ADVANCED DRIVER ASSISTANCE SYSTEM CAR FOLLOWING MODEL OPTIMIZATION FRAMEWORK USING GENETIC ALGORITHM IMPLEMENTED IN SUMO TRAFFIC SIMULATION

by

Matthew Carroll

May, 2022

Director of Thesis: Rui Wu, PhD

Major Department: Computer Science

Abstract

As advanced driver-assistance systems (ADAS) such as smart cruise control and lane keeping have become common technologies, self-driving above SAE level 3 are being competitively developed by major automobile manufacturers, autonomous vehicles (AVs) will prevail in the near future traffic network. In particular, evasive action algorithms with collision detection by sensors and faster braking response will enable AVs to drive with a shorter gap at higher speeds which has not been possible with human drivers. Such technologies will be able to improve current traffic performance as long as raising concerns on safety are addressed. Therefore, there have been efforts to improve understanding between stakeholders such as regulatory authorities and developers to draw a consensus about autonomous driving standard and regulations. Meanwhile, a mixed traffic network with human driving vehicles and AVs will show transient system behavior based on penetration rate of AVs thereby requiring different optimal AV settings. We are interested in understanding this system behavior over transitional period to achieve an optimal traffic performance with safety as a hard constraint. We investigate the system behavior with agent-based simulation with different penetration rates by mixing of human-driving and AV vehicle models, identify the key parameters of ADAS algorithms for traffic flow, and find the optimal parameter set per penetration rate by using genetic algorithm (GA). Simulation results with optimal parameter values reveal improvement in average traffic performance measures such as flow (5.6% increase), speed (4.9% increase), density (15.9% decrease), and waiting time (48.2% decrease). We provide simulation examples and discuss the implication of the optimal parameter values for both traffic control authorities and AV developers during the transitional period.

ADVANCED DRIVER ASSISTANCE SYSTEM CAR FOLLOWING MODEL OPTIMIZATION FRAMEWORK USING GENETIC ALGORITHM IMPLEMENTED IN SUMO TRAFFIC SIMULATION

A Thesis

Presented to The Faculty of the Department of Computer Science

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Science in Software Engineering

by

Matthew Carroll

May, 2022

ADVANCED DRIVER ASSISTANCE SYSTEM CAR FOLLOWING MODEL OPTIMIZATION FRAMEWORK USING GENETIC ALGORITHM IMPLEMENTED IN SUMO TRAFFIC SIMULATION

by

Matthew Carroll

APPROVED BY:

DIRECTOR OF THESIS: ______________________________
Rui Wu, PhD

COMMITTEE MEMBER: ______________________________
Jinkun Lee, PhD

COMMITTEE MEMBER: ______________________________
Qin Ding, PhD

COMMITTEE MEMBER: ______________________________
Zhen Zhu, PhD

CHAIR OF THE DEPARTMENT ______________________________
OF COMPUTER SCIENCE: Venkat Gudivada, PhD

DEAN OF THE ______________________________
GRADUATE SCHOOL: Paul J. Gemperline, PhD

ACKNOWLEDGEMENTS

I would like to express my gratitude toward my advisors, Dr. Jinkun Lee and Dr. Rui Wu, for their continuous support toward my M.S. studies and research. They gave consistent motivation and enthusiasm where their guidance helped me throughout the process of designing, developing, and implementing this study. I appreciate all their knowledge they have passed on to me.

Besides my advisors, I give my thanks to my other committee members, Dr. Qin Ding and Dr. Zhen Zhu, for their interest and insightful feedback pertaining to this study.

Finally, I acknowledge and appreciate Coleman Ferrell for his work developing decision trees and determining parameter impacts on performance measure which are included in and used to improve this study.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Major vehicle manufacturers are shifting towards producing autonomous capable electrical vehicles. There are concerns in how self driving vehicles will effect traffic and how safe these vehicles really are. The increase in self-driving cars on our roads has resulted in safety concerns with the possibility of accidents created by them and the possibility of negative effects on overall traffic performance. It is important to ensure the public's safety and provide assurance that self-driving cars will not cause potential danger. It is essential to be able to take preemptive actions in simulating traffic networks and agents accurately so that these situations can be investigated before incidents occur. While doing this, we take into consideration how traffic performance can be improved for different transitional periods. Previous papers have investigated similar approaches to improving ADAS in a connect vehicle environment [6]. This study investigated the use of GA to improve the traffic performance for freeway traffic simulations [6]. We propose a framework for an urban road network that can implement other vehicle behavior models. In our framework, multiply performance measures are incorporated and investigated to determine the correlation between them. Included in this framework is the alteration of the ratio of Intelligent Driver Model (IDM) to Krauss vehicles which represents the market penetration rate of ADAS vehicles. As the penetration rate of self-driving capable vehicles increases, the replacement of conventional vehicles by these vehicles is predicted to result in changes to the traffic system. This transition will be made gradually by the increased availability of affordable autonomous vehicles on the market. Therefore, we are interested in developing microscopic agent-based traffic network simulation models with different traffic mix ratios of conventional and autonomous vehicles. There are traffic simulation platforms available to create road network models and implement traffic demand to analyze traffic performance. Some examples of software available is VISSIM, Simmobility, and Simulation of Urban MObility (SUMO). These traffic simulation software platforms are capable of creating traffic networks that represent real world networks and are capable of introducing different vehicle models that represent a variety of vehicle behaviors. Since we look to develop a framework for analyzing and improving mixed traffic performance, the software is simply a tool to do so as long as the software is capable of producing such simulations. We chose SUMO due to its proven scholarly academic use and since it is an open source software available to the public. SUMO is a traffic simulation platform that can incorporate millions of agents to investigate various traffic network models [7]. In doing so, we aim to analyze the efficacy of our proposed mixed traffic scenarios. We propose methods to improve self driving vehicle behavior performance by developing a framework to alter behavior model parameters then implement methods of measuring performance. We use the framework to analyze penetration rates of self-driving vehicle models into a traffic network of conventional vehicles behavior models. Using the proposed framework, we will be able to predict how the replacement of conventional vehicles will transform the traffic network over time. The insight learned will be useful for better decision making in traffic network control and the development of self-driving vehicles embedded control systems.

The major contributions of this thesis are:

- This paper provides a framework that allows us to alter car following behavior in a simulated environment to improve traffic performance measures including density, waiting time, flow, and average speed.
- This framework can be applied to other self-driving vehicle behavior models to provide a means to verify that the behavior meets safety constraints and limitations while improving the vehicle models performance within a traffic system.

The rest of the thesis provides sections describing our proposed methodology, results analysis and discussion, conclusions, and future work.

Chapter 2

Methodology

Using statistical and computer science methods, we look to deliver a framework to analyze and improve traffic performance while considering constraints and limitations of self-driving vehicles. We use SUMO traffic simulation platform to simulate traffic networks since it allows for different vehicle models to be implemented into traffic networks modeled from physical roads [7]. Sumo provides different vehicle models for car following and lane changing. The vehicle behaviors of interest in this study are the car-following behavior models. We use Krauss as the conventional driver [5], and the IDM [11] as the self-driving vehicle. Python is used to interact with SUMO to alter input parameters for IDM vehicle types, alter the ratio of IDM to Krauss models, run simulations, and collect output performance measurements. With the use of python, SUMO's XML configuration files can be accessed allowing for behavioral model parameter modification and performance data collection. In the next sections we explain SUMO capabilities and how it was used to alter car following models, generate a road network, and apply traffic demand.

2.1 Vehicle Behavior Modeling

Vehicle behavior modeling allows for realistic vehicle behavior to be implemented into simulated traffic scenarios to better understand how traffic network changes will effect the performance of the network. It is important to understand vehicle behavior models since they determine how simulated vehicles interact with each other within a network. Since this study involves the analysis of autonomous vehicle behavior for different penetration rates onto conventional roads, we implement both the Krauss model [5] and the Intelligent Driver Model [11]. Each of the models have behavioral parameters that can be altered to change the behavior of the vehicle. In the physical world, AV behavior is determined by vehicle/hardware limitations, safety regulations, and how the vehicle behavior model is developed. NIST.SP.1900-301 proposes safety measures based on the design of ADAS vehicles for environmental factors [3]. We look at how the behavioral models can be altered. Since it is possible to alter autonomous vehicle behavior within these constraints, it is vital to be able to understand how parameter changes for IDM vehicles effect the interactions within a network and how they can be used to improve performance. Through the use of simulation, we gain a better understanding of vehicle constraints and are provided methods to improve efficiency of traffic networks. The next sections discusses the car following models, IDM and Krauss, and gives a brief description of lane changing models provided by SUMO.

2.1.1 Krauss Car-Following Model

The Krauss model [5] was developed based on addressing the assumptions made on general properties of traffic flow that dictates the individual behavior of drivers within a traffic network. Two types of vehicle motion are addressed where the first scenario being a free flow scenario and the other being the behavior exhibited when a vehicle is interacting with another vehicle [5]. The model is developed with consideration that the main goal is to not collide when vehicles react with each other, otherwise, the model should reach a desired maximum speed, being the speed limit of the road [5]. Equation 1 gives the equation used to determine a safe velocity, $v_{safe}(t)$, for a following vehicle.

$$v_{safe}(t) = v_l(t) + \frac{g(t) - g_{des}(t)}{\tau_b + \tau} \tag{2.1}$$

Where $v_l(t)$ is the velocity of the leading vehicle (m/s), $g(t)$ is the current gap (m), $g_{des}(t)$ is the desired minimum gap (m), τ_b is the time scale (s), and τ is the reaction time for drivers (s), [5]. Equation 2 is used to calculate the desired velocity, $v_{des}(t)$.

$$v_{des}(t) = min[v_{max}, v(t) + a(v)\Delta t, v_{safe}(t)] \tag{2.2}$$

Where $v_{des}(t)$ is the desired velocity of following vehicle (m/s), v_{max} is the maximum velocity the vehicle can travel in the road (m/s), $v(t)$ is the current velocity of the following vehicle (m/s), and $a(v)$ is the maximum acceleration parameter (m/s^2), [5]. Equation 3 is used to ensure that the velocity of the following car does not become negative.

$$v(t + \Delta t) = max[0, v_{des}(t)] \tag{2.3}$$

Equation 4 determines the position for the following vehicle for the next time step, $x(t + \Delta t)$.

$$x(t + \Delta t) = x(t) + v\Delta t \tag{2.4}$$

Given Equations 1 through 4 there are five input parameters that can alter the vehicle behavior for the Krauss model as well as a consideration for emergency deceleration for unexpected conditions that would cause collision [5]. The six input parameters are shown below in Table 2.1.

