



2007-03-16

Calibration Procedure for a Microscopic Traffic Simulation Model

Carole Turley

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Civil and Environmental Engineering Commons](#)

BYU ScholarsArchive Citation

Turley, Carole, "Calibration Procedure for a Microscopic Traffic Simulation Model" (2007). *All Theses and Dissertations*. 846.
<https://scholarsarchive.byu.edu/etd/846>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

CALIBRATION PROCEDURE FOR A MICROSCOPIC
TRAFFIC SIMULATION MODEL

by

Carole Turley

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering

Brigham Young University

April 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Carole Turley

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Grant G. Schultz, Chair

Date

Mitsuru Saito

Date

E. James Nelson

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Carole Turley in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Grant G. Schultz
Chair, Graduate Committee

Accepted for the Department

E. James Nelson
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering
and Technology

ABSTRACT

CALIBRATION PROCEDURE FOR A MICROSCOPIC TRAFFIC SIMULATION MODEL

Carole Turley

Department of Civil and Environmental Engineering

Master of Science

The inputs to a microscopic traffic simulation model generally include quantitative, but immeasurable data describing driver behavior and vehicle performance characteristics. Engineers often use default values for parameters such as car-following sensitivity and gap acceptance because it can be difficult to obtain accurate estimates for these parameters.

While recent research has indicated that the accuracy of a simulation model can significantly improve if driver behavior parameters are calibrated to local data, this is not a typical practice. Manual calibration of these parameters is often too time-consuming to be cost-effective. Researchers have applied automated calibration procedures using genetic algorithms to these problems with some success, but many engineers lack the tools or the skill set necessary to easily program and implement such a procedure. A

graphical user interface (GUI) for a genetic algorithm would make automated calibration much more accessible to students and professional engineers.

Another barrier that limits the practicality of calibrating driver behavior parameters is the number of available calibration parameters. The CORSIM (short for CORridor SIMulation) model developed by the Federal Highway Administration contains dozens of optional calibration parameters controlling various aspects of driver behavior. Determining the sensitivity of the model to these parameters is an important step toward finding the appropriate parameter values.

The purpose of this thesis is to develop a GUI for a genetic algorithm and perform needed sensitivity analyses to aid in model development and calibration. This thesis tests a simple, automated procedure utilizing a genetic algorithm for the calibration of driver behavior and vehicle performance parameters that can easily be applied by engineers in the field. An existing genetic algorithm script that has been applied to other research has been adapted to fit the purposes of this thesis. As part of this procedure, a sensitivity analysis was performed to recommend which parameters should be included in model calibration.

The results of the research suggest that fewer than half of the available driver behavior parameters are necessary to calibrate a model of a linear freeway network. The calibration tests also demonstrate the importance of calibrating to at least two measures of effectiveness in order to better match existing conditions and provide an acceptable level of error for the simulation model.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Grant G. Schultz, for his encouragement, ideas, and support throughout the research process. I would also like to thank my committee members, Dr. Mitsuru Saito and Dr. E. James Nelson for the contribution they have made to my undergraduate and graduate education and that of many other students at Brigham Young University.

The Ira A. Fulton College of Engineering and Technology provided the funding for this project in the form of a generous grant. Eric Rasband at the Utah Department of Transportation was very helpful in providing traffic volume data and insight into the data collection process.

Several friends contributed to this project in important ways. Kordel Braley, Paul Jensen, and Hunter Young were kind enough to allow me to commandeer their research computers to run some long calibration tests. I am also very grateful to Shannon Francis, Chelsea Pollock, Jonathan Johnson, Boyd Bills, and Megan Rasmussen for taking the time to help collect travel time data. Thank you for your help and most especially for your friendship.

Finally, I would like to thank my parents, Steve and Deon Turley for teaching me to value excellence and for encouraging me to reach my potential.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	vii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
1 Introduction.....	1
1.1 Background.....	2
1.1.1 Traffic Simulation Models.....	2
1.1.2 Genetic Algorithms.....	3
1.2 Problem Statement	3
1.3 Research Objectives.....	4
1.4 Organization of the Thesis	4
1.4.1 Literature Review.....	5
1.4.2 Calibration Program.....	5
1.4.3 Data Collection	5
1.4.4 Calibration Procedure	6
1.4.5 Conclusion	6
2 Literature Review.....	7
2.1 Traffic Analysis Tools	7
2.1.1 Sketch Planning Tools	8

2.1.2	Travel Demand Models.....	9
2.1.3	Analytical Tools.....	10
2.1.4	Traffic Signal Optimization Tools.....	11
2.1.5	Traffic Simulation Models.....	11
2.1.6	Common Microscopic Traffic Simulation Models.....	14
2.2	Car-following Models.....	17
2.2.1	Generalized General Motors Model.....	18
2.2.2	PITT Car-following Model.....	19
2.3	Genetic Algorithms.....	20
2.3.1	Alternative Optimization Methods.....	20
2.3.2	Genetics Terminology.....	22
2.3.3	Components of a Genetic Algorithm.....	22
2.3.4	History of the Genetic Algorithm.....	23
2.3.5	Implementing a Genetic Algorithm.....	25
2.4	Calibration of Microscopic Traffic Simulation Models.....	32
2.4.1	Singapore Expressway Model.....	32
2.4.2	Covington, Virginia Highway Model.....	34
2.4.3	Houston, Texas Interstate Model.....	36
2.5	Concluding Remarks.....	36
3	Calibration Program.....	39
3.1	Graphical User Interface.....	40
3.1.1	File Management.....	40
3.1.2	Calibration Control.....	44

3.1.3	Driver Behavior Parameters.....	45
3.2	Genetic Algorithm Implementation	51
3.2.1	Initialization	51
3.2.2	Iterations	56
3.3	Concluding Remarks.....	58
4	Data Collection	61
4.1	Test Site	61
4.2	Data Collection and Reduction	62
4.2.1	Geometry Data	63
4.2.2	Volume Data	64
4.2.3	Travel Time Data	66
4.3	Concluding Remarks.....	87
5	Calibration Procedure	89
5.1	Sensitivity Analysis	89
5.1.1	Parameter Selection	89
5.1.2	Determination of Population Size.....	95
5.2	Calibration.....	99
5.3	Validation.....	101
5.4	Concluding Remarks.....	104
6	Conclusion	107
6.1	Sensitivity Analysis	107
6.1.1	Sensitivity Analysis Conclusions.....	108
6.1.2	Questions Raised by Sensitivity Analysis.....	108

6.2	Calibration.....	109
6.2.1	Calibration Conclusions.....	109
6.2.2	Questions Raised by Calibration.....	109
6.3	Validation.....	110
6.3.1	Validation Conclusions.....	110
6.3.2	Questions Raised by Validation.....	111
6.4	Concluding Remarks.....	111
	References.....	113
	Appendix A. Sample Input File for Calibration Program.....	119
	Appendix B. Sample Files from Calibration Program.....	121
	Appendix C. Traffic Volume Counts from UDOT.....	133
	Appendix D. MOEs from Parameter Selection Analysis.....	135

