally affects congestion. However, although this model with the slow speed limit (35 mph) was not affected by driver strategies, faster speed limits (65 mph) seem to be affected by driver strategy, where merging as soon as possible or merging uniformly at some point between the sign and the lane closure are preferable to waiting until the last second. Thus, in terms of overall societal benefits across speed limits, we recommend the policy option of encouraging drivers to not wait until the lane closure is rapidly approaching to change lanes.

3.5 Simulation 3: Vary Sign Placement

We then ran a simulation that compared three sign placements. The first was a "500 ft" placement, at 500 feet from the lane closure. The second was a "half-mile" placement, at 2640 feet from the lane closure. The third was a "mile" placement, at 5280 feet from the lane closure. In all instances, once the car decides it wants to merge, it begins checking the left lane for a free spot and merges as soon as there is an opening in the left lane to merge. This simulation involved 100 runs for the close placement, 100 runs for the half-mile placement, and 65 runs for the mile placement (265 runs total). The number of cars (30 cars distributed across two lanes), closure (the right-most lane being closed at the end), total distance travelled (5400 feet), driver strategy (strategy 3, a uniform distribution of car merge decisions, where there is an equal probability of an agent deciding to merge at any patch between the sign and the lane closure so that neither extreme is favored), and speed limit (70 miles per hour) were held constant so we could accurately compare the sign distances. Results were measured in terms of average speed of a car in feet per second for each run, such that they are comparable under both the fairness and efficiency metrics.

3.5.1 Results of Simulation 3

After 265 runs, the following data was collected, as summarized below.

Table 6: Data Summary

Sign Distance	Mean Speed	Standard Deviation
Strategy 1 (“500 ft”)	85.3 ft/s	8.5
Strategy 2 (“Half-mile”)	111.6 ft/s	8.8
Strategy 3 (“Mile”)	132.9 ft/s	6.2

Qualitatively, it is clear that placing a sign earlier led to faster car speeds and decreased congestion.

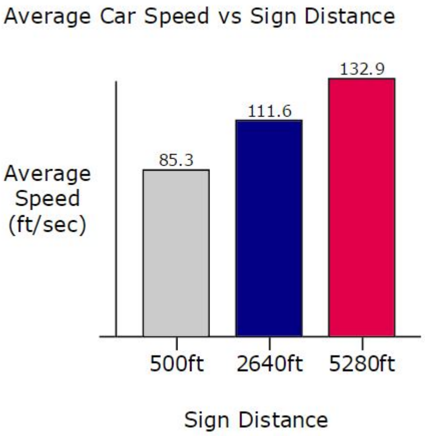


Figure 10: Congestion times

The next step is significance determination. A Tukey HSD test was conducted, with HSD[.05]=2.94 and HSD[.01]=3.66. The following table summarizes the results:

Table 7: Tukey HSD Results

Distance Comparison	Significance
Distance 1 vs. Distance 2 (“500ft” vs “Half-Mile”)	p < .01 is significant
Distance 1 vs. Distance 3 (“500ft” vs “Mile”)	p < .01 is significant
Distance 2 vs. Distance 3 (“Half-Mile” vs “Mile”)	p < .01 is significant

This shows that moving a sign further back to even 1 mile from the lane change should significantly increase car speed. Thus, distance 3 of the sign, 1 mile from the lane change, best achieves both the fairness metric by decreasing maximum hardship (since the maximum car driving time is minimized) and the efficiency metric (by decreasing overall congestion).

3.6 Simulation 4: Sensitivity Analysis

To test the sensitivity of our model, we ran a fourth simulation that repeated the procedure from the previous simulation but varied the number of cars (to 5, 10, 15, 20, 25, 30, 35, and 40 cars instead of holding cars constant at 30) to determine the sensitivity of our model in response to traffic changes hour traffic or post-midnight absences of traffic. We sought to determine whether our recommendations would hold true in these scenarios. The following graph summarizes our findings:

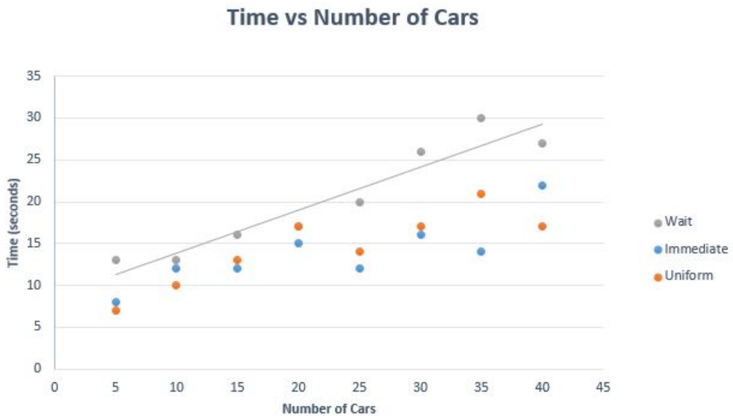


Figure 11: Sensitivity Analysis with Varying Numbers of Cars

As demonstrated by the grey line, the strategy to wait always led to longer total times and were thus associated with higher congestion levels. Thus, we conclude that the recommendation to not wait until the lane closure is directly ahead to merge lanes, and rather to merge earlier if possible, is robust with regards to varying traffic conditions.