Table 2.1: Krauss Car Following Model Parameters

Attribute	Default	Range	Description
minGap	2.5	≥ 0	Minimum Gap when standing (m)
accel	2.6	≥ 0	The acceleration ability of vehicles of this type (in m/s^2)
decel	4.5	≥ 0	The deceleration ability of vehicles of this type (in m/s^2)
emergencyDecel	9	$\geq$ decel	The maximum deceleration ability of vehicles of this type in case of emergency (in m/s^2)
sigma	0.5	[0,1]	The driver imperfection (0 denotes perfect driving)
tau	1	≥ 0	The driver's desired (minimum) time headway. Exact interpretation varies by model. For the default model Krauss this is based on the net space between leader back and follower front

For this study, the default parameters for the Krauss model remain the same. The Krauss model is used as a conventional driver model, therefore, this model is used with varying penetration rates of IDM vehicles. The next section will go over the IDM vehicles.

2.1.2 Intelligent Driver Model (IDM)

IDM focuses on single lane car-following behavior [11]. IDM is a car-following model that represents the behavior of an automated system SAE level 4 and is conservatively designed to be collision free [4]. The IDM proposes a model that is simple but reproduces collective dynamics results in a plausible microscopic acceleration and deceleration behavior. The IDM vehicles are able to replicate realistic self-driving behavior with relatively few intuitive parameters [4]. The IDM car-following model can be described in the following equation where Equation 5 describes the $\dot{v}_\alpha$ as the acceleration (m/s^2) of the IDM vehicle.

$$\dot{v}_\alpha = a^{(\alpha)} \left[1 - \left(\frac{v_\alpha}{v_0^{(\alpha)}} \right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right] \tag{2.5}$$

Where $a^{(\alpha)}$ is the maximum acceleration (m/s^2), v_α is the current velocity of the IDM vehicle (m/s), $v_0^{(\alpha)}$ is the velocity the vehicle would drive at in free traffic (m/s), δ is the acceleration exponent, $s^*(v_\alpha, \Delta v_\alpha)$ is the desired gap (m), and s_α is the current gap (m) [11]. Given this equation, the free flow acceleration, $a_f(v_\alpha)$ can be describe using Equation

6 shown below.

$$a_f(v_\alpha) := a^{(\alpha)}\left[1 - \left(\frac{v_\alpha}{v_0^{(\alpha)}}\right)^{\delta}\right] \tag{2.6}$$

Given Equation 5, The tendency to brake with decceleration, $-b_{int}(s_\alpha, v_\alpha, \Delta v_\alpha)$ in (m/s^2), can be described using Equation 7 shown below.

$$-b_{int}(s_\alpha, v_\alpha, \Delta v_\alpha) = a^{(\alpha)}\left[1 - \left(\frac{s^*}{s_\alpha}\right)^{2}\right] \tag{2.7}$$

The desired gap, $s^*(v, \Delta v)$, is calculated with Equation 8 shown below.

$$s^*(v, \Delta v) = s_0^{(\alpha)} + s_1^{(\alpha)}\sqrt{\frac{v}{v_0^{(\alpha)}}} + T^\alpha v + \frac{v\Delta v}{2\sqrt{a^{(\alpha)}b^{(\alpha)}}} \tag{2.8}$$

Where $s_0^{(\alpha)}$ is the minimum gap (m), $s_1^{(\alpha)}$ is the jam distance (m) with a typical value of 0, v is the current vehicle velocity (m/s), T^α is the time headway (s) referred to as ”tau”, Δv is the velocity difference (m/s) or approach rate, and $b^{(\alpha)}$ is the deceleration (m/s^2). The velocity difference is calculates using $\Delta v = v_\alpha - v_{\alpha-1}$, where α dictates the following IDM vehicle and $\alpha - 1$ would represent the leading vehicle [11]. Table 2.2 shows the input parameters for the IDM vehicle model with their defaults, range, and description.

Table 2.2: Intelligent Driver Model (IDM) Car Following Model Parameters

Attribute	Default	Range	Description
minGap	2.5	≥ 0	Minimum Gap when standing (m)
accel	2.6	≥ 0	The acceleration ability of vehicles of this type (in m/s^2)
decel	4.5	≥ 0	The deceleration ability of vehicles of this type (in m/s^2)
emergencyDecel	9	$\geq$ decel	The maximum deceleration ability of vehicles of this type in case of emergency (in m/s^2)
tau	1	≥ 0	The driver's desired (minimum) time headway. Exact interpretation varies by model. For the default model Krauss this is based on the net space between leader back and follower front
delta	4	N/A	acceleration exponent

2.1.3 Lane-Changing Models

The lane-changing model determines lane choice on multi-lane roads and speed adjustments related to lane changing. When simulating traffic networks with multi-lane roads there will be routes that require vehicles to change lanes. Lane changing is a hard determinant when coming to traffic efficiency and is important in developing an accurate behavior model. This study focuses on the car-following behavior models Krauss and IDM, for more information on lane-changing models refer to [1] on the references section of this paper.

2.2 Traffic Network Generation

When creating a traffic simulation, elements that represent a physical road network are needed which include roads, lanes, traffic lights, and stop signs [7]. These elements come together to form the road network. It is very time consuming when developing a simulation scenario based on real world data [7]. In SUMO, the traffic network consists of nodes and edges that represent roads [7]. Edges are geometric line consisting of a series of line segments that represent the lanes on the road. Attributes describe these edges with width, speed limit, and access permissions. Within Sumo, tools are provided that grab open source road network data from OpenStreetMap [8]. OSM Wizard is a tool provided by SUMO and is a preferred choice for importing road data [7]. This allows a user to select a region in OpenStreetMap and import the associate elements within a latitude and longitude environment. In this study, A portion of Greenville, NC was imported which included East Carolina University and its surrounding roads. Figure 2.1 shows the network created by the OSM Wizard.

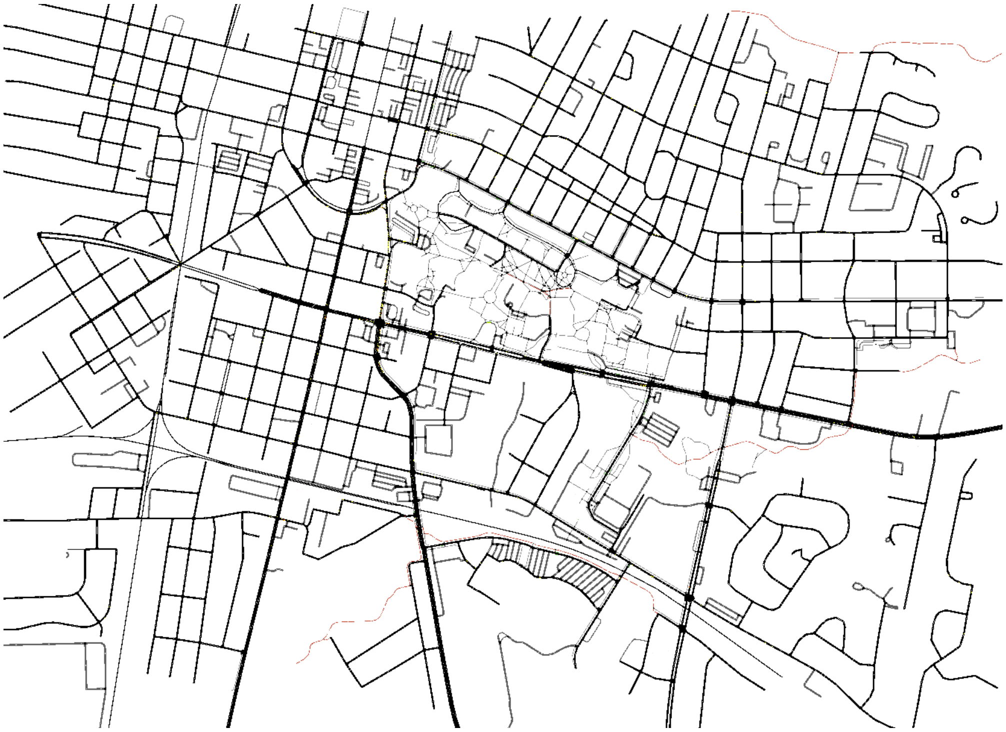


Figure 2.1: Subsection of Greenville, NC Traffic Network in SUMO

The OSM Wizard generated random traffic for the simulation which will be described more in the next section.

2.3 Traffic Demand

The demand for the Greenville, NC traffic network can be generated through tools provided by SUMO [7]. We describe methods used for importing traffic demand and list resources for validating and calibrating demand of a traffic simulation to match the traffic system being represented.

2.3.1 Demand Generation

Sumo provides different methods of defining traffic demand through either trips, flows, or routes [7]. Information that is needed for these elements are departure time, origin, destination, and transport mode. Transport mode can be defined as either a pedestrian or as a vehicle. Routes does not require information about origin and destination but instead uses a sequences of edges to define the path taken by the vehicle [7]. There are different tools provided by SUMO so generate the traffic demand. For our purposes, the demand was generated when importing the network from OSM Wizard [7]. If demand is selected from the OSM Wizard tool the demand is created using the randomTrips.py file provided by SUMO [7]. The trips are created based on the roads imported but are not an accurate representation of the actual demand of the roads. To further improve this model, traffic data of the physical traffic flow is desired to generate an accurate demand that represents the real world traffic demand [2]. Figure 2.2 shows the traffic network that has demand created with the randomTrips.py program where the colors represent the average speed of vehicles on an edge.