LIST OF TABLES

Table 3-1: Default Car-Following Parameters.....	47
Table 3-2: Default Values for Lane Changing Parameters.....	49
Table 4-1: Calculation of Mainline Entry Volumes	64
Table 4-2: Times Recorded at Node Locations for Northbound Runs	76
Table 4-3: Northbound Travel Times	77
Table 4-4: Times Recorded at Node Locations for Southbound Runs	86
Table 4-5: Southbound Travel Times	87
Table 5-1: Selected MOEs Resulting from Varying Random Number Seeds.....	91
Table 5-2: Ranges of MOEs from Varying Parameter Values	93
Table 5-3: Default Ranges and Results of Sensitivity Analysis	94
Table 5-4: Average Results of Population Size Analysis	97
Table 5-5: Fitness and MAER Values of Default and Calibrated Parameter Sets.....	102
Table 5-6: Parameter Sets Generated by Five Calibration Tests	103
Table 5-7: Results of Validation Tests.....	103

LIST OF FIGURES

Figure 2-1: Schematic of single-point crossover.	30
Figure 2-2: Schematic of multiple-point crossover.	30
Figure 3-1: File Management dialog box.....	41
Figure 3-2: Calibration control dialog box.	45
Figure 3-3: Driver Behavior Parameters dialog box.....	46
Figure 3-4: Vehicle Performance Parameters dialog box.	50
Figure 3-5: Genetic algorithm procedure.....	52
Figure 4-1: Location of the I-15 test site.	63
Figure 4-2: Observed ramp volumes with calculated mainline volumes.....	65
Figure 4-3: Speed profile for northbound run 1.....	68
Figure 4-4: Speed profile for northbound run 2.....	69
Figure 4-5: Speed profile for northbound run 3.....	69
Figure 4-6: Speed profile for northbound run 4.....	70
Figure 4-7: Speed profile for northbound run 5.....	70
Figure 4-8: Speed map for northbound run 1.....	71
Figure 4-9: Speed map for northbound run 2.....	72
Figure 4-10: Speed map for northbound run 3.....	73
Figure 4-11: Speed map for northbound run 4.....	74
Figure 4-12: Speed map for northbound run 5.....	75

Figure 4-13: Speed profile for southbound run 1.....	78
Figure 4-14: Speed profile for southbound run 2.....	79
Figure 4-15: Speed profile for southbound run 3.....	79
Figure 4-16: Speed profile for southbound run 4.....	80
Figure 4-17: Speed profile for southbound run 5.....	80
Figure 4-18: Speed map for southbound run 1.	81
Figure 4-19: Speed map for southbound run 2.	82
Figure 4-20: Speed map for southbound run 3.	83
Figure 4-21: Speed map for southbound run 4.	84
Figure 4-22: Speed map for southbound run 5.	85
Figure 5-1: Comparison between population sizes.	96
Figure 5-2: Comparison of hybrid, smallest, and largest populations.	97
Figure 5-3: Comparison of average time required between population sizes.....	98
Figure 5-4: Change in volume MAER versus decrease in travel time MAER.....	100

1 Introduction

Researchers in a variety of fields have studied the ability of computers to imitate and predict natural phenomena. Traffic simulation models attempt to mimic a population of drivers in a theoretical highway network. In attempting to simulate human behavior and individual decisions, traffic simulation models must begin with quantitative, but often immeasurable data describing driver characteristics. Some researchers have addressed the problem of determining the appropriate values of variables describing driver and vehicle characteristics such as courtesy, aggressiveness, familiarity with routes, and maximum deceleration rates.

Genetic algorithms have proven to be a useful tool in calibrating driver behavior and vehicle performance parameters. Like traffic simulation models, genetic algorithms model behavior observed in the real world. Using principles of natural selection, these algorithms can simulate the “evolution” of a set of parameters that can realistically represent driver behavior.

While recent research has indicated that the accuracy of a simulation model can significantly improve through the use of automated calibration procedures, this is not a typical practice. Many engineers lack the tools or the skill set necessary to easily program and implement such a procedure. A graphical user’s interface (GUI) is necessary to make automated calibration more accessible to professional engineers.

Another barrier that limits the practicality of calibrating driver behavior parameters is the number of available calibration parameters. The CORSIM (short for CORridor SIMulation) model developed by the Federal Highway Administration (FHWA) contains dozens of optional calibration parameters controlling various aspects of driver behavior. Determining the sensitivity of the model to these parameters is an important step toward finding the appropriate parameter values.

This chapter presents a brief overview of the history and background of traffic simulation models, describes the specific problems that this thesis addresses, and summarizes the remaining chapters to address the problems identified.

1.1 Background

Both traffic simulation modeling and genetic algorithms have a history dating to the middle of the twentieth century. Over the past decade, researchers have developed ways to combine the two technologies for calibration purposes. This section presents some history and background on both types of simulation.

1.1.1 Traffic Simulation Models

Research into traffic simulation modeling began in the 1950s, not long after the introduction of the first digital electronic computers, but simulation was not commonly used by engineers until computers became commonplace in the 1970s. CORSIM was developed by the FHWA in the 1970s and quickly became one of the earliest widely used traffic simulation models (1). It became even more popular in the 1990s with the release of the first version of the Traffic Software Integrated System (TSIS), a GUI that assists in

pre- and post-processing of CORSIM models (2). Since that time, the number of commercially available traffic simulation packages has proliferated to include VISSIM (Verkehr In Staedten SIMulation), SimTraffic, Paramics, and others. Individual software packages are discussed in Chapter 2.

1.1.2 Genetic Algorithms

John Holland developed the first genetic algorithms at the University of Michigan in the 1960s. The first applications to civil engineering problems were not long in following. David Goldberg's landmark dissertation applying a genetic algorithm to a pipe optimization problem was among the first demonstrations of how these algorithms could find solutions to complex, real-world problems (3).

Cheu et al. (4) published one of the first applications of a genetic algorithm to traffic simulation was in the 1990s. Since then, several researchers have looked at the possibility of applying genetic algorithms to calibration problems, but this has yet to become state-of-the-practice for transportation engineers.

1.2 Problem Statement

Research has shown that calibration of driver behavior parameters using a genetic algorithm can efficiently improve the accuracy of microscopic traffic simulation models; however, practicing engineers rarely take this approach. Many engineers do not have the tools or the skill set to carry out calibration procedures which have been performed by researchers. A GUI for an automated calibration procedure is necessary to make these tools more available to students and professionals. Also, CORSIM has dozens of optional

parameters which could be calibrated. Determining which of these should be adjusted can be both difficult and time-consuming. Research into the sensitivity of the CORSIM model to each of the optional calibration parameters was necessary.