3.7 Strengths

The model has high applicability to a variety of situations because of its customizability. The number of cars can be adjusted to account for different road densities. The slow down and speed up time can be adjusted according to user input; this could be affected by the physical constraints of the cars and the relative attentiveness of the driver. This allows for the model to be easily expanded to situations in which drivers may have slower reaction times as a result of ex. texting and driving. The amount that a driver can see in front of them can also be easily changed by user input, ex. affected by time of day. The speed limit of the road in question can be modified for increased application to any type of road. Multiple merging strategies were taken into account (cases 1, 2, and 3). Finally, cars affect other cars in their lane in the model as they do in real life, separating into an exponential distribution naturally as cars aimed to get as close as possible to each other without crashing.

3.8 Weaknesses

Multiple lanes were not taken into consideration– if this were a highway, if the left lane were to get congested, another lane may exist in which cars from this lane could decide to move into another lane. The behavior of the cars in the left lane is assumed to remain unchanged unless a car from the right enters the lane and it is thus affected by it; in real life the cars in the left lane would also be able to make their own decisions about whether they wanted to decelerate for those in the right lane to merge (though unlikely) or sit on the lane to aggressively encourage drivers in the right lane to merge. A binomial situation was not taken into account wherein cars are more likely to merge either when they first see the sign or at the end of the lane.

4 Analytical Model

We will now approach the problem from a purely mathematical standpoint in order to approximate car densities on the road. In order to accomplish this, we must make additional assumptions. Here, density denotes cars per distance in a lane.

4.1 Assumptions

Assumption: The distance between cars can be approximated by an exponential probability distribution.
Justification: Cars will generally accelerate until they are traveling as close as comfortably possible to the car in front of them to minimize total travel time.

Assumption: If a car can merge left, it will.
Justification: This assumption is made to make the mathematical analysis possible. By making this assumption, the analytical model is restricted to the merge as soon as possible strategy. This limits the model, but shows the results of traffic patterns when drivers obey this rule.

Assumption: Car densities do not change significantly between two adjacent cells in a particular lane, when the road is divided into a grid.
Justification: This assumption is made to make the algorithms that run later more tractable. Also, the number of cars that are able to merge left is compensated for with cars merging into that lane. Thus, we obtain an approximation.

Assumption: Cars can not move forward to pass a car in an adjacent lane.
Justification: When driving, a car can not force the car in front of it to speed up by speeding up into it. Because a car can not guarantee that it can move forward relative to adjacent cars, we assume that it can only move back relative to adjacent cars. This is especially true when the density of cars is high. If it is low, the car can usually just move to the left; if it can not, then it would take just as much time to move forward as it would to move back on average. Because forward movement relative to surrounding cars can not be guaranteed, cars will move back to merge more often than they would move forward. This assumption is made to make the problem analytically solvable.

Assumption: Each cell in the road grid has a constant density.
Justification: By making a constant density within a cell, we discretize the model and make the problem more mathematically solvable. This has no impact on the type of distribution used to model the cars. Also, with relatively small cells, we can approximate the density at any point.

4.2 Variables

n= number of lanes

q= number of cells in a lane

$\lambda = \frac{1}{Mean\ Distance\ Between\ Cars}$

$x =$ distance from car in front of you - $\frac{1}{\lambda} - car_{length}$

$you =$ distance of the front of your car to the road block

car_{front}= distance of the front of the car in front of you to the road block

$Relative\ Velocity =$ velocity of cars in lane left of you - your velocity

Density(cell)= the car density in a cell

4.3 Determination of Variables

We know that the distance between cars in a lane can be approximated by an exponential distribution:

$$F(x, \lambda) = \lambda * e^{-\lambda * x}$$

We also know that if the following inequalities are true for a car, then that car is able to move one lane to the left.

$$x + \frac{1}{\lambda} + car_{front} + length > you + car_{length}$$

$$car_{front} + car_{length} < you$$

The first equation simplifies to:

$$x > you - car_{front} - \frac{1}{\lambda}$$

Let us denote τ_1 the minimum value that satisfies this inequality. Thus, the probability that the car can change lanes is the integral from τ_1 to ∞ of the probability distribution F(x,λ) with respect to x.