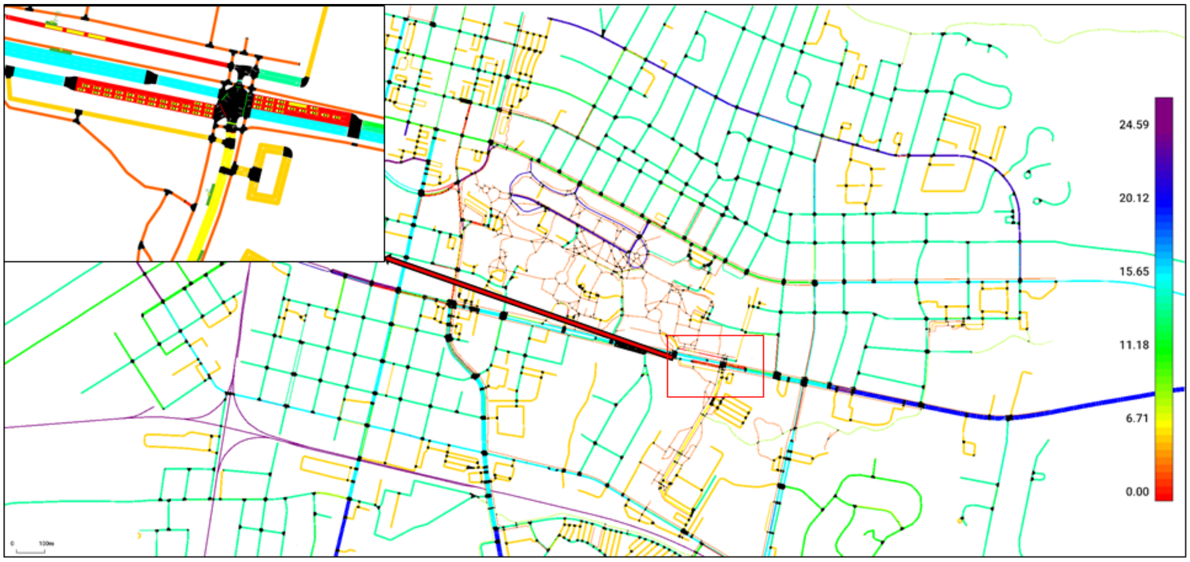


Figure 2.2: Greenville, NC Traffic Network Average Speed (m/s)

Figure 4 shows congestion at an intersection resulting in the average speed being reduced. The legend of the map is in m/s and is provide in the SUMO GUI [7]. The next section will discuss the performance measurements that are used to analyze traffic efficiency.

2.4 Performance Measure - Microscopic

The traffic network performance can be quantified through microscopic performance measurements that describe the traffic conditions and efficiency. SUMO produces nine performance measurements that include density, sampled seconds, waiting time, occupancy, time loss, speed, entered, and collisions [7]. Flow is a tenth performance measure calculated by dividing the entered performance measure sampling time [7]. This study looks at considering density, waiting time, flow, speed, and collisions as the primary performance measurements. Density is the amount of vehicles within a given space measured in vehicles per kilometer per edge or lane, waiting time is the average time vehicles spend idle, flow is the rate at which vehicle move through the network measured by vehicles per second per edge/lane, speed is the average velocity of vehicles measured in meters per second, and collisions is when vehicle come in contact with one another measured as the number of collisions that occurs during the simulation [7]. Collisions is used as a hard constraint since any collisions within the simulation is considered a nonviable solution. Figure 2.3 shows the relationship between density and flow within the traffic model for different penetration rates of IDM vehicles.

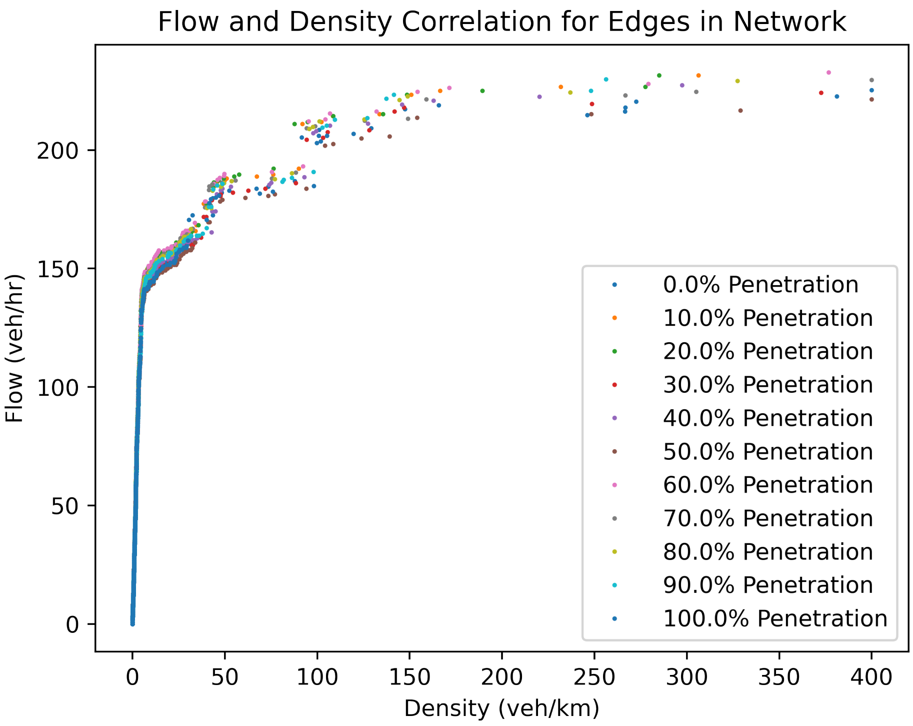


Figure 2.3: Flow and Density for Edges in Network Depicting a Positive Correlation Between The Two Measurements

Given Figure 2.3, we can see that on free flow roads with lower densities and flows, the changes are not noticeable. Based on the simulation results, a positive correlation exists between density and flow. However, there are significant changes in density and flow on high demand roads. Given the results, we can determine that mixed traffic flow scenarios have an impact on performance of congested roads and the vehicle behavior should be adjusted according to penetration rates of IDM vehicles onto the traffic network.

2.5 IDM Behavior Improvement Framework

The framework developed is a methodology to improve the performance of self-driving vehicles while maintaining safe driving environments for drivers. Our methodology uses iterative traffic simulation and statistical methods to analyze and determine behavioral model parameters to improve traffic performance. To transition this framework to real world applications, modern self-driving behavioral modeling, traffic flow validation, and traffic network validation must be implemented when developing a traffic simulation environment to ensure the viability of the results. This is discussed further highlighting possible concerns and the processes that should be used to solve these issues.

2.5.1 Framework Structure

The theoretical methodology of developing the framework comes from the following requirements that must be met to ensure an accurate and reliable result. The following requirements can be used to ensure quality results when implementing the framework.

Feasibility and Limitation Analysis - Method to determine the infeasible regions and limitations of each behavior model parameter. The purpose of this is to remove any lower or higher parameter regions that would result in collisions, cause safety concerns, are physically impractical, and would cause an uncomfortable driving experience for users. Standards of the industry can be applied as well to guarantee a reliable outcome. This will help to guarantee a valid model and that requirements are met to provide realistic behavior modeling.

Individual Parameter Optimization - This phase involves running separate algorithms designed to improve the performance or fitness of an objective function. In this case, the objectives are to increase flow and average speed while reducing density and waiting times for vehicles. Each parameter is individually improved where the relationships between the performance measures can be investigated.

Performance Measure Standardization - Given the data sets from previous individual parameter optimization, the data-sets can be used to create a distribution of possible performance measure values. This distribution can then be used to standardize each performance measurement where they can be combined. This is crucial since the performance measurements should be balanced when combined in a fitness function. Weights can then be used to increase the importance of specific performance measures of interest.

Combined Performance Measure Optimization - Given the feasible parameter regions and the standardization of performance measures, an optimization method can be applied to produce a weighted result where multiple performance measures are considered. The result will be considered a final solution for improved behavioral model parameter selection.

Behavior Model Validation - Once a final solution is developed, The behavioral model must be validated to ensure the driving behavior is realistic and comfortable for a human drive. This process involves determining the extreme conditions that the behavioral model may experience as well as the model's observed behavior at the time of these occurrences. The transient states between one situation to another are also of interest to determine that conditional state changes and how the vehicle will react given abrupt environmental changes.

2.5.2 Framework Implementation

The implementation of the given framework involves applying statistical methods to analyze performance measures returned from simulation instances. The methods are used within this framework structure are one at a time (OAT) sensitivity analysis, GA, and z-score standardization. OAT sensitivity analysis is used to determine parameter impact on performance measures and where safety is compromised. GA was used as the optimization method to improve the model. This method allows us to alter behavioral model parameters through crossover and mutation where a fitness function is then implemented to quantify the performance where lower is better. We apply z-score standardization to each performance measure and combining them giving a balanced fitness model for overall performance analysis. The fitness function was used in GA to give a balanced solution that considers multiple performance criteria. After a final solution is developed, the model is validated by plotting the behavioral model based on the inputs for current and desired velocity. The given implementation is explained in greater detail in the analysis and discussion section.

2.5.3 Translating Methodology to Commercial Vehicles

Given the framework developed, it is important to discuss the transition necessary for this framework to be applied in the physical world. Below we discuss the main concerns and solutions for implementing this framework to help developers utilize the framework to improve their own models.

Creating an Accurate Road Network - Replicating an traffic network accurately into a simulation environment is important since the the network attributes must match real world road networks to be able to find viable solutions for improvement. SUMO provides means to import a traffic network based on OpenStreetMap data. Since this is an open resource and is updated by the community, there may be discrepancies and outdated network features. These variations in network features may create issues and point researchers in the incorrect direction for altering existing roads. Studies have shown that implementing deep learning methods on aerial and satellite images for road extraction has outperformed other methods for road data extraction [10]. These methods have proven themselves in developing traffic network simulations with accuracy of 98.83% [10].

How to Import Manufacturers Driving Models - In typical self-driving vehicles, deep learning and neural network models are used to train the driving behavior of the vehicles [9]. It is common that data-sets are collected from vehicles where the models are developed by using hyper-parameters and altering the data-set till a model's results meet an expected accuracy. Since this differs from the traditional equation model for IDM vehicles, we can alter how we use GA's where hyper-parameters and k-folds are used to reach a specified accuracy while also improving the performance measures of a simulated system. Comma is a company that provides ADAS systems that can be integrated into modern vehicles [9]. Comma provides an open source software for training a neural network to provide automated driving assistance [9]. Although this is not a product meant for fully autonomous driving, it does give an example of how behavior modeling of self-driving vehicles is performed and provides proof that this modeling is common within the industry. Given these models, it is possible to implement them into simulation environments. SUMO allows for new car-following and lane changing models to be imported into the SUMO infrastructure. A manufacturers behavioral modeling approach can be integrated into SUMO where parameters relating to the model can be defined similar to the preexisting IDM model. With this approach, we can take a desired self-driving model and implement them in a simulation environment.

Understanding and Restrictions and Limitations - Restrictions for self-driving vehicles mainly concerns that ADAS equipped vehicles can reliable handle critical driving situations. NIST.SP.1900-301 [3] gives specifications for self-driving vehicles to assess safety of the vehicle control system. They provide descriptions of environmental factors, referred to as an Operating Envelope Specification (OES), the vehicles may encounter and relates them to the Operational Design Domain (ODD) [3]. How the vehicles are able to interact with their environment is an important factor in determining a driving behavior that assures the safety of passengers and the public. The specifications propose a NIST Framework for Cyber-Physical Systems (CPS Framework) that identifies and classifies constraints on system behavior. Other Limitations to consider for behavioral models are the capabilities of a given vehicle, and the behavior that is deemed comfortable to passengers. Constraints developed from restrictions and limitations should be integrated into the framework to ensure the solution developed is a viable behavioral model.