1.3 Research Objectives

The purpose of this thesis is to test a simple procedure using a genetic algorithm for the calibration of driver behavior and vehicle performance parameters that can easily be applied by engineers in the field. An existing genetic algorithm script that has been applied to other research has been adapted to fit the purposes of this thesis. As part of this procedure, a sensitivity analysis determined which parameters should be included in model calibration. This thesis also demonstrates the effects of calibrating to two different measures of effectiveness (MOEs) (volume and travel time).

In order to make this procedure more available to engineers and students who may not be comfortable with programming or with DOS-based interfaces, a dialog-based GUI was created as part of this research. The GUI allows users to easily calibrate CORSIM parameters using the genetic algorithm that was applied in this research.

1.4 Organization of the Thesis

This thesis is organized into six chapters: this introduction, a literature review, a description of the calibration software developed for this research, a presentation of the methodologies used to collect and reduce data for the calibration and validation tests, a description of how the proposed procedure was tested on a local freeway network, and a conclusion. An overview of each main chapter follows.

1.4.1 Literature Review

Chapter 2 places microscopic traffic simulation in the context of the larger range of traffic analysis tools. Several categories of traffic analysis tools are discussed, with emphasis on commercially available microscopic traffic simulation software. A brief history of the development of the genetic algorithm is also given, as well as some background on the components and structure of a genetic algorithm. Finally, recent research on the utilization of genetic algorithms for the calibration of microscopic traffic simulation models is summarized and discussed.

1.4.2 Calibration Program

Chapter 3 describes the genetic algorithm used for this research and the GUI that was developed for this thesis. This chapter describes each parameter that can be adjusted by the calibration program and may be used as a user's manual by others wishing to apply this software to other research or field applications.

1.4.3 Data Collection

In order to test the proposed calibration procedure, a model of a hypothetical network based on Interstate 15 in Utah County, Utah was calibrated to volume and travel time data. Some volume data for the network was provided by the Utah Department of Transportation (UDOT). Chapter 4 describes the methodologies used to reduce the available volume data and prepare them to be input to the calibration program. It also presents the methodologies used to collect and reduce travel time and speed data.

1.4.4 Calibration Procedure

Chapter 5 presents the methods and results of the three steps of the calibration procedure. First, it describes the sensitivity analysis that was performed to determine the appropriate population size and to select the parameters that should be included in the calibration. Second, it describes how the model was calibrated using the genetic algorithm. The calibration step will be performed five times to demonstrate the effects changes to the fitness function can have on model outputs. Finally, the parameter sets generated by the calibration step will be validated on a separate network with similar characteristics.

1.4.5 Conclusion

Chapter 6 summarizes the conclusions from the sensitivity analysis, calibration, and validation tests. Questions that were raised by the research presented in this thesis are also identified as areas that may warrant further research.

2 Literature Review

This chapter describes several traffic analysis tools, with an emphasis on microscopic traffic simulation. It then presents a brief background on genetic algorithms. Finally, recent research applying genetic algorithms to the calibration of driver and vehicle parameters in traffic simulation models is discussed.

2.1 Traffic Analysis Tools

Understanding the consequences of a highway or traffic operations improvement before project implementation is essential to traffic engineers and planners. Decision-makers are hesitant to invest in highway projects that promise to improve safety and reduce congestion without some evidence that these promises are realizable. Experimenting with the transportation system can be costly, dangerous, and impractical. Engineers have been able to avoid such experiments by turning to computer software for a comparatively low-cost method of analyzing transportation projects.

Traffic software has been developed for applications at every level of planning and traffic analysis. Depending on the objectives of a given study, any of a variety of tools may be appropriate. In some cases, one tool might be capable of a particular analysis, but not as well suited as another that has been designed especially for that purpose. Alexiadis et al. (5) identify seven categories of traffic analysis tools:

- Sketch planning tools,
- Travel demand models,
- Analytical tools,
- Traffic signal optimization tools, and
- Traffic simulation models (macro- meso- and microscopic).

Each of these categories of traffic analysis tools is suited to different situations and applications. This thesis focuses on microscopic traffic simulation models. In order to understand what traffic simulation is, it is important to understand what it is not. For this reason, sketch planning tools, travel demand models, analytical tools, and traffic signal optimization tools are briefly discussed here. Examples of tools in each category are listed as these tools are discussed. A more complete list is available in the literature (5).

2.1.1 Sketch Planning Tools

Sketch planning tools use simplified planning techniques and highly aggregated data to produce general, order-of-magnitude estimates of travel demand. Sketch planning tools can range from simple spreadsheet models to more sophisticated software packages (5). Tools in this category allow engineers and planners to quickly and cheaply screen a large number of alternatives in order to create a short list of viable alternatives that will undergo a more detailed analysis (6). Some of the existing sketch planning tools include:

- HDM-4 (Highway Development and Management) (7),
- QuickZone (8),

- STEAM (Surface Transportation Efficiency Analysis Model) (9), and
- TEAPAC (Traffic Engineering Applications PACKage) SITE (10).

2.1.2 *Travel Demand Models*

Travel demand models use current traffic and transit ridership characteristics along with population and employment forecasts to predict future travel demand in terms of mode choice, destination, temporal distribution, route choice, etc. Travel demand models were originally developed to determine the impacts of major highway projects. Travel demand models were not designed to evaluate operational changes or travel management strategies (5).

Traditional travel demand models employ an approach known as the Urban Transportation Modeling System (UTMS), which is also called the four-step planning model. The four-step planning model involves predicting the number of trips produced by and attracted to each zone (trip generation), creating origin-destination (O-D) matrices which link origins and destinations (trip distribution), determining the portion of travelers that will use each available mode (mode split), and assigning each trip to a particular route (trip assignment). This method, though widely used, has been criticized almost from its inception because it is not based on a coherent theory of travel behavior, leading to some inconsistencies within the model (6).

More recent developments in travel demand modeling have utilized an activity-based approach (11). This approach recognizes travel as a derived demand. In other words, people travel because they want to do something at a place other than where they are, rather than because they want to make a trip for its own sake (6). This recognition

has created more intuitive logic within the models, but the data requirements for such models are much more complicated than the simple population and employment statistics necessary for the UTMS.

Some of the existing travel demand models include:

- Quick Response System (QRS) II (12),
- TransCAD (13),
- CUBE/MinUTP (14),
- CUBE/ Transportation Planning Plus (TP+) (15), and
- VISUM (16).

2.1.3 Analytical Tools

Analytical procedures based on the Highway Capacity Manual (HCM) have been the industry standard for many years. Recent advances in computing technology, however, have made simulation models more practical than they previously were. Traffic analysis tools based on HCM procedures can quickly predict capacity, speed, delay, and queuing on isolated, small-scale transportation facilities. HCM-based tools are less effective when applied to over-saturated conditions (5). Some of the existing HCM-based tools include:

- SIDRA (Signalized Intersection Design and Research Aid) (17),
- Canadian Capacity Guide (CCG)/Calc 2 (18),
- Highway Capacity Software (HCS)+ (19), and
- TRAFFIX™ (20).