$$p = \int_{\tau_1}^{\infty} F(x, \lambda) dx = -e^{-\lambda * x}|_{\tau_1}^{\infty} = e^{-\tau_1 \lambda}$$

Because we can not speed up to pass cars if a car is in front of us, we can only slow down. The person behind us is forced to slow down to avoid a car accident. Now we must solve for the probability that we can merge when slowing down. The following inequalities describe when the car is able to merge after slowing down.

$$x + \frac{1}{\lambda} > car_{length} + c$$

where c is a small safety coefficient. This gives:

$$x > -\frac{1}{\lambda} + car_{length} + c$$

These inequalities satisfy the conditions as merging, in this case, is only dependent upon the distance between the two cars that the car is trying to move between. This is due to the fact that the car is starting immediately behind the car that it slowed down to let pass. Now, let τ_2 be the smallest x value for which the above inequality is true. The probability that we can merge after slowing down is given by the following equation:

$$q = \int_{\tau_2}^{\infty} F(x, \lambda)dx = -e^{-\lambda * x}|_{\tau_2}^{\infty} = e^{-\tau_2 \lambda}$$

4.4 Car Density

We will begin by discretizing the model and formulating the road as a grid. This will make the problem more tractable and allow us to account for probabilities that change with lane number and distance to the road block when modeling the problem.

First, we will formulate the road as a grid. Each cell of the grid is has an index of the following format: (lane, k), k>0. Let the left most lane have lane=1. Let the cells with k=1 occur immediately after the drivers come in contact with the sign. Now, let us create a row with k=0, where all entries in the row the car density in each lane. This is assumed to be equal before the drivers come in contact with the sign (k=0). Now that we have the structure of the problem, we will proceed to solve the problem in the following sections.

4.4.1 Car Density Algorithm

The car densities in each cell are dependent upon multiple variables and states of the previous, surrounding cells. Thus, we will create an algorithm to compute the car density distribution on the road.

Let us first determine the length of the cell. We will treat this as the average distance traveled before a car would be able to change lanes. The maximum time and minimum time are given by:

$$cell_{length} = \frac{2car_{length}}{Relative\ Velocity} \text{ and } cell_{length} = 0 \text{ respectively.}$$

Because the initial positioning of cars between lanes is assumed to be uniform, the average time is the average of the previous two equations. This gives:

$$cell_{length} = \frac{car_{length}}{Relative\ Velocity}$$

Now that the grid has been sized, we will move to the algorithm. A flow chart representation is shown below to show how densities are determined.