Verifying Traffic Simulation Demand using Measured Data - Traffic demand within a simulation should be equivalent to measured data taken from traffic networks of interest. Traffic demand calibration methods can be used to alter and verify that a simulations demand is correctly depicting the flow of a real world network [2]. Along with this method of demand validation, there are other approaches that can be taken to verify that the demand in the simulation matches the physical traffic network. These methods can be implemented into SUMO traffic simulations to ensure that the demand is being represented correctly so that results of a implementing the framework are reliable and applicable to real traffic scenarios in the network of interest.

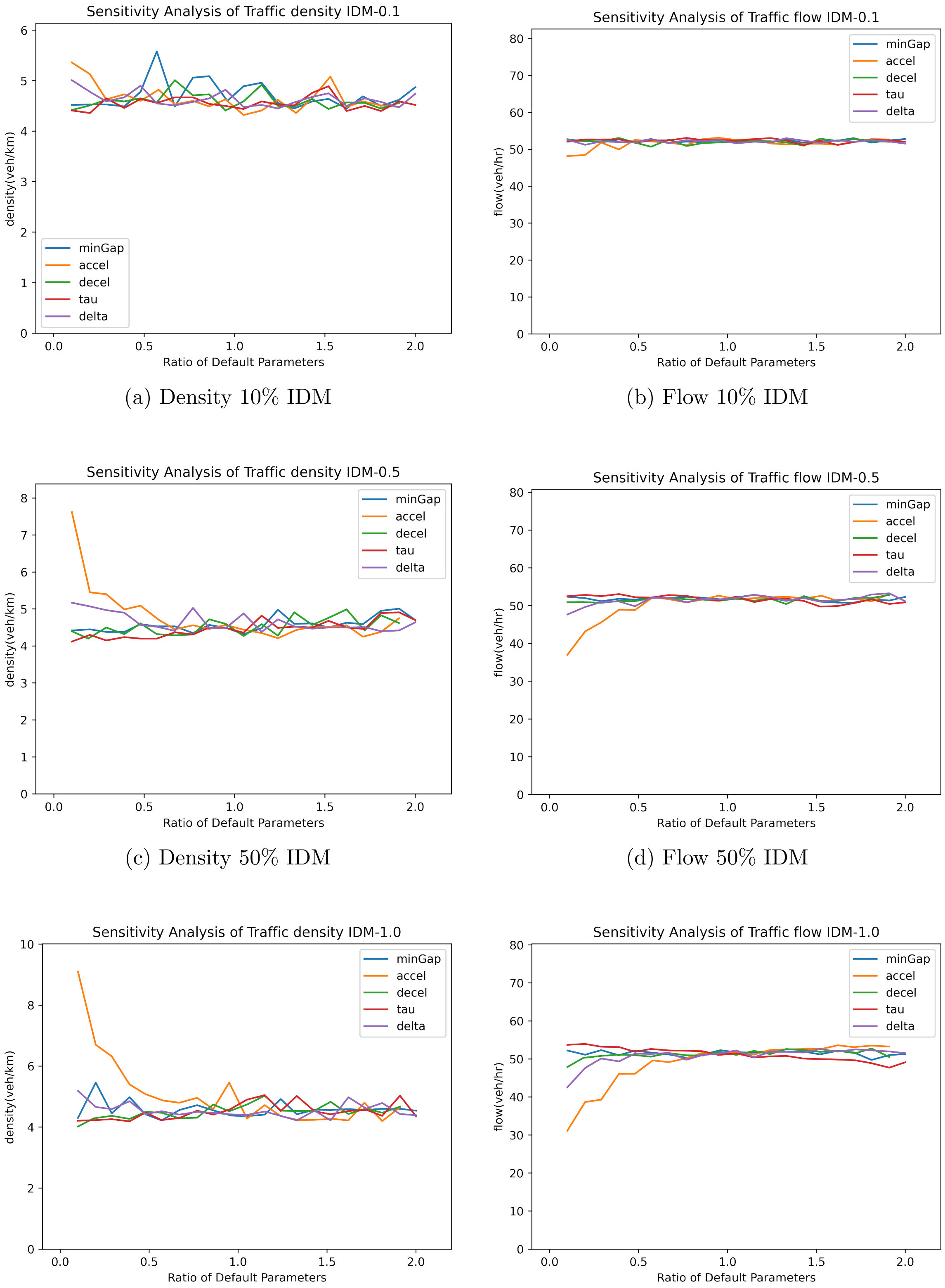
Chapter 3

Analysis and Discussion

This study investigates the effects of modifying autonomous vehicle car following behavior. SUMO is used to simulate different penetration rates of autonomous vehicles onto current network using SUMO's IDM and Krauss models. Krauss models remain unaltered while IDM model parameters are changed. The penetration rate describes the amount of IDM vehicles to the Krauss models vehicles within the network. Sensitivity analysis is used to show the impact parameters have on performance measures. Regression tree analysis uses data-sets of simulation results to show relationships between IDM parameters and the performance parameters. GA is used to find an optimal solution that increase the traffic performance while maintaining safety. The results for each analysis method are discussed in the following sections.

3.1 Sensitivity Analysis

One at a Time (OAT) Sensitivity Analysis is applied to the Greenville traffic simulation to determine the importance of individual car following model parameters and the infeasible regions that will cause collisions. The analyzed parameters include minGap, accel, decel, tau, and delta [7]. Emergency deceleration was not considered since it is a conditional parameter that is only applied when specific conditions are met, therefore it is not important for the model in the sense it does not directly effect the car following model performance. By doing this, the aim is to better understand the impact of IDM model parameters and predict how the parameter changes may impact density, waiting time, flow, speed, and collisions. The simulation was run for 20 iterations spaced evenly where parameter values were changed from 10% to 200% of the default parameter set by SUMO. This procedure was repeated for ten penetration rates of 10% increments between 0% and 100% where the penetration is given as a percentage of IDM to Krauss vehicles. The effects from the penetration rate can be seen in the figures below were we show the penetration rates for 10%(0.1), 50%(0.5), and 100%(1.0) to illustrate the differences in the mixed traffic flow networks. Shown on the next page in Figure 3.1 are the results for sensitivity analysis of traffic performance of flow and density with given penetration rates of 10%, 50%, and 100% of IDM vehicles.



(a) Density 10% IDM

(b) Flow 10% IDM

(c) Density 50% IDM

(d) Flow 50% IDM

(e) Density 100% IDM

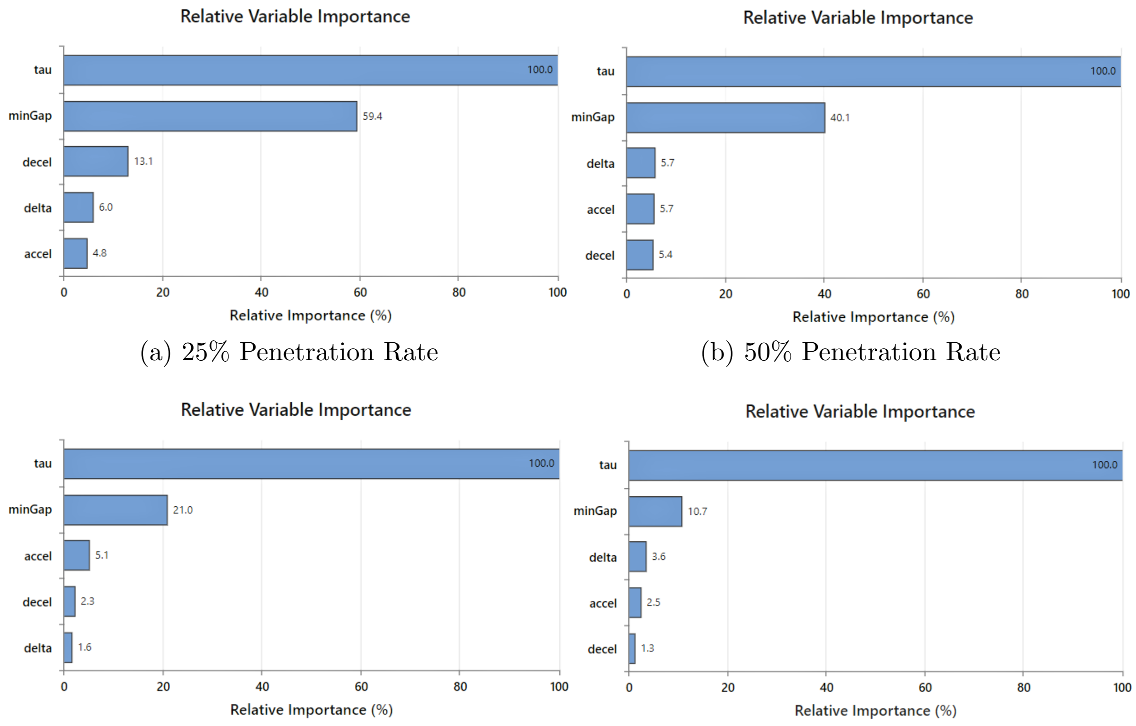
(f) Flow 100% IDM

Figure 3.1: Sensitivity Analysis of Traffic Performance Measures for Flow and Density at Varying Penetration Rates of IDM Vehicles Showing Increased Impact of Parameter Changes at Higher Penetration Rates

The results from Figure 3.1 show us that there is variance in flow and density when the IDM parameters are changes. From a, c, and e of Figure 3.1 we can see density changes depending on the parameter being altered. We see that minimum gap and acceleration had the most significant changes in density. The deceleration had moderate effects to the density and finally, tau and delta have less impact on density. From b, d, and f of Figure 3.1 we can see a steadier change of flow in correlation to changes in parameter values. As the penetration rate increases, the parameters had a negative effect on the flow of the system. For both flow and density it can be seen that as the penetration rate of IDM vehicles increase, we see more of an impact when parameters are altered. It is also noted that each parameter impact changes depending on the penetration rate and that the parameter changes do not guarantee that the optimal parameter set created from GA at one penetration rate will give the same optimal parameters at other penetration rates. In the next section the method of regression tree analysis is performed on a data set created from the GA ran on sumo simulations of varying penetrations of IDM vehicles.