2.1.4 Traffic Signal Optimization Tools

Traffic signal optimization tools are intended to optimize signal-phasing and timing plans. Traffic signal optimization tools can be applied to single intersections, arterial streets, signal networks, and ramp metering rates (5). Traffic signal optimization tools are not designed to analyze major highway or transit improvements or changes in network geometry. Some of the existing traffic signal optimization tools include:

- Synchro (21),
- Progression Analysis and Signal System Evaluation Routine (PASSER) II-02 (22),
- TRANSYT (TRAffic Network StudY Tool)-7F (23), and
- TSDWIN (Time Space Diagram WINdow) (24).

2.1.5 Traffic Simulation Models

Traffic simulation models have been classified as either macroscopic or microscopic. Some models, called mesoscopic models, combine elements of both. In recent years, a fourth level of detail, called nanoscopic simulation has emerged. This section describes these four categories.

2.1.5.1 Macroscopic Models

In macroscopic models, vehicle movement is governed by the flow-density relationship without tracking individual vehicles (25). The simulation takes place on a section-by-section basis and is based on deterministic relationships of flow, speed, and density in the traffic stream (5). While this can adequately represent reality at a large

scale, macroscopic models make some counterintuitive assumptions. For example, a car exists simultaneously at every point along its route during the entire period (morning peak, mid-day, evening peak, and off-peak) when its trip takes place (26).

Some of the existing macroscopic traffic simulation models include:

- TRAF-CORFLO (CORridor FLOW Model) (27),
- KRONOS (28), and
- SATURN (Simulation and Assignment of Traffic to Urban Road Networks) (29).

2.1.5.2 Mesoscopic Models

Mesoscopic models were developed as a compromise between computationally intensive microscopic models and more efficient macroscopic models. Mesoscopic models are becoming less common as the computing power necessary for microscopic modeling becomes more available. In mesoscopic models, the unit of traffic flow is the individual vehicle, but movement is governed by the average speed on the link (5).

Some of the existing mesoscopic models include:

- CONTRAM (CONTinuous TRaffic Assignment Model) (30),
- DYNAMIT-P (DYNAmic traffic assignment Massachusetts Institute of Technology) (31), and
- DYNASMART-P (DYnamic Network Assignment-Simulation Model for Advanced Roadway Telematics) (32).

2.1.5.3 Nanoscopic Models

Nanosimulation or traffic safety modeling is a relatively new area of simulation which attempts to model drivers' steering behavior and more detailed components of perception-reaction time in order to include traffic safety in the model. In traditional microscopic traffic simulation, cars are programmed to avoid collisions (33).

HUTSIM is a simulation model being developed at Helsinki University of Technology (HUT). HUTSIM developers are currently seeking to incorporate nanoscopic principles, such as lapses in reaction time and errors in response, into the driver-behavior models in HUTSIM (34, 35). Nanosimulation is usually categorized as a sub-category of microsimulation.

2.1.5.4 Microscopic Models

Microscopic computer simulation of traffic was first introduced in 1955, when D. L. Gerlough published his dissertation, "Simulation of Freeway Traffic on a General-purpose Discrete Variable Computer" at the University of California, Los Angeles (33). Microscopic models track individual vehicles, each with its own set of driver and vehicle characteristics. Whereas macro- and mesoscopic models track only the lateral movement of vehicles, microscopic models also examine behavior between lanes of traffic, creating a two-dimensional model (referring to the analysis, not to the animations created in post-processing).

Driver and vehicle characteristics, interactions with the network geometry, and interactions between vehicles are all factors that determine movements (25). These models are driven by car-following, lane-changing, and gap acceptance models (which can be thought of as sub-models). Most microscopic traffic simulation models utilize

variations on the General Motors (GM) model (33), which remains the industry standard today.

Some of the existing microscopic traffic simulation models include:

- SimTraffic (36),
- VISSIM (37),
- AIMSUN (38),
- TEXAS (Traffic EXperimental Analytical Simulation) (39),
- INTEGRATION (40),
- CORSIM (1), and
- Paramics (26).

2.1.6 Common Microscopic Traffic Simulation Models

For many years, CORSIM was the industry standard for microscopic traffic simulation. Recently, VISSIM has gained popularity and now competes with CORSIM as one of the most widely used microscopic traffic simulation packages throughout the United States and Europe. Other commonly used models include Paramics and SimTraffic. This section briefly describes each of these.

2.1.6.1 CORSIM

CORSIM was developed by the FHWA in the mid- 1970s in response to the need for a simulation model with reasonable computer usage requirements that could represent traffic flow in large urban areas containing surface street networks and freeways.

CORSIM combines two models: NETSIM (NETwork SIMulation), which represents

traffic on urban streets and FRESIM (FREeway SIMulation), which represents freeway traffic (1). FRESIM and NETSIM were use slightly different car-following and lane-changing logic, so a vehicle being modeled in CORSIM will behave differently depending on whether is it traveling on a freeway or surface street link.

CORSIM can model various intersection controls, including actuated and pre-timed signals, and most surface geometries and traffic flow conditions. The network geometry is represented by a system of links and nodes, where the links represent roadway segments and nodes represent entry points, intersections, or changes in the roadway (11). CORSIM was developed to model primarily automobile traffic, but can also model bus routes and truck lanes. CORSIM was not designed to represent light rail.

In real life, driver and vehicle characteristics vary with a large degree of randomness. Because of this, CORSIM uses random number seeds to assign driver and vehicle characteristics to each vehicle in the simulation. The stochastic nature of the CORSIM model requires that a simulation be run several times using different sets of random number seeds in order to gain an understanding of network performance (1).

While CORSIM can be used as a stand-alone Dynamic Linked Library (DLL), it is commonly run within the TSIS environment (refer to Section 1.1.1), which includes tools for pre- and post-processing a CORSIM model (1).

2.1.6.2 VISSIM

VISSIM was developed in Germany at the University of Karlsruhe in the early 1970s, but the software was not made commercially available until 1993 (11). Because of its European roots, VISSIM was designed to model a variety of modes, including general traffic, buses, light rail, heavy rail, trucks, pedestrians, and bicyclists.

VISSIM comprises a simulator and a signal state generator. The simulator contains network geometry and generates traffic. As in CORSIM, VISSIM uses links to represent roadway segments. VISSIM does not, however, have the traditional node structure found in CORSIM. The signal state generator contains the signal control logic that can be used to model virtually any control logic, including fixed time, actuated, adaptive, transit signal priority, and ramp metering. The node-less network structure and separate signal state generator both give the user greater flexibility in defining the traffic environment (11).

2.1.6.3 Paramics

The Paramics model was developed in the early 1990s as a collaborative effort between Quadstone (a high-performance software firm) and SIAS (a traffic engineering firm) at the University of Edinburgh (26). The two firms have since separated and now distribute the software as two separate packages. Paramics was first used in the United States in 1999 (41). The Paramics model does not have driver behavior calibration parameters, but rather works from the assumption that appropriate demand data and network geometry will yield correct results regardless of driving characteristics specific to a certain location (26). Driver behavior characteristics can only be modified to match local data using the model's application programming interface (API) (41).