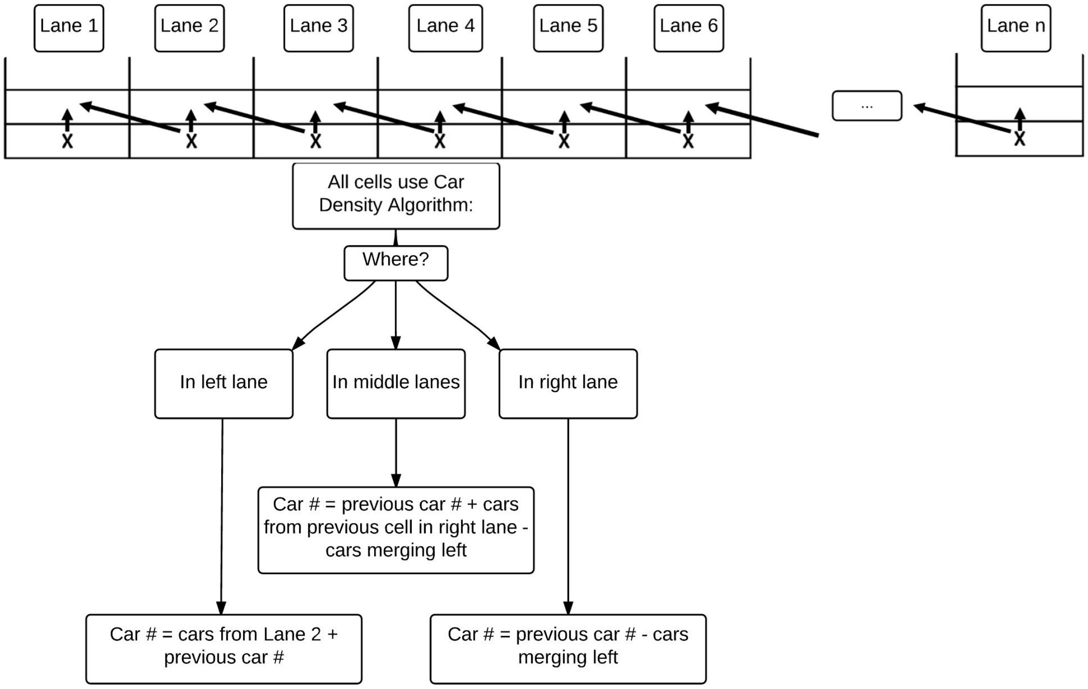

Lane 1
Lane 2
Lane 3
Lane 4
Lane 5
Lane 6
Lane n
...
X
All cells use Car Density Algorithm:
Where?
In left lane
In middle lanes
In right lane
Car # = previous car # + cars from previous cell in right lane - cars merging left
Car # = cars from Lane 2 + previous car #
Car # = previous car # - cars merging left

Figure 12: Car Density Algorithm

The algorithm is run in a program and is used on every cell in the grid. The program starts with cell (1,1) and orders them as follows in an n by q grid:

```
add (1,1) to sequence
apply the following iteratively
(lane number, k) is the last cell in sequence
if lane number$\neq$n then lane number=lane number+1
add cell to sequence
if lane number = n then lane number = 1 and k=k+1
add cell to sequence
if lane number = n and k=q then end
```

Example output is shown below:
(1,1)(2,1)...(n-1,1)(n,1)(1,2)(2,2)...(n-1,k)(n,k)(1,k+1)...(n,q) for some k, where k represents a cell number in a lane.

The program then applies the algorithm to each element of the sequence in order.

4.4.2 Implicit Density Calculations

The implicit determination of the number of number of merging cars will be explained here.

The probability of merging left for any k is proportional to $p_{lane,k}$.

$$p_{lane,k} = \int_{\tau_{1(lane,k)}}^{\infty} F(x,\lambda)dx$$

$$\tau_{1lane,k} = you - car_{front} - \frac{1}{\lambda_{lane-1,k}}$$

This says that the number of cars merging to the left is proportional to the average distance between the cars in the adjacent lane on the left. This is a function of the car density in a cell. The program then uses the car density and the length of the cars to find λ in order to recompute $\tau_{1lane,k}$. From there, the program integrates to find $p_{lane,k}$. The density of cars that merge left is given by

$$Density(lane,k) * p_{lane,k}$$

The program computes density of cars that merge instead of number of cars because the size of the cell is constant and there is a ratio between the two. This is the same as finding density times probability times size of cell divided by size of cell, which results in the previous equation. Thus, the program avoids this and works in terms of car densities.

The number of cars entering cell (lane, k) is equal to the number of cars leaving cell (lane+1, k-1). Thus, we calculate this in a similar manner. Because the program can not use the new car density to calculate the new car density, it relies on the previous car density to do this calculation. This relies on the assumption that there will not be large density changes between cell (lane, k) and cell (lane, k+1). However, this allows the program to approximate the car density values in the following cells. The algorithm uses these steps implicitly to calculate how densities change as the lane number and k change.

4.4.3 Car Density Map

The following is a sample map of car densities obtained from the algorithm using Excel.