3.2 Regression Tree Analysis

We use Classification and Regression Tree (CART) tool in Minitab to analyze simulation results from random parameter values for different penetration rates (25%, 50%, 75%, and 100%) to identify boundary limits of several parameters. The benefits of parameter boundaries are two-fold: (i) ensuring hard-constraint such as zero collision by setting a threshold for a related parameter and (ii) reducing parameter regions to improve GA efficiency. A CART analysis involves organizing data into similar groups based on correlations between inputs and outputs. Tree-based machine learning assists in constructing a tree diagram to help better visualize the grouping. Multiple trees are produced with different number of nodes, but only the tree diagram with the maximum R-squared value is considered since this tree breaks down the groups completely. Specific inputs have more significant effects on the classification of groups, thus they have more influence on the resulting output. The relative importance of the input parameters helps in identifying which parameters and at what value lead to a distinct outcome. First, we consider all the parameters' relative importance on collision performance and traffic performance. Figure 3.2 displays the relative importance graphs created for the collision performance.



(a) 25% Penetration Rate

(b) 50% Penetration Rate

(c) 75% Penetration Rate

(d) 100% Penetration Rate

Figure 3.2: Relative Parameter Importance for Collision Performance Measure

The parameter tau is found to be the most important decision basis when classifying the groups by collisions, while the accel parameter is the most important when classifying by the chosen performance-related outputs. Hence, decision trees are generated using tau and accel as the primary determiners. An example of the decision tree for the flow output at the 75% penetration rate is displayed in Figure 3.3. When looking at the terminal nodes, or groups, terminal node 7 has the highest average flow of vehicles. As a result, a solution will only be considered defensible if the accel value at 75% penetration rate is greater than 5.135 m/s^2. A summary of the boundaries for all the decision trees is compiled in Table 3.1. These findings refine the viable parameter region and narrow the search in the GA, thereby allowing the GA to run more efficiently. Additionally, the boundaries validate the correctness of the GA and the accuracy of the results.

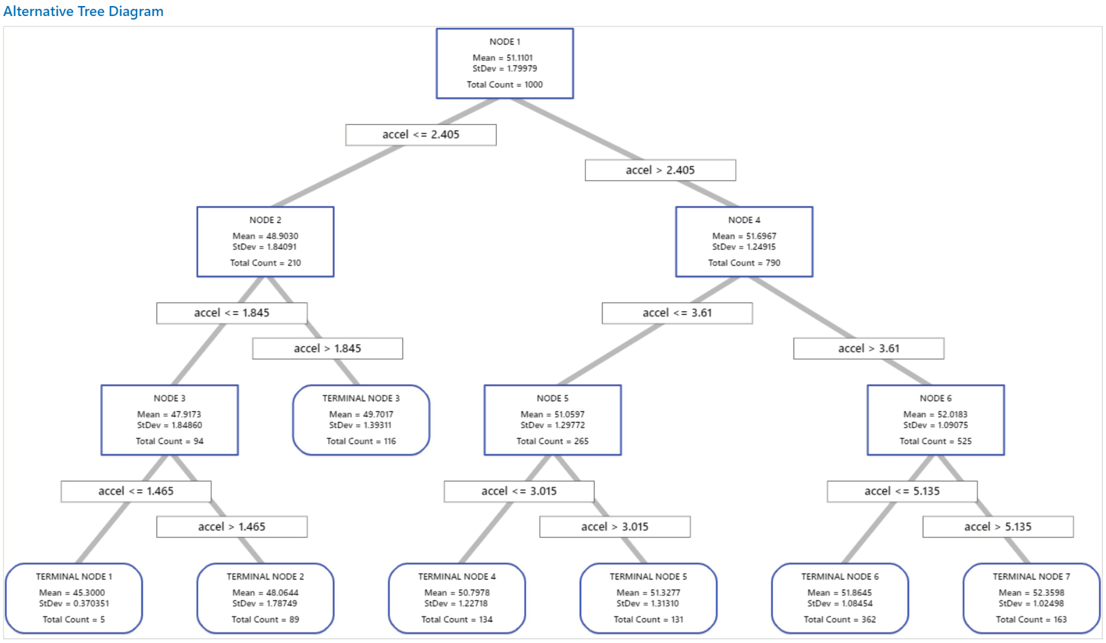


Figure 3.3: Decision Tree for Flow Output at 75% Penetration Rate

Table 3.1: Optimal Decision Tree Boundaries for Each Measured Output

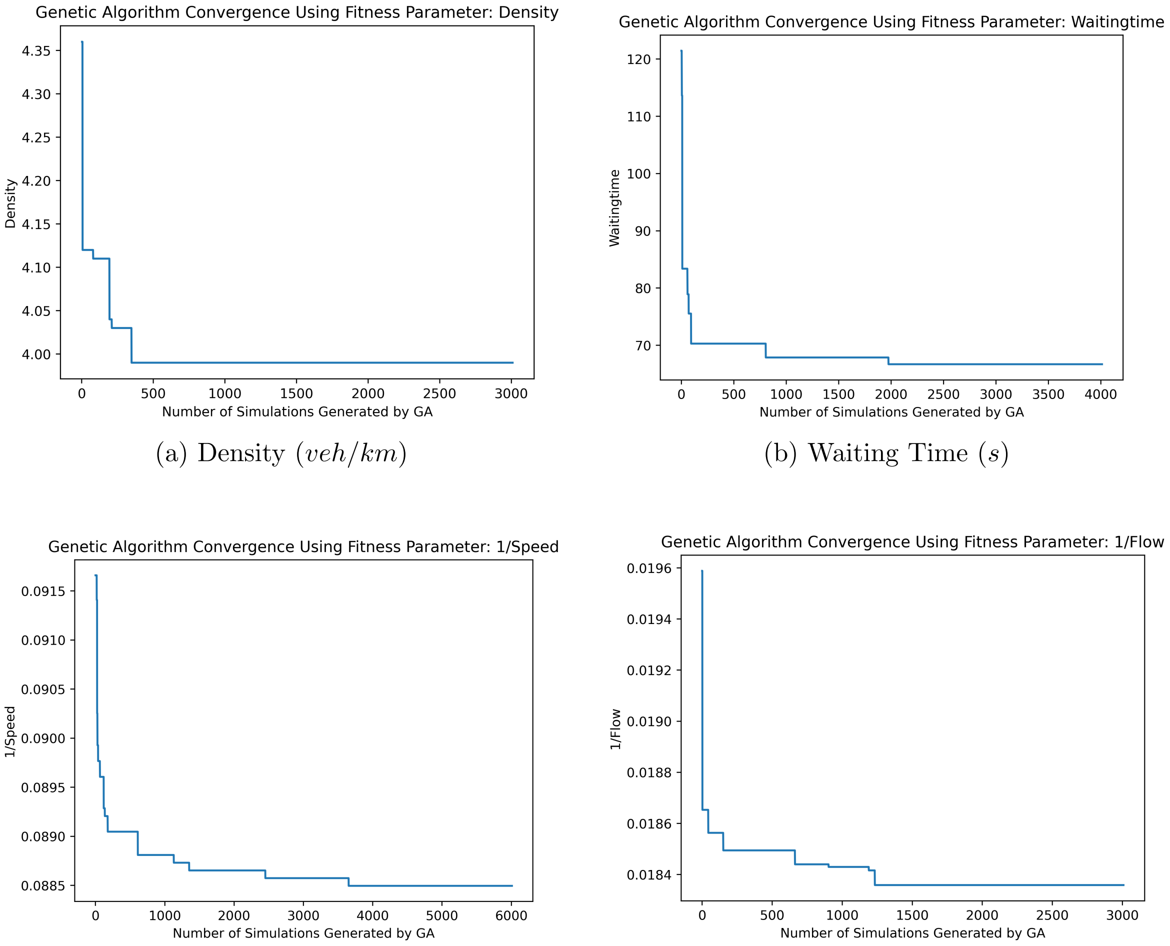
	Collisions	Density	Waiting Time	Flow	Speed
Penetration Rate	tau	accel	accel	accel	accel
25%	> 0.795	> 2.595	> 1.785	> 5.675	> 4.865
50%	> 0.785	> 2.455	> 2.525	> 3.725	> 4.100
75%	> 0.765	> 4.565	> 3.625	> 5.135	> 5.135
100%	> 0.865	> 5.785	> 2.905	> 4.585	> 5.005

3.3 Genetic Algorithm Optimization

GA is a technique that alters input variables through mutation and crossover and checks the results to improve a fitness function that represents the performance of a system. In this case, desired fitness is determined by the performance measurements representing the whole system since the aim is to improve the entire traffic network. GA is used to investigate several performance measurements that include density, waiting time, flow, and speed. Density, waiting time, flow, and speed are related where density in relation to waiting time and flow in relation to speed have positive correlations. The two groups have negative correlations to each other. This push and pull between the performance measurements is important to consider in determining a proper representation of the overall performance of the system. Furthermore, a hard constraint is set for collisions since any simulation that produces collisions is considered infeasible since the model behavior is creating an unsafe environment. We use the IDM car following parameters acceleration, deceleration, minimum gap, tau, and delta to alter the IDM car following model behavior. We do this repeatedly for each of the performance measurements until GA does not produce a new solution with considerably improvement for a set amount of iterations. It is important that a fitness function is developed that includes all the performance measures of interest. With a fitness function that considers all four performance measures, it allows the implementation of weights to control what measurements take priority while considering the improvement of all measurements of interest.

3.3.1 Individual Performance Measurement Fitness Models

This section shows the convergence of GA where the models use fitness functions for individual performance measurements. The following results shown are for a traffic network with 100% IDM vehicle penetration rate. The convergence graphs for density, waiting time, flow, and speed are shown in Figure 3.4.



(a) Density (veh/km)

(b) Waiting Time (s)

(c) 1/Speed $(1/(m/s))$

(d) 1/Flow $(1/(veh/s))$

Figure 3.4: GA Convergence for Performance Measures

From the graphs above in Figure 3.4, the convergence of the performance measures are acceptable since they do not exhibit significant changes over thousands of iterations of simulations. Since GA looks to decrease the fitness function, the inverse of flow and speed are used to increase these measurements. Next, we look at the elite solutions found throughout the convergence to determine trends in IDM parameters and the relationships between the performance measures. Table 3.2 has the IDM parameters and performance measures of interest for convergence of density.