2.1.6.4 SimTraffic

SimTraffic was developed to work hand in hand with the signal optimization program, Synchro, and to provide a user-friendly modeling and visualization alternative to CORSIM (36). While the primary strength of SimTraffic lies in its ability to model

signalized intersections, SimTraffic developers claim that it can be applied to freeways and larger networks as well. SimTraffic was developed by Trafficware and bases its vehicle and driver performance characteristics on the vehicle and driver performance characteristics developed by the FHWA. Thus, the internal logic of the model is very similar to that of CORSIM. As of Version 6, SimTraffic does not simulate transit, ramp metering, on-street parking, or high-occupancy vehicle (HOV) lanes. It can model most network geometries, including limited applications of roundabouts. It also models individual pedestrians, where CORSIM only accounts for pedestrians as delay factors (36).

2.2 Car-following Models

The logic used to determine when and how much a car accelerates or decelerates is crucial to the accuracy of a microscopic simulation model. Most simulation models use variations on the GM model. Although it was developed in the 1950s and 1960s, it has remained the industry standard for describing car-following behavior and continues to be verified by empirical data. A variation on the GM model is the PITT car-following model, which is utilized in FRESIM. Other car-following models that have been developed more recently include the Van Aerde (40) and the University of Delaware (42) models, which will not be discussed here but can be found in the literature. The GM family of models is perceived to be the most commonly used in microscopic traffic simulation models and are, therefore, the focus of this thesis.

2.2.1 Generalized General Motors Model

The first GM model modeled car-following as a stimulus-response process in which the following vehicle attempts to maintain space headway (43). When the speed of a leading vehicle decreases relative to the following vehicle, the following vehicle reacts by decelerating. Conversely, the following vehicle accelerates when the relative speed of the leading vehicle increases. This process can be represented by the first GM model, given in Equation 2-1 (42, 43):

$$\ddot{x}_F = \alpha_F (\dot{x}_L(t) - \dot{x}_F(t)) \quad (2-1)$$

where: \ddot{x}_F = acceleration of the following vehicle,
 \dot{x}_F = speed of the following vehicle,
 \dot{x}_L = speed of the leading vehicle,
 α_F = sensitivity of the following vehicle, and
 t = time.

Equation 2-2 is the generalized form of the GM model, also called the fifth GM model (42, 43).

$$\ddot{x}_F(t+T) = \lambda \frac{[\dot{x}_F(t+T)]^m}{[x_L(t) - x_F(t)]^l} [\dot{x}_L(t) - \dot{x}_F(t)], \quad (2-2)$$

where: T = following driver's perception-reaction time, and
 l, m, λ = constants relating to driver sensitivity.

Varying the values of l and m produces different variations on the GM model.

Kikuchi et al. (42) refer to two of the more common GM models as the Pipes model and

the Gazis' model. The Pipes model set both l and m equal to zero, reducing the generalized GM model to Equation 1, with $\alpha = \lambda$. In the Gazis' model, $m = 0$ and $l = 1$, so that car-following sensitivity varies inversely with the distance between the two vehicles. The Gazis' model is the microscopic equivalent of the macroscopic Greenburg model.

2.2.2 PITT Car-following Model

FRESIM uses the PITT car-following model, which is expressed in terms of desired space headway, shown in Equation 2-3 (4, 44).

$$h_s(t) = L + m + kv_2(t) + bk[v_1(t) - v_2(t)]^2 \quad (2-3)$$

where: $h_s(t)$ = desired space headway at time t ,

L = length of leading vehicle,

m = minimum car-following distance (PITT constant),

k = car-following sensitivity factor for following vehicle,

b = relative sensitivity constant,

$v_1(t)$ = speed of leading vehicle at time t , and

$v_2(t)$ = speed of following vehicle at time t .

Equation 2-3 can be solved for the following vehicle's acceleration, given by Equation 2-4 (44).

$$a = \frac{2(x - y - L - m - v_2(k + T) - bk(v_1(t) - v_2(t))^2)}{T^2 + 2kT} \quad (2-4)$$

where: a = the acceleration of the following vehicle,

T = the duration of the scanning interval,

x = position of the leading vehicle, and
 y = position of the following vehicle.

2.3 Genetic Algorithms

Many traffic simulation models require the user to define characteristics for several different driver types. Since an engineer cannot quantify a priori how courteous or aggressive the drivers on a particular system might be, these driver behavior characteristics must often be reverse engineered based on field data. Some method of optimization must be used to find a set of parameters that best reflects observed characteristics of the system. Genetic algorithms lend themselves well to this task and are accepted as an efficient and robust optimization and search methodology used in a variety of fields.

This section will briefly discuss optimization methods that can provide alternatives to genetic algorithms. Next, basic genetics terminology is introduced to provide a framework for a description of the components of a genetic algorithm. This will be followed by a brief history of the development of genetic algorithms. Finally, the steps to implementing a genetic algorithm, as recommended by the literature, will be presented.

2.3.1 *Alternative Optimization Methods*

Prior to the development of genetic algorithms, commonly used optimization methods fell into three categories: calculus-based, exhaustive searches, and random searches (3).

2.3.1.1 Calculus-based optimization

Many optimization problems can be modeled as either minimization or maximization problems. If the search space or the fitness function is well-understood and can be represented as a continuous function, extrema can be found by finding the roots of the derivatives. If there are many roots, these must be individually examined to determine which of these are global extrema (3).

2.3.1.2 Exhaustive searches

Some optimization problems cannot be represented as continuous functions. In these cases, every possible solution may be evaluated to find the optimal solution. While computers make exhaustive searches more practical than they formerly were, they are still impractical for very large search spaces (3).

2.3.1.3 Random searches

Random searches are similar to exhaustive searches, but they can cover the search space uniformly earlier in the search. They may be able to find an acceptable solution more quickly than an exhaustive search, but are subject to the same limitations when applied to a large search space (3).

Any of these methods may be useful for certain applications but none is robust enough to be applied to every problem (3). All of these methods tend to converge on local, rather than absolute optima. Many problems cannot be solved by any of the above mentioned methods. In order to address these shortcomings, researchers have turned to principles of genetics to develop a more robust optimization method.

2.3.2 *Genetics Terminology*

As the name suggests, genetic algorithms are based on principles of genetics observed in the natural world. While an in-depth understanding of genetics is not required to implement a genetic algorithm, a basic understanding of the terminology can be helpful. Some of the vocabulary that genetic algorithms have borrowed from biology is presented here (45).