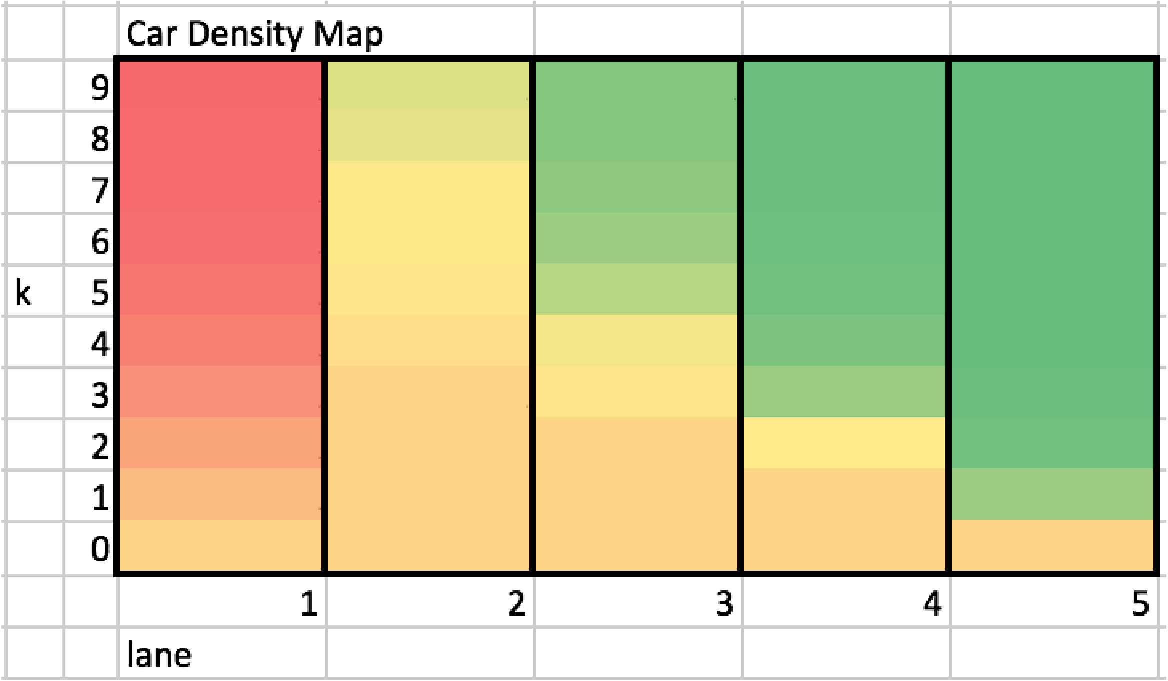


Figure 13: Car Density Map

Here, dark green denotes minimal density (almost no cars), yellow is the car density before the sign, and dark red is high car density.

4.5 Additional Results

Using the probabilities, we can determine the expected number of times that a car would have to slow down and let cars on the left pass before the car is able to merge into the left lane.

$$\begin{gathered}\textit{Expected Slow Down Number} = (1-p)\sum_{k=0}^{\infty} k(1-q)^k = \frac{(1-p)(1-q)}{q} \\ = (1-e^{-\tau_1\lambda})(1-e^{-\tau_2\lambda})e^{\tau_2\lambda}\end{gathered}$$

This result tells us that the cars in one lane will move at a slower rate than those lane immediately left of it in order to join the adjacent lane. However, our model does not take into account cars being able to speed up and pass adjacent cars. Because any car can not force this situation, but it can guarantee that it can slow down to merge, we postulate that the result still holds when employing this strategy as more cars will move back to merge. This trend is also observable when commuting to school or work during rush hour, which adds validity to the description of the model.

4.6 Implications

By creating an algorithm to determine the probable densities of cars at any point in any lane we can apply this analysis to determine where optimal sign placement would be for a variety of cases.

4.7 Strengths

The analytical model allows us to approximate the density distribution of cars on the road when they employ the merge left as soon as possible strategy. This gives us a method for which to look at general traffic flows and abstracts the problem to n lanes with b lanes blocked. By abstracting the problem, we can look at the core of the traffic behavior. In addition to that, we are given another method for which to analyze strategies. Having multiple analyses allows us to analyze the problem from multiple perspectives. Additionally, this method allows us to approximate an optimal sign postage distance in order to minimize traffic.

4.8 Weaknesses

The analytical approach requires a few inherent weaknesses. The greatest weakness is its great increases in complexity when limiting assumptions are removed. With much more analysis and results from computational models, we may be able to gain better approximations using statistics in order to extend the model.

The analytical model constrains the problem to discrete values, which results in approximations. However, this makes the problem more tractable. These approximations worsen with distance after the sign because the new approximations are made on old approximations. This is similar to applying Euler's method to a function of three variables. As long as the slope of the continuous car density function is not too great and the step size too great for the method to capture the behavior of the function, it will be relatively accurate. We also know that the method produces a lower bound for values in the lower left triangle of the grid because of the increasing slope, and an upper bound for values in the upper right triangle of the grid because of the decreasing slope. Because information on the actual function is sparse, little is known about the accuracy of the approximations with distance after the sign. However, the structure of the algorithm suggests that the approximation is relatively accurate. The slope between the cell (lane, k) and cell (lane, k-1), the slope between cell (lane, k) and cell (lane+1, k-1), and the slope between cell (lane, k) and cell (lane -1, k+1) are relatively small for the step size. This suggests that because the algorithm approximates new values based on these cells, and the slopes between them are small, the algorithm closely approximates the values of the actual car density function.

The analytical model also neglects the fact that cars may be able to move forward in order to pass a car on the left. This was removed from the problem in order to improve tractability. Including this would result in greater merging, but only marginally so. As stated in the justification of this assumption, because the cars can not force forward movement relative to adjacent cars, the car is less likely to be able to move forward. This probability changes with average car density on the road. Thus, this assumption has little affect on the model when the average car density is high. The impact of this assumption also decreases as the cars approach the road block as car density increases dramatically in the left lanes. Although this assumption results in error, the error is minimized by large car densities and the bottleneck effect of the road block.