Table 3.2: IDM Parameters and Performance Measures for Convergence of Density

	IDM Parameter					Performance Measure			
Iteration	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed
2	2.53	3.66	4.99	1.67	3.07	4.36	113.18	50.06	10.09
6	4.08	5.71	5.34	1.80	5.30	4.12	82.45	51.44	10.46
80	3.68	5.79	10.44	1.28	2.94	4.11	88.07	53.00	10.83
194	2.37	5.79	3.79	1.38	2.94	4.04	81.38	51.36	10.50
210	4.44	5.79	3.79	1.06	4.86	4.03	88.86	52.60	10.89
348	4.49	5.79	4.20	1.06	3.70	3.99	85.87	52.87	10.83

Table 3.2 shows some notable trends with tau, and acceleration. Where tau converges toward 1(s) and acceleration converges toward 5.79(m/s^2). It is also shown that as the density converges, the flow and speed tend to increase while waiting time decreases. Table 3.3 has the IDM parameters and performance measures for convergence of waiting time.

Table 3.3: IDM Parameters and Performance Measures for Convergence of Waiting Time

	IDM Parameter					Performance Measure			
Iteration	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed
0	2.04	4.60	5.57	1.52	5.41	4.95	121.46	51.69	10.54
5	3.09	4.33	12.81	0.75	2.21	4.57	113.63	52.61	10.87
9	2.73	4.35	5.56	1.70	7.01	4.04	83.38	51.59	10.5
59	1.49	4.67	5.56	1.70	3.63	4.07	78.91	51.06	10.41
70	2.19	5.69	5.68	1.70	3.63	4.04	75.54	51.07	10.46
93	2.19	5.16	5.07	1.70	3.63	4.27	70.29	51.82	10.41
805	1.87	5.16	3.94	1.70	3.76	4.01	67.87	51.135	10.36
1975	1.87	4.84	6.81	1.55	3.76	4.10	66.69	51.32	10.57

Table 3.3 shows similar trends as Table 3.2. Where tau converges toward 1.55(s) and acceleration converges toward 4.84(m/s^2). It is also shown that as the waiting time converges, the density decreases with a few variations where they are lowered with waiting time improvements. Table 3.4 has the IDM parameters and performance measures of interest for convergence of speed.

Table 3.4: IDM Parameters and Performance Measures for Convergence of Speed

	IDM Parameter					Performance Measure			
Iteration	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed
0	3.76	4.68	9.50	0.81	2.94	4.45	137.16	53.05	10.91
18	2.45	5.04	10.07	1.25	7.62	4.38	130.53	53.10	10.94
24	2.21	4.66	10.07	0.88	5.49	4.38	131.88	53.31	11.08
29	2.21	5.01	10.07	0.88	4.66	4.23	113.52	53.52	11.12
38	2.21	4.85	10.07	0.88	4.66	4.38	96.51	53.70	11.14
65	2.21	5.18	10.07	0.88	4.66	4.19	104.49	53.83	11.16
119	2.21	5.75	14.05	0.88	4.66	4.43	127.01	52.95	11.20
135	2.82	5.59	14.05	0.88	5.74	4.30	103.38	53.36	11.21
177	2.82	5.59	12.83	0.88	7.55	4.38	111.94	53.44	11.23
611	1.90	5.59	13.96	0.88	7.55	4.34	118.55	52.95	11.26
1130	2.48	5.59	13.67	0.88	6.30	4.16	86.74	53.63	11.27
1352	2.48	5.59	13.67	0.88	7.84	4.33	90.94	54.14	11.28
2451	1.36	5.66	13.50	0.88	7.38	4.40	110.24	54.67	11.29
3653	1.37	5.76	13.50	0.89	7.38	4.20	94.19	54.43	11.30

Table 3.4 shows similar trends as Table 3.2 and 3.3. Where tau converges toward 0.89(s) and acceleration converges toward 5.76(m/s^2). It is also shown that as the speed converges, the density decreases and flow increases with a few variations where they are lowered with speed improvements. Table 3.5 has the IDM parameters and performance measures of interest for convergence of flow.

Table 3.5: IDM Parameters and Performance Measures for Convergence of Flow

	IDM Parameter					Performance Measure			
Iteration	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed
0	4.53	3.08	8.85	1.04	5.36	4.69	200.16	51.05	10.60
2	3.15	4.50	14.01	0.92	4.26	4.34	129.54	53.61	11.01
44	3.08	4.00	9.85	0.92	4.92	4.44	115.81	53.87	10.98
151	1.76	5.66	12.00	0.92	3.78	4.43	112.36	54.07	11.12
663	1.76	5.62	10.82	0.97	3.87	4.39	104.72	54.23	11.09
903	1.92	5.62	8.64	0.97	7.95	4.89	102.93	54.23	11.19
1190	1.92	5.62	8.64	0.97	6.31	4.33	117.67	54.30	11.15
1233	2.03	5.52	10.51	1.00	6.31	4.30	108.20	54.47	11.17

Table 3.5 shows similar trends as Tables 3.2, 3.3, and 3.4. Where tau converges toward $1(s)$ and acceleration converges toward $5.52(m/s^2)$. It is also shown that as the flow converges, the density decreases and speed increases with a few variations where they are lowered with flow improvements.

3.3.2 Standardization of Performance Measures

After running sensitivity analysis for individual performance measures we combine the datasets and create distributions for each performance measure of interest. From this data we can implement standardization to balance performance measures to determine the overall performance of a given simulation. The data is plotted below in Figure 3.5 showing the distribution of the four performance measures of interest.

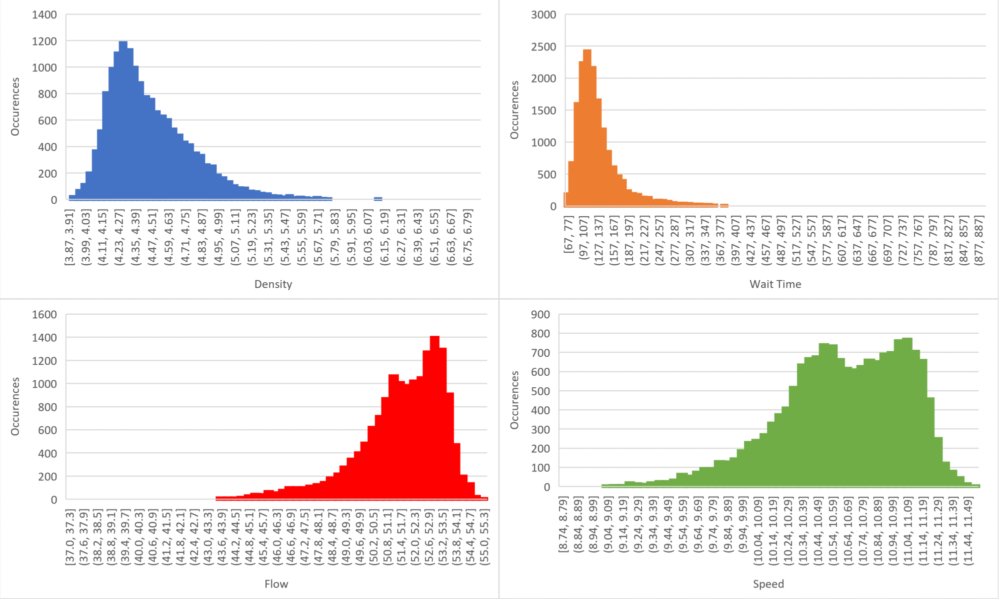


Figure 3.5: Distribution of Performance Measures

With the given plots, we see that the distributions are roughly normal with some leaning toward one side or the other. We further illustrate the distributions using whisker plots to show outliers in the distribution. In Figure 3.6 are the whisker plots for each of the performance measures.

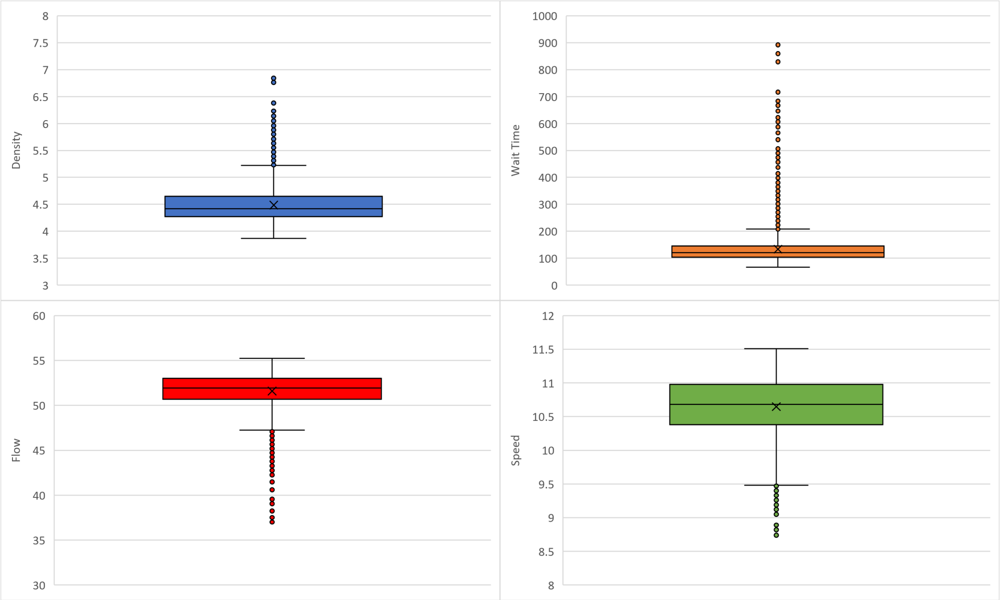


Figure 3.6: Whisker Plots of Performance Measures

The whisker plots prove that there are outliers present in each of the performance measure distributions. Given the outliers, the standardization method is a better fit for balancing performance measure. We calculate the following statistical metrics to be used to standardize each measure. Below is Table 3.6 showing the mean, median, minimum, maximum, and standard deviation.

Table 3.6: Statistical Data From GA Data Sets

Performance Measure	Mean	Median	Minimum	Maximum	Standard Deviation
density	4.48	4.42	3.87	6.84	0.30
waitingTime	133.46	120.67	66.69	892.08	51.81
flow	51.59	51.94	37.01	55.22	1.92
speed	10.65	10.68	8.74	11.51	0.41

Since there exists maximum and minimum values as outliers for given performance measures from the data-set, it is not ideal to use a normalization method. Given that the mean and median for performance measurement data sets are similar, we can say that the distributions are normal enough to where z-score is an effective method in standardizing the performance measures. We use the standardization z-score method to calculate values to represent the performance of the system for each measurement respectively. The next section discusses the z-score fitness method and how it is applied.