- *Chromosome*: long strings of DNA that carry genetic information. Humans have 46 chromosomes. In a genetic algorithm, each member of the population has one chromosome.
- *Crossover*: The process by which copies of chromosomes from each parent exchange genetic material to form the child's chromosomes.
- *Mutation*: The process by which genetic material within a chromosome changes, resulting in characteristics not present in either parent.
- *Fitness*: A subjective measure used by Charles Darwin to describe how likely an organism is to survive and produce offspring.

2.3.3 *Components of a Genetic Algorithm*

The basic genetic algorithm itself can be said to have evolved over the past three decades. In spite of the existence of several different breeds, the following characteristics could be said to define the genetic algorithm species (3, 45).

- *Initial population*: Unlike other optimization methods, genetic algorithms use a population of possible solutions as a starting point, rather than a single initial guess.

- *Encoding*: All variable characteristics of a solution are represented by a chromosome. This is usually a bit string in which numeric data are stored as binary digits, although researchers have found other encoding methodologies more appropriate for categorical or continuous data.
- *Fitness function*: Each possible solution is evaluated in terms of its “fitness,” based on some predetermined function. The algorithm determines which solutions will reproduce based on this fitness function.
- *Crossover*: The individuals chosen to reproduce exchange portions of their chromosomes in order to generate new individuals with different combinations of the characteristics their parents have.
- *Mutation*: In addition to exchanging pieces of the bit string between individuals, portions of a chromosome are also moved around internally in order to preserve diversity in the population.
- *Randomization*: The crossover and mutation operators are performed randomly, based on predefined probabilities. Selection is generally also randomized, depending on the selection method.

2.3.4 *History of the Genetic Algorithm*

Biologists were the first to use early computers to attempt to model genetics and evolution (45). In 1957, the work of A.S. Fraser, published in *Australian Journal of Biological Sciences*, modeled genetic systems using a method very similar to a modern genetic algorithm (46).

The application of adaptation and natural selection to optimization problems was not long in following. John Holland and his students at the University of Michigan began exploring adaptive systems theory in the 1960s (3). The first published application of a genetic algorithm was the 1967 dissertation of one of these students, Bagley. Bagley developed an optimum strategy for the playing hexapawn, which is a simplified version of chess (3, 47).

Meanwhile, other researchers were also exploring the optimization potential of principles of evolution. In Germany, Rechenberg introduced an optimization method he called Evolutionsstrategie (evolution strategies) in 1965 (45). In 1966, Fogel, Owens, and Walsh introduced “evolutionary programming,” a method in which candidate solutions are randomly mutated and the fittest is selected (3).

John Holland set forth the theoretical basis for modern genetic algorithms in 1975 with the publication of his book, *Adaptation in Natural and Artificial Systems* (48). That same year Ken DeJong made the first effort to apply a genetic algorithm to optimizing the parameters of another genetic algorithm (3).

In 1985, David Goldberg, a Civil Engineering student at the University of Michigan published a dissertation in which he applied a genetic algorithm to a pipe optimization problem. Goldberg is largely responsible for popularizing the genetic algorithm and bringing it from academia to the toolbox of practicing engineers and scientists (49).

2.3.5 *Implementing a Genetic Algorithm*

This section presents the steps involved in implementing a genetic algorithm, from deciding whether a genetic algorithm is appropriate to determining when to stop the algorithm. Each of these steps will be presented in the following sections.

2.3.5.1 When is a Genetic Algorithm Appropriate?

Genetic algorithms are most appropriate for problems subject to the limitations of the optimization methods presented in Section 1.3.1 (3). This includes problems with any of the following conditions (45):

- The search space is large,
- The search space is not smooth or unimodal,
- The search space not well understood, or
- The fitness function is noisy (i.e. there is a random term in the function).

One limitation of genetic algorithms is that, while they tend not to converge on local extrema, they have no way of determining whether a solution is the absolute best possible. Rather, they can only compare a solution to others that have been tried. Thus, genetic algorithms are most appropriate in situation where a “good enough” solution is desired, rather than an absolute optimum (45).

2.3.5.2 Selecting Parameters to Vary

While one of the strengths of the genetic algorithm is its ability to handle several optimization parameters, an excessive number of parameters will slow down and generally complicate the process. Haupt and Haupt (49) give the example of designing a

car to optimize gas mileage. In this hypothetical problem, the engineer could vary several possible parameters, such as weight, height, width, shape, color, type of engine, tire pressure, etc. Of these, some, such as weight and shape, will have a great impact on gas mileage. Others, such as color, will have minimal impact and will unnecessarily complicate the problem if they are included in the genetic algorithm. Parameters can be selected based on engineering judgment, or sensitivity analyses (49).

2.3.5.3 Selecting Constraints

Two approaches are available for selecting constraints (the minimum and maximum value for each parameter). The first is to use hard bounds, so the variable is not allowed to have any value outside a specified interval ($0 < x < 10$, for example). Binary encoding facilitates these kinds of hard bounds most naturally. The second approach is to transform the variable using a function with a domain equal to the desired interval ($x = 5 \sin y + 5$, for example). This is more appropriate for continuous variable encoding (49).

2.3.5.4 Selecting a Fitness Function

For some problems, selecting a fitness function may be relatively straightforward. To refer back to the example of optimizing a car's gas mileage, the car which can travel the farthest using the least amount of gas would be the best solution. However, if an engineer wished to design a car with both the greatest possible gas mileage and passenger seating capacity, a more complicated function would be necessary. Theoretically, a fitness function could be designed to include as many MOEs as the programmer wishes. However, the same caveat applies here as to parameter selection, wherein an excessive number of MOEs could unnecessarily complicate the algorithm.

2.3.5.5 Encoding

The most common approaches for representing parameters in a genetic algorithm are binary bit strings and real-value (decimal digit) encoding. Other methods, such as tree-encoding, which have also been developed in the several years, will not be discussed here, but can be found in the literature (45).

Researchers at the University of Michigan used binary encoding for the first genetic algorithms. Binary encoding has become state of the practice and is the basis for much of the research that has been done on genetic algorithms. It has the advantage of preserving schemata, or similarities between different chromosomes (3). Schemata were very important to Holland and his colleagues, but more recent research has suggested that schemata do not play as important a role as previously believed (45). Binary encoding naturally holds the parameter constraints by keeping a fixed number of bits. For example, a string with two bits can have 4 possible values: 00, 01, 10, and 11. These correspond to the decimal numbers 0, 1, 2, and 3. A value greater than three can be represented by adding more digits. This makes binary encoding a very natural way to represent parameters with a finite selection of values.

Some applications lend themselves more naturally to continuous-value encoding (49). While Holland's early theory suggests that real value encoding should not perform as well because of its inability to preserve schemata, empirical evidence suggests that, for many applications, it can perform as well or better (45). Additionally, since most other optimization methods use a continuous, rather than binary encoding methodology, it can allow genetic algorithms to cooperate more easily with other methods to form hybrids.

Continuous parameter representation is most appropriate for a bound, but continuous search space (49).