4.9 Sample Calculations: Three-lane highway reduced to two lanes and three-lane highway reduced to one lane

When observing drivers strategies from the analytical model, we are able to apply the methods for n total lanes and b blocked lanes. Let us denote o=n-b. We will solve the problem for n=3 and o=1,2 and for n=2 with o=1. The results are shown below.

n=2, o=1:

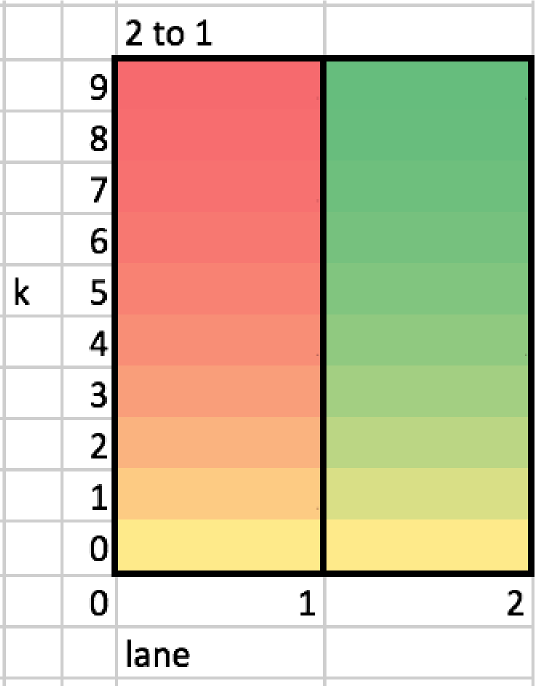


Figure 14: caption

n=3, o=1:

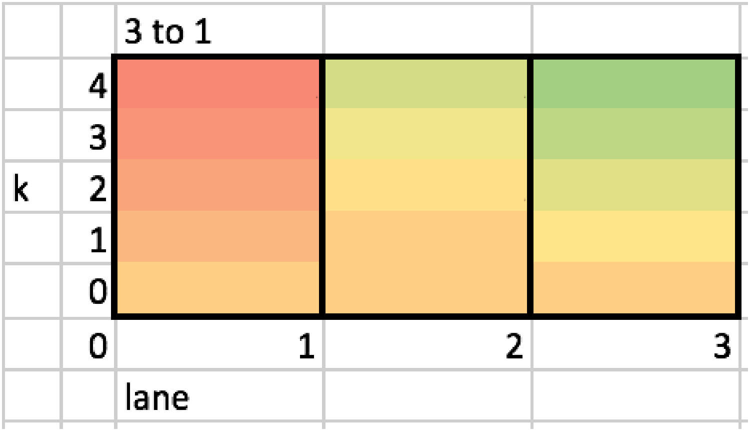


Figure 15: caption

n=3, o=2:

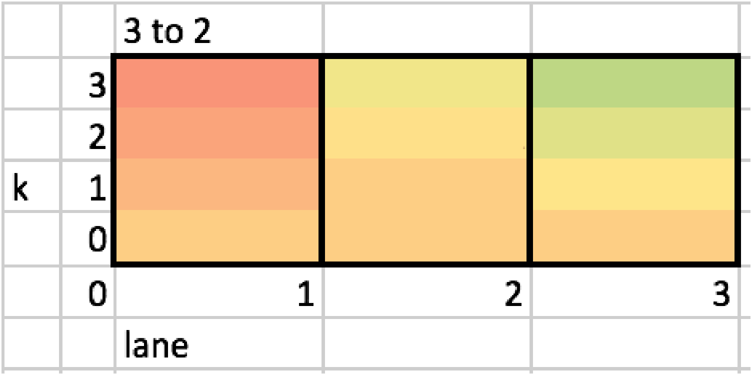


Figure 16: caption

4.9.1 Conclusions

These images show that when n=3 and o=2, cars achieve a distribution of cars that allows them to minimize buildup when k=3. However, when n=3 and o=1, the cars do not achieve a distribution that minimizes buildup until k=4. Because both tests were run under the same conditions, we can conclude that it takes longer for the cars to fit into one lane from 3. We can also determine the actual length of the cells because of the conditions. This tells us that the cars on the highway with n=3 and o=1 should begin to merge sooner than the cars on the highway with n=3 and o=2. This would result in less traffic. Using this information, signs could also be placed earlier in order to maximize the flow of traffic.

5 Suggested Guidelines

5.1 Guidelines for Drivers' Education Materials

When approaching a lane closure, it is advisable to begin merging as soon as possible. This will save you overall time and will also minimize congestion, saving time for those around you. Additionally, when approaching a lane closure, you should be mindful of those around you since most drivers in the closing lane should be attempting to merge as soon as possible, as should you.