3.3.3 Multi-Performance Measurement Fitness Model

The most significant differences between the three algorithms is the minimum gap, deceleration, and delta. When density is used as convergence the minimum gap converges to 4.49(m), deceleration converges to 4.2(m/s^2), and delta converges toward 3.7. When flow is used as convergence the minimum gap converges to 2.03(m), deceleration converges to 10.51(m/s^2), and delta converges toward 6.31. When speed is used as convergence the minimum gap converges to 1.37(m), deceleration converges to 13.5(m/s^2), and delta converges toward 7.38. This differences are significant enough to propose a new method in determining a optimal solution where density, waiting time, flow, and speed are all considered into the algorithm. Next we use GA with a combined model including the four performance measurements discussed above. To standardize the data we use the Z-score for each performance measurement set created from the previous GAs. Each GA data set is changed into a distribution for the performance measurement of interest. The z-score equation is given below.

$$Z - score = \frac{x - \mu}{\sigma} \tag{3.1}$$

Where x is the performance measure for each iteration of the GA, μ is the mean calculated from the previous GA data set, and σ is the standard deviation for the previous GA data set. This method allows for each measurement to be considered equally in the combined fitness function. To combine the performance measures we denote each variable with an abbreviation relating the variable to the performance measure it represents. The following equation is developed using the z-score equation to consider all four performance measures.

$$fitness = \left(\frac{x_{dens} - \mu_{dens}}{\sigma_{dens}} + \frac{x_{wait} - \mu_{wait}}{\sigma_{wait}}\right) - \left(\frac{x_{flow} - \mu_{flow}}{\sigma_{flow}} + \frac{x_{speed} - \mu_{speed}}{\sigma_{speed}}\right) \tag{3.2}$$

The above equation takes into account each of the performance measures values and standardizes them so that they are of equal importance in the fitness function. The subscripts of *dens*, *wait*, *flow*, and *speed* reflect the density, waiting time, flow, and speed respectively. Since the flow and speed are variables that are desired to increased, we give their Z-scores a negative value. Note, that weights can be added to each Z-score of Equation 3.2. We use a multiplier of 0.25 to each weight since there are four performance measures of interest. This results in the following shown below in Equation 3.3.

$$\begin{aligned} fitness = &\left(w_{dens}\frac{x_{dens}-\mu_{dens}}{\sigma_{dens}} + w_{wait}\frac{x_{wait}-\mu_{wait}}{\sigma_{wait}}\right) \\ &- \left(w_{flow}\frac{x_{flow}-\mu_{flow}}{\sigma_{flow}} + w_{speed}\frac{x_{speed}-\mu_{speed}}{\sigma_{speed}}\right) \end{aligned} \tag{3.3}$$

$$w_{dens} = w_{wait} = w_{flow} = w_{speed} = 0.25 \tag{3.4}$$

We use w to denote the weight variable of each performance measure with its subscript representing the parameter it pertains to. For our purposes, we decide to leave the weights as 1 since there has been no evidence that the importance of any specific performance measure is greater than the others. Given our final fitness function, we implement this in our GA to find a well rounded solution. The graph showing the convergence for the combined fitness function is shown below.

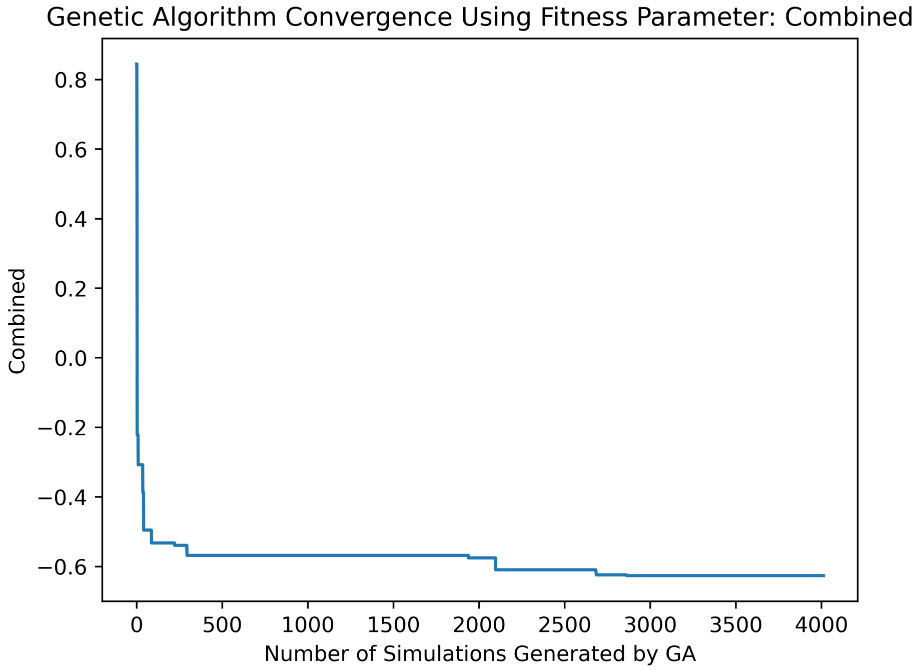


Figure 3.7: Convergence at 2500 Iterations for Combined Fitness Function Including Density, Waiting Time, Flow, and Speed using Combined Z-scores Equation (Refer to Equation 3.3)

From the graph above in Figure 3.7, the convergence is acceptable since it does not exhibit significant changes over thousands of iterations of simulation. Next, we look at the elite solutions found during convergence. Table 3.7 shows the IDM parameters and performance measurements for the GA that uses the combined Z-score fitness function.

Table 3.7: IDM Parameters and Performance Measures for Convergence of Combined Model

	IDM Parameter					Performance Measure				
Iteration	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed	Fitness
0	2.57	2.14	6.55	1.95	2.33	4.71	159.99	46.15	9.40	0.84
2	2.27	1.96	14.59	1.04	7.54	4.86	200.92	49.56	10.39	0.47
3	4.18	4.57	9.01	1.69	5.20	4.20	84.33	51.99	10.56	-0.22
9	2.96	5.72	13.61	1.02	2.41	4.38	113.12	53.54	10.96	-0.30
36	4.04	5.72	8.13	1.02	5.08	4.22	105.93	53.16	11.04	-0.38
41	2.96	5.72	13.61	1.02	7.98	4.20	90.62	53.59	11.18	-0.49
87	2.96	5.72	3.93	0.86	7.98	4.07	97.04	53.89	11.12	-0.53
222	4.89	5.72	3.93	0.80	6.09	4.00	89.18	53.43	11.09	-0.54
294	4.89	5.72	3.93	0.80	6.17	4.03	88.37	53.87	11.13	-0.57
1938	4.89	5.72	12.57	0.80	7.99	4.13	98.60	54.23	11.28	-0.58
2097	3.87	5.27	12.18	0.80	7.99	4.09	91.18	54.28	11.28	-0.61
2685	3.87	5.27	12.14	0.80	7.99	4.07	87.62	54.37	11.26	-0.62
2862	3.54	5.27	12.14	0.80	7.99	4.00	82.59	53.78	11.26	-0.63

From Table 3.7 we see as the fitness decreases, the density and waiting time decrease while the flow and speed increase as desired. There is some push and pull between the performance measures but over all they tend to improve with each elite solution found. We see similar trends with IDM parameters acceleration and tau. Acceleration converged toward 5.27 where tau converges toward 0.80. Further investigation into the lower tau limit is needed since the definition of tau is the headway time where below 1 means the model is giving a headway time below the step time that the simulation runs at. Next, we look at the different solutions from the fitness functions to determine which method produces a well rounded solution and to select a final set of IDM parameters for our model.

3.3.4 Solutions for Fitness Functions

This section shows the final solutions produced through GA utilizing different fitness functions for both individual performance measurements and combining performance measurements. Table 3.8 shows the different fitness types and their final solutions produced from applying GA to the traffic simulation.

Table 3.8: IDM Parameters and Performance Measures for Convergence of GA Models

	IDM Parameter					Performance Measure			
Fitness Type	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed
Density	4.49	5.79	4.20	1.06	3.70	3.99	85.87	52.87	10.83
Waiting Time	1.87	4.84	6.81	1.55	3.76	4.10	66.69	51.32	10.57
Flow	2.03	5.52	10.51	1.00	6.31	4.30	108.20	54.47	11.17
Speed	1.37	5.76	13.50	0.89	7.38	4.20	94.19	54.43	11.30
Combined	3.54	5.27	12.14	0.80	7.99	4.00	82.59	53.78	11.26

From Table 3.8 we see the results from the different fitness functions. There are correlations between certain performance measurements. The waiting time and density have a direct correlation where the waiting time decreases as does the density. It is also seen that as the flow increases, the speed also increases. The combined fitness shows the second lowest density, third highest flow, second highest speed, and second lowest waiting time. This proves that the combined Z-score fitness function successfully weighs the performance measurements equally and results in solution that takes into account each of the measurements. Next, the combined performance measure GA is applied to penetration rates of IDM for 25%, 50%, 75%, and 100%. An additional limitation is added where tau's lower limit is set to 0.86 since, given simulation results, this is a threshold for when collisions become most likely.

3.3.5 Results for Varying Penetration Rates of IDM Vehicles

Using the combined performance measure fitness function and inserting a lower limit of tau determine to be 0.86, we perform GA for four penetration rates. The penetration rates are 25%, 50%, 75%, and 100%. We perform this step to analyze the improvement that can be made to mixed traffic scenarios of varying ratios and to determine if driving behavioral model parameters should be altered for different penetration rates of IDM vehicles. Figure 3.8 shows the convergence of the combined performance measure fitness at the various penetration rates.