2.3.5.6 Selection

The advantage of a genetic algorithm over a random search lies in the ability to preserve those characteristics that make one solution more or less fit than another (3). Thus, it is crucial that the most fit individuals are more likely to pass on their characteristics to the next iteration than the least fit. Researchers have proposed several approaches to selecting which chromosomes will be allowed to reproduce.

Fitness proportionate roulette wheel selection is the original and most common selection method. In this method, once the fitness of each individual has been evaluated, the fitness values are calculated as a proportion of the total fitness for the entire population. This proportion is then used as the probability that a particular chromosome will be chosen to reproduce. Some variations on this method are sigma scaling, elitism, and rank proportionate selection (45).

Sigma scaling has been found to curtail premature convergence. In the early iterations, the range of fitness values will be large. As the algorithm begins to converge, this range becomes much smaller. Ideally, a wider range of fitness values should be selected in the early iterations in order to prevent the algorithm from converging on a local maximum. Later on, only the very best solutions should be kept. In sigma scaling, the probability of selection is based on the number of standard deviations away from the mean fitness, rather than absolute fitness value. This forces the algorithm to become more selective in later iterations (45).

While sigma scaling is intended to keep the algorithm from discarding solutions with low fitness values that may develop into better solutions, elitism forces the algorithm to keep the very best solutions. A certain number of best solutions are set aside before the selection operator is called. These elite solutions will always survive to the next iteration and always reproduce until a better solution is found (45).

Rank proportionate scaling is a method of making the selection process less sensitive to large gaps in fitness values. Individuals are ranked according to their fitness and the probability of selection is based on an individual's rank, rather than on absolute fitness. Thus, an individual that is only slightly better than its nearest neighbor is just as likely to be chosen as if it were twice as fit (45).

2.3.5.7 Crossover

Single-point crossover is the simplest form of the crossover operation. Once two individuals have been selected to mate, a random point on the bit string is chosen and the chromosomes exchange the section to the right of that string. Single-point crossover is illustrated in Figure 2-1. This method is often the most appropriate because it is the simplest; however, it has the disadvantage of favoring the bits at the ends of the string to be exchanged. The bits in the middle of the string are less likely to be exchanged (45). When disadvantage of favoring the bits at the end of the chromosome becomes an important issue, it can be overcome by using multiple-point crossover (45). A random number of crossover points are chosen for each chromosome and the bits are exchanged in the sections between these points. Figure 2-2 shows an example of multiple-point crossover with three crossover points.

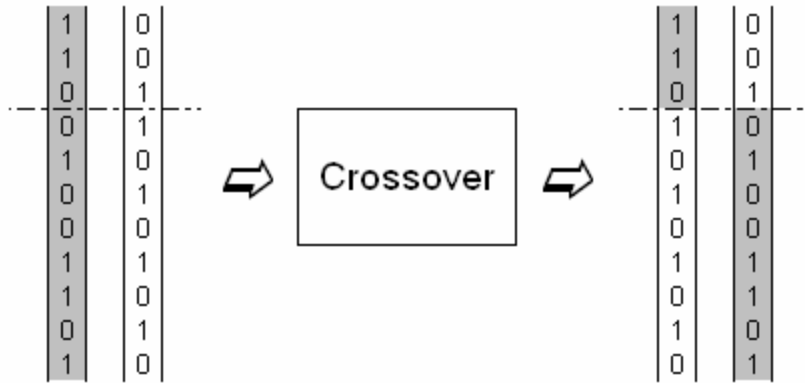


Figure 2-1: Schematic of single-point crossover.

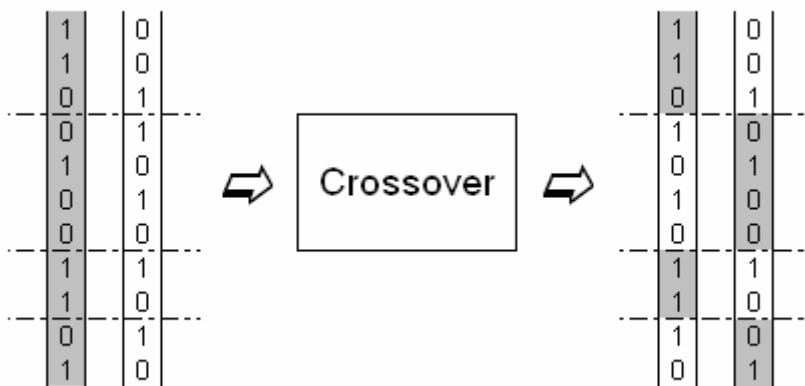


Figure 2-2: Schematic of multiple-point crossover.

2.3.5.8 Mutation

The algorithm maintains diversity in the population by randomly mutating some individuals at each iteration. In bit string chromosomes, a position on the chromosome is randomly chosen and “flipped” (45). DeJong's research points to an optimal probability of mutation of less than 1 percent (3). That is, any one bit in the string might have a

probability ($P = 0.01$) of getting “flipped.” The probability of mutation for an entire chromosome can be expressed by Equation 2-5.

$$P_C = P_B^n, \quad (2-5)$$

where: P_C = probability of mutation for the chromosome,

P_B = probability of a bit getting “flipped,” and

n = the number of bits in the bit string.

Thus, if the probability of mutation is expressed in terms of chromosomes rather than individual bits, DeJong’s research would suggest that P_C be less than 0.3 for a chromosome with a length of 30 bits.

2.3.5.9 Stopping

One of the weaknesses of a genetic algorithm is the inability to determine if a particular solution is the best possible. Stopping criteria define when a solution is “good enough.” Generally, this will either be a predefined minimum fitness value or a maximum number of iterations (3).

2.3.5.10 Setting Genetic Algorithm Parameters

From the previous sections, it is apparent that some parameters for the genetic algorithm itself must be set before the simulation begins. These parameters include:

- Probability of crossover,
- Probability of mutation,
- Population size,

- Minimum fitness for stopping, and
- Maximum number of iterations.

DeJong's research attempted to determine the optimal values for population size, probability of crossover, and probability of mutation on a suite of test cases in 1975 (3). In 1986, Greffenstette proposed that, since the selection of these parameters is an optimization problem, another genetic algorithm could be used to find them. More recent research has focused on self-adapting parameters (50).

2.4 Calibration of Microscopic Traffic Simulation Models

This section summarizes three studies that have utilized genetic algorithms to calibrate driver behavior parameters in CORSIM. Specifically, these studies focused on the lane-changing and car-following parameters that are incorporated into the FRESIM component of CORSIM. These parameters will be discussed in detail in Section 3.1.3.

2.4.1 Singapore Expressway Model

Genetic algorithms were first applied to the calibration of microscopic traffic simulation models in 1998 by Cheu et al. (4). Cheu and his colleagues built a FRESIM model of a 3.7 mile section of the Ayer Rajar Expressway in Singapore. Traffic data for surveillance were collected using video surveillance from pedestrian bridges located at several locations along the section. Two data sets were used in this study, each consisting of two weekdays within the same two week period. One day of data was used for parameter calibration, the other for verification.