5.2 Guidelines for Signage

Signs should alert drivers of the lane closure and advise them to merge as quickly as possible. We considered the following options for sign placement: 500 feet from the 2 to 1 lane closure, half a mile from the lane closure, and 1 mile from the lane closure. Of these options, the sign should be placed 1 mile away from a lane closure on a highway. Inclusion of another sign soon after, ex. 0.75 miles from the lane closure, can serve as a reminder to encourage drivers to merge into the neighboring lane as quickly as possible. Furthermore, if more lanes close, as shown in the 3 lane to 2 lane example compared to the 3 lane to 1 lane, it is advisable to place signs even further ahead of the lane closure than if only one lane closes.

References

[1] Deffenbacher, J.L., Deffenbacher, D.M., Lynch, R.S., Richards, T.L. (2003). Anger, aggression and risky behavior: A comparison of high and low anger drivers. *Behaviour Research and Therapy,* 41(6), 701-718.

[2] AAA Foundation. “Road Rage” On the Rise, AAA Foundation Reports.
https://www.aaafoundation.org/sites/default/files/roadragePR.pdf

[3] Rawls, John. *A Theory of Justice.* Revised ed. Cambridge: Harvard University Press, 1999.

[4] Stanford Encyclopedia of Philosophy. *Rule Consequentialism.* 9 January 2008.