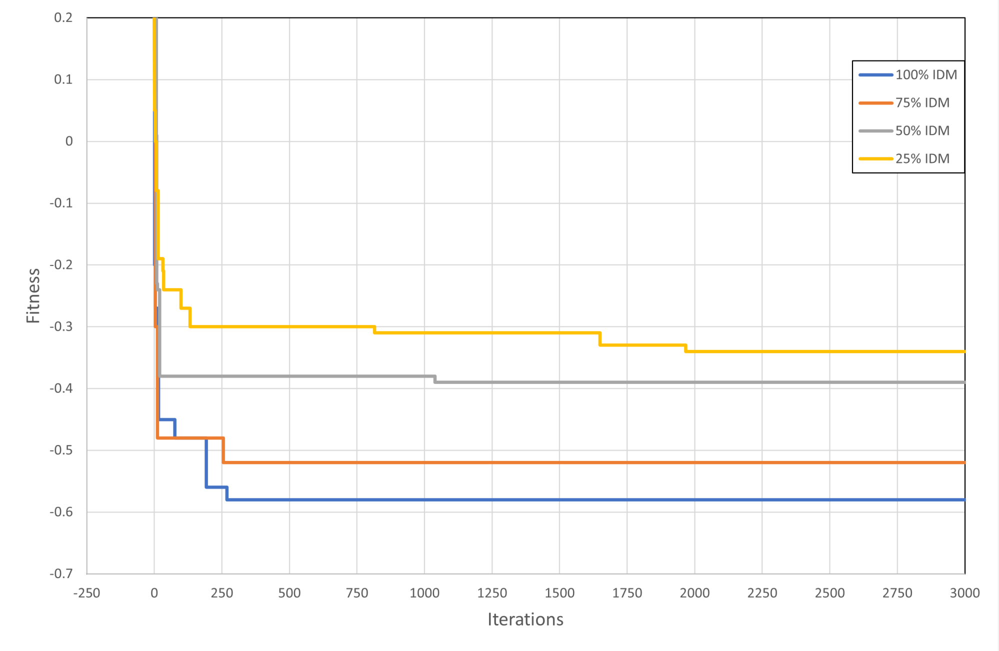


Figure 3.8: Convergence for Penetration Rates of IDM Vehicles

Given the convergence shown above, we see that as the penetration rate of IDM vehicles increases, the improvement in performance also increases. We can see that at lower penetration rates the convergence takes longer than the simulations with more IDM vehicles. Table 3.9 shows the IDM parameters and performance measures for the penetration rates analyzed.

Table 3.9: IDM Parameters and Performance Measures for Convergence of IDM Penetration Rates

	IDM Parameter					Performance Measure				
Rate	minGap	accel	decel	tau	delta	density	waitingTime	flow	speed	Fitness
25%	1.75	5.31	13.64	1.1	2.6	4.2	90.94	53.06	10.79	-0.34
50%	1.51	5.06	6.984	1.0	6.53	4.09	92.68	52.51	10.91	-0.39
75%	1.53	4.7	5.81	0.92	6.98	4.12	81.47	53.78	11.06	-0.52
100%	4.05	5.58	4.67	0.93	5.84	4.06	76.4	54.02	11.08	-0.58

As the penetration rate increases, the overall performance can be improved. We see that the density, waiting time, flow, and speed are generally improved as more IDM vehicles are introduced into the network. There are some changes in IDM behavioral parameters where the most significant changes are with minGap, decel, and delta. The minGap stays around 1.6 m for lower penetration rates and increase to 4.05 m at 100% penetration. The decel parameter decreases from 13.64 m/s^2 to 4.67 m/s^2 as the penetration of IDM vehicles increases. Delta is 2.6 at 25% and increases to 6.53 at 50% where it has some variation between the final two penetration rates. The next section discusses the relationships between performance measures where the optimal solutions are shown visually by combining performance measures with positive correlations.

3.3.6 Correlation of Performance Measures

Plotting the performance measures establishes the possible solutions based on what performance measure are desired to be increased. With the use of weights, the performance measures can take priority over others when calculating the overall performance of a system. For our purposes, we combine measures that have the same dimensional impact to be able to illustrate a simplistic representation of the performance measure correlation. Density and waiting time are both measurements that are desired to be reduced to show improvement in a network, on the other hand, flow and speed are combined since the goal for these measures is to be increased. When combining the performance measures, the z-score method is used once again to make sure that the combined measurement are represented equally in each axis. The goals for the performance measures are to decrease density and waiting time and to increase flow and speed. Figure 3.9 shows the plotted combined data-set created from GA algorithms.

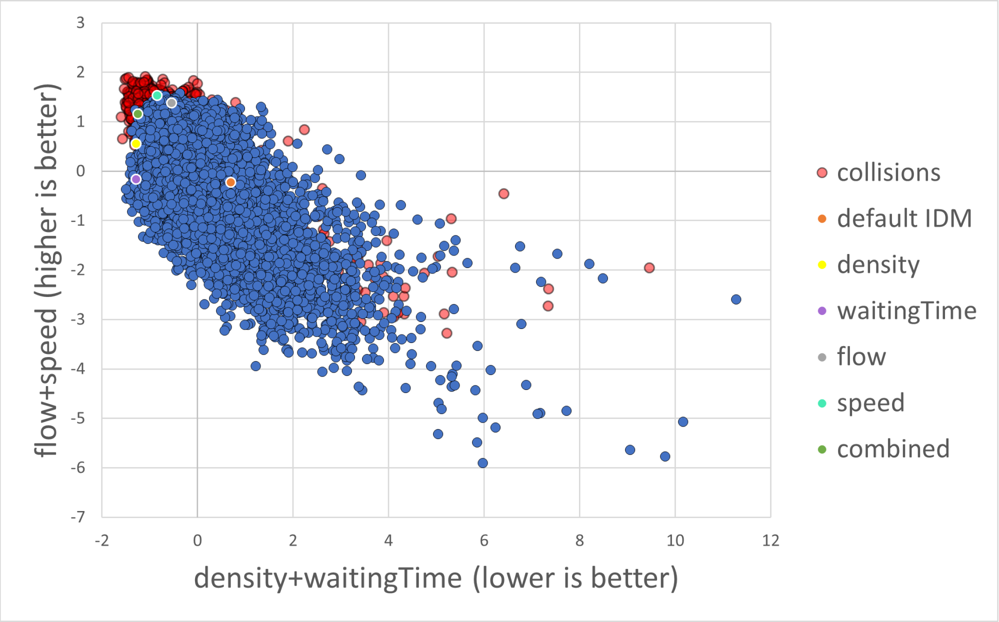


Figure 3.9: Correlation of Performance Measures, Density, Waiting Time, Flow, and Speed for Simulations of Altered IDM Parameters using Z-score to Combine Measures with Aligning Goals

The above plot shows all performance measures from different simulations of 100% IDM vehicles. The solutions for given performance measures are shown in the legend. The blue points are simulations that had no collisions where the transparent red dots are simulations where collisions occurred. We can see from the plot that depending on the fitness function used, there is bias toward the performance measure used. When using the combined fitness function, the overall performance is balanced among the given performances measure. The performance of the simulation using default IDM model parameters is included to show how the GA's improved the performance of the system.

Chapter 4

Conclusions

The framework proposed applies sensitivity analysis, standardization, and GA to investigate how much self-driving vehicle behavioral model parameters can improve traffic performance while taking into consideration safety related constraints. The framework includes feasibility analysis, performance measure standardization, combined parameter optimization, and behavioral model validation. This framework was applied to four penetration rates of IDM vehicles including 25%, 50%, 75%, and 100% IDM Vehicles. Sensitivity analysis reveals that the penetration rates of IDM vehicles have impact on the traffic performance measures and that optimal IDM behavior parameter value can improve the traffic performance. As penetration rates of IDM vehicles increase, notable performance improvement is observed. A CART analysis performed on the constraint of achieving zero collisions, refined the boundaries of the feasible region for the IDM behavior parameter tau as the hard constraint for safety while searching optimal parameter values. The GA was used for convergence of performance measures including density, waiting time, flow, and speed. It was observed that each model had similar results for specific parameters including tau and acceleration while other parameter values varied. This shows that we can determine solutions for tau and acceleration based on the algorithm results. In order to consider several simultaneous performance measures, a fitness function that used z-score was used. Depending on priority, the weights can be changed to emphasize performance measure importance. The results from this study have successfully demonstrated that this framework can be implemented to develop an improved behavioral model for autonomous vehicles while ensuring safety.

Chapter 5

Future Work

This studies focus was on the development of a model-based simulation framework to improve vehicle behavior models. Results are based on road networks and demand generated through SUMO instead of real traffic data because the fidelity of the simulation software has already been proven by multiple researchers. Still, we can use real traffic data input if we want to estimate specific situation or time. Another addition for this framework is to integrate automotive standards for self-driving vehicles. There are standards given through government organizations [3] as well as car manufactures who apply their own standards for self-driving vehicles. Since manufacturers are applying deep learning to develop self-driving models [9]. It would be beneficial to import these models into simulation environments to apply the provided framework. It may allow developers, manufacturers, or future traffic network planners to implement their models in a traffic simulation similar a virtual twin to help improve their models. Implementation for a variety of road networks will definitely improve the fidelity of this framework. This is important since the vehicles in the physical world will come across different situations that may not be present in all simulated traffic networks. To improve the solution, machine learning methods could be implemented so that the model responds to traffic scenarios differently based on attributes of the road type, speed limits, congestion, and interactions observed by other vehicles. It is important to consider that certain vehicle behavior will be better suited for different traffic situations. Given the listed additions discussed in this section, the self-driving behavior model optimization framework will provide directions that can be applied to real-world self-driving vehicles.

BIBLIOGRAPHY

[1] Erdmann, J. Lane-Changing Model in SUMO. In *Proceedings of the SUMO2014 Modeling Mobility with Open Data* (Berlin, Deutschland, May 2014), vol. 24, Deutsches Zentrum für Luft- und Raumfahrt e.V., pp. 77–88. ISSN: 1866-721X.

[2] Gonzalez-Delicado, J. J., Gozalvez, J., Mena-Oreja, J., Sepulcre, M., and Coll-Perales, B. Alicante-Murcia Freeway Scenario: A High-Accuracy and Large-Scale Traffic Simulation Scenario Generated Using a Novel Traffic Demand Calibration Method in SUMO. *IEEE Access 9* (2021), 154423–154434.

[3] Griffor, E., Wollman, D., and Greer, C. Automated driving system safety measurement part 1: Operating envelope specification, 2021-12-02 05:12:00 2021.

[4] Guériau, M., and Dusparic, I. Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (Sept. 2020), pp. 1–8.

[5] Krauss, S. Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics.

[6] Liu, H., Wei, H., Zuo, T., Li, Z., and Yang, Y. J. Fine-tuning ADAS algorithm parameters for optimizing traffic safety and mobility in connected vehicle environment. *Transportation research. Part C, Emerging technologies 76* (Mar. 2017), 132–149.

[7] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wiessner, E. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems* (2018), IEEE.

[8] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[9] Santana, E., and Hotz, G. Learning a Driving Simulator. *arXiv:1608.01230 [cs, stat]* (Aug. 2016). arXiv: 1608.01230.

[10] Sofla, R. A. D., Alipour-Fard, T., and Arefi, H. Road extraction from satellite and aerial image using SE-Unet. *Journal of Applied Remote Sensing 15*, 1 (Mar. 2021), 014512.

[11] Treiber, M., Hennecke, A., and Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E 62*, 2 (Aug. 2000), 1805–1824. Publisher: American Physical Society.