6 Appendix

6.1 Computational Model Code: Netlogo

```
turtles-own [

before-sign ;; beginning x coordinate of the car in the right lane before the merge sign
decide ;; for the “uniform” module
speed ;; the current speed of the car
lane ;; the current lane of the car
target-lane ;; the desired lane of the car
]

; each patch corresponds to 15 feet long

to setup
clear-all
draw-road
reset-ticks
set-default-shape turtles “car”
crt number [ setup-cars ] ;; creates cars based on number slider from user input
if sign = “close” [
ask patch (max-pxcor - 33.33) -5 [ set pcolor white ] ;; sign about 500 feet ahead of the end
]
if sign = “half-mile” [
ask patch (max-pxcor - 176) -5 [ set pcolor white ] ;; sign about half a mile ahead of the end
]
if sign = “mile” [
ask patch (max-pxcor - 352) -5 [ set pcolor white ] ;; sign about a mile ahead of the end
]
end

to draw-road
ask patches [
set pcolor green
if ((pycor ¿ -4) and (pycor ¡ 0)) [ set pcolor gray ] ;; road
if ((pycor = 0) or (pycor = -4)) [ set pcolor black ] ;; side of road
if ((pycor ¡ -2) and (pycor ¿ -5) and (pxcor ¿ max-pxcor * 0.9)) [ set pcolor orange] ;; blockage/end of lane starts at 0.9 of the distance of the user-set world
]
end

to setup-cars
set color black
set lane (random 3) ;; pick lane randomly
set target-lane lane
if (lane = 0) [ ;; if in bottom/right lane, create cars in front of sign
if sign = “close” [
set before-sign random-xcor
while [ before-sign ¿ max-pxcor - 33.33 ] [
set before-sign random-xcor
]
setxy before-sign -3
]
if sign = “half-mile” [
set before-sign random-xcor
while [ before-sign ¿ max-pxcor - 176 ] [
set before-sign random-xcor
]
setxy before-sign -3
]
if sign = “mile” [
set before-sign random-xcor
while [ before-sign ¿ max-pxcor - 352 ] [
set before-sign random-xcor
]
setxy before-sign -3
]
]
if (lane = 1) [ ;; if in middle lane, distribute cars randomly
setxy random-xcor -2
]
if (lane = 2) [ ;; if in upper/leftmost lane, distribute cars randomly
setxy random-xcor -1
]
set heading 90
set speed 0.1 + random 9.9 ;; set random speed, max is 10 patches/tick which is 102.27 mph
if case = “uniform” and sign = “close” [ ;; if uniform distribution of cars moving module has been chosen, patch will be assigned at which the car begins to try to merge
set decide max-pxcor - 33.33 + random (max-pxcor * .75 - (max-pxcor - 33.33)) ;; make ’decide’ patch a random patch in the range between seeing the sign and the end
]
if case = “uniform” and sign = “half-mile” [
set decide max-pxcor - 176 + random (max-pxcor * .75 - (max-pxcor - 176)) ;; range between seeing the sign and the end
]
if case = “uniform” and sign = “close” [
set decide max-pxcor - 352 + random (max-pxcor * .75 - (max-pxcor - 352)) ;; range between seeing the sign and the end
]
```

```
loop [
ifelse any? other turtles-here [ fd -1 ] [ stop ] ;; make sure no two cars are on the same patch
]
end

to drive
ifelse any? turtles [ ;; only 'goes' if there are turtles/cars present/on the road
ask turtles [
if case = "immediate" and [ycor] of self != -2 [ ;; if immediate module then change lanes
when it sees the sign
if sign = "close" [
if [xcor] of self ¿ max-pxcor - 33.33 [change-lanes] ]
if sign = "half-mile" [
if [xcor] of self ¿ max-pxcor - 176 [change-lanes] ]
if sign = "mile" [
if [xcor] of self ¿ max-pxcor - 352 [change-lanes] ]
]
if case = "wait" and [xcor] of self ¿ max-pxcor * 0.75 - 3 and [ycor] of self != -2 [change-lanes]
;; if wait module then change lanes near the end of the road
if case = "uniform" and [ycor] of self = -3 [
if [xcor] of self ¿ decide [ change-lanes ] ; if uniform module, change lanes when current
position has just passed the patch which has been randomly assigned to start merging at
]
]
;; look one ahead and see if someone is ahead, if so, match speed and decelerate to allow for
space between cars otherwise accelerate
;; if not, check if allowed to see 2 ahead otherwise accelerate
ask turtles [
ifelse (any? turtles-at 1 0) [ ;; each car checks to see if there is a car in front of it - if so, sets
own speed to front car's speed and decelerate
set speed ([speed] of (one-of (turtles-at 1 0)))
decelerate
]
;; if not car immediately ahead but a little further
ahead, match speed of that car and decelerate to allow for space beteen cars
ifelse (look-ahead = 2) [
ifelse (any? turtles-at 2 0) [
set speed ([speed
of (one-of turtles-at 2 0))
decelerate
]
accelerate ;; if no cars 2 in front then accelerate


] [
accelerate ;; if no cars 1 in front then accelerate
] ]
if (speed ¿ speed-limit * 0.0978) [ set speed speed-limit * 0.0978 ] ;; abides by speed limit -
0.0978 is conversion factor from user inputted mph to patches/tick
]
; Now that all speeds are adjusted, give turtles a chance to change lanes
ask turtles [
;; control for making sure no one crashes
ifelse (any? turtles-at 1 0) and (xcor != min-pxcor - .5) [ ;; accounts for back of car
set speed [speed] of (one-of turtles-at 1 0)
]
ifelse ((any? turtles-at 2 0) and (speed ¿ 1.0)) [
set speed ([speed
of (one-of turtles-at 2 0))
]
ifelse ([ycor
of self = -3 and [xcor] of self + speed ¿ max-pxcor * 0.9) ;; if the x coordinate is going to be
past the block then start stopping
begin-stop

fd speed
;; otherise continue as normal
]
]
]
;; delete turtles that have cleared off the middle or top/leftmost lane that everyone has
merged into
ask turtles [
if [xcor] of self ¿ max-pxcor - 1 and [ycor] of self = -1 [die]
if [xcor] of self ¿ max-pxcor - 1 and [ycor] of self = -2 [die]
]
tick
]
stop

end

;; increase speed of cars
to accelerate
set speed (speed + (speed-up * 5280 / (15 * 3600))) ; converts mph user input to patch/tick
end

;; reduce speed of cars
to decelerate
set speed (speed - (slow-down * 5280 / (15 * 3600))) ; converts mph user input to patch/tick
if (speed ¡ 0.01) [ set speed 0.01 ]
```

```
end

;; start stopping completely
to begin-stop
while [ speed ¿ 0 ] [
set speed (speed - (slow-down * 5280 / ( 15 * 3600)))
]
end

to change-lanes
;; at the beginning all target lanes are the same as current lanes
if any? turtles [
ifelse (target-lane = lane) [
if (target-lane = 0) [ ;; if bottom lane then wants to go to middle lane
set target-lane 1
] ;; otherwise do nothing
]
;; if where the car wants to go (target lane) is not the same as where it is then change lanes

;; if the target lane is 1 and it is not in lane 1 and there is not a turtle above it and there is a turtle in front of it then decelerate
ifelse (target-lane = 1) [
ifelse (pycor = -2) [ ;; if target lane is 1 but is already in lane 1 according to pycor then set lane as 1, causing to go through above loop instead
set lane 1


ifelse (not any? turtles-at 0 1) [ ;; if target lane is 1 and is not in lane 1 according to pycor then check if there is a car next to it in the lane it wants to move into
set ycor (ycor + 1) ;; if no cars in desired space then move up


ifelse (not any? turtles-at 1 0) [ ;; if cars in desired space in desired lane then move forward according to the folloing
ifelse xcor + speed ¡ (max-pxcor * 0.95) [ ;; if will not move off the end of the lane in the next step then move forward
fd speed

decelerate
;; if will move off the end of the lane in the next step then decelerate instead of maintaining current speed
]
decelerate ;; if there are cars right in front of it then decelerate
] ] ] [ ] ] ]
end
```