Parameter sets were encoded into chromosomes consisting of 40 decimal digits. Fitness calculations only included data that had not been used to determine simulation inputs, in to avoid overestimating the accuracy of the model.

The MOEs selected for calibration were speed and volume. The fitness function selected by Cheu et al (4) is shown in Equation 2-6.

$$F = e^{-AAE(u_1)/s_1} + e^{-AAE(u_2)/s_2} + e^{-AAE(u_3)/s_3} + e^{-AAE(q_2)/s_4} + e^{-AAE(q_3)/s_5} \quad (2-6)$$

where: F = fitness value,

u_i = average speed at detector station i (mph),

q_i = average flowrate at detector station i (vehicles per 30 second),

s_i = scale factor i , and

$AAE(x)$ = Average absolute error, given by Equation 2-7.

$$AAE(x) = \frac{\sum_{t=1}^T |x^{field}(t) - x^{FRESIM}(t)|}{T} \quad (2-7)$$

where: T = number of time steps,

$x^{field}(t)$ = field observation at timestep t , and

$x^{FRESIM}(t)$ = FRESIM calculation at timestep t .

Thus, the algorithm seeks to maximize the fitness value by minimizing average absolute error.

Sensitivity analyses by Cheu et al. (4) showed that a population size of 4, while much lower than what is typically used for genetic algorithms, converged to a higher final fitness value than populations of sizes 10 and 40. Further sensitivity analyses

revealed free flow speed, minimum car-following distance, and car following sensitivity factors to have the most impact on the accuracy of the model. The calibrated parameters for the various data sets suggested that drivers may have different characteristics during each period (morning peak, mid-day, and evening peak) and that different parameters sets may be appropriate for each of these periods.

2.4.2 Covington, Virginia Highway Model

Park and Qi (51) proposed a very thorough procedure for calibrating microscopic traffic simulation models using a genetic algorithm. They applied their procedure to a five-mile highway segment near Covington, Virginia using Paramics, CORSIM, and VISSIM. The proposed procedure placed a greater emphasis on preprocessing to determine whether a genetic algorithm was actually necessary and to ensure that the initial population was somewhat evenly spread over the search space.

In the procedure proposed by Park and Qi (51), the model being calibrated was first run with default parameters. If an acceptable fit to the field data was obtained on this first run, further calibration was unnecessary. Otherwise, an initial distribution was created using Latin Hypercube Design (LHD) sampling (51) to ensure that the entire range of possible parameter values was covered. Because of the stochastic nature of the CORSIM model, multiple runs were performed with each of these parameter sets. A feasibility test was performed in order to verify that the specified parameter ranges reasonably cover the field data. Parameter ranges were deemed acceptable if the field data fell within the middle 90 percent of the simulated data.

Park and Qi (51) used decimal digits to encode values in the chromosomes. The objective function was a measure of error, given by Equation 2-8.

$$e = \frac{|x_o - x_s|}{x_o}, \quad (2-8)$$

where: e = error,
 x_o = field observation, and
 x_s = simulation calculation.

The genetic algorithm sought to minimize error directly rather than transforming it to an exponential maximization problem. Schultz (52) points out that the advantages to such a transformation are that it can provide a better measure of overall effectiveness and a better mathematical representation of the results. The exponential fitness function that is most commonly used (the exponential of the opposite of the error) places a greater emphasis on small reductions in error if the error is already small than if the error is large. After the genetic algorithm was completed, the calibrated parameter sets were validated using field data from the same facility, collected on a different day (51).

Many CORSIM parameters can be defined by distributions. These include amber interval response, gap distribution for left turns, distribution of free-flow speed by driver type, start-up lost time distribution, discharge headway distribution, and car-following sensitivity factors. For parameters that can be defined by distributions, factors within the distributions were not adjusted individually. Rather, Park and Qi defined three to five possible distributions and allowed the algorithm to select one of these (51). This approach has the advantage of simplifying the calibration process because there is only one

parameter to adjust, rather than 10. The disadvantage is that the algorithm does not explore all possible solutions.

The best parameter sets found by the genetic algorithm were found to produce only slight advantages over the best sets found by LHD sampling (used to create the initial population). Because genetic algorithms require much more computing power than LHD sampling, this introduces the question of whether the extra effort is worthwhile (51).

2.4.3 *Houston, Texas Interstate Model*

Schultz and Rilett (53) address the issue of calibrating parameters which can be defined by distributions in their study of car-following parameters in CORSIM. They proposed that the genetic algorithm search for an optimal center and spread (mean and standard deviation) for the distribution. This method has the advantage of significantly reducing the number of parameters to be optimized while maintaining a continuous search space. Schultz and Rilett applied this method to a model of a 13.9-mile section of an interstate in Houston, Texas using both normal and lognormal distributions and found that the algorithm converged to a slightly higher fitness value with the lognormal distribution.

2.5 **Concluding Remarks**

This chapter presented an overview of various categories of traffic analysis tools, with an emphasis on microscopic traffic simulation and car-following models. A brief background on the history and theoretical basis of the genetic algorithm was also

presented, followed by a summary of recent studies which have applied genetic algorithms to the problem of calibrating microscopic traffic simulation models.

3 Calibration Program

The genetic algorithm used for this research is based on one developed and written in the perl language at Texas A&M University and used by Schultz and Rilett (52, 53). This algorithm was developed to calibrate the driver behavior and vehicle performance parameters associated with the FRESIM component of the CORSIM program. Several modifications were made to the original perl script, primarily intended to improve the efficiency of the program, facilitate incorporation into a GUI, and automate several procedures, such as calculating the length of the chromosomes and performing log transformations, which previously had been done manually. A stopping criterion was also added to terminate the algorithm if an acceptable level of error has been achieved.

The CORSIM program uses input files which readily lend themselves to manipulation by a perl script. The CORSIM input files are text files with the extension *.trf. All information required for the program is recorded on lines of text, sometimes referred to as cards. Each record type is identified by a number at the end of the line which indicates what type of data can be found on that card. In order to modify specific CORSIM parameters, the perl script searches for the record type that contains those parameters and replaces that line of text with a line containing new parameters.

This chapter describes the genetic algorithm program and GUI developed for this research. This information is presented to describe the methodology and logic of the calibration procedure. It is also intended as a user's manual for those wishing to apply this program to further research or industry applications.

3.1 Graphical User Interface

The modified version of the calibration program reads a text file containing all necessary inputs, including minimum and maximum values for each parameter to be optimized, and filenames for the CORSIM file to be optimized (*.trf), as well as files containing observed volumes and travel times. This input file also contains several control variables for the genetic algorithm, such as the population size, maximum number of iterations, and mutation and crossover probabilities. An example of an input file is included in Appendix A. The user can edit the input file in its current form directly; however, care must be taken to preserve the format of the file in order to ensure that it will be correctly read by the perl script. To avoid potential problems, the user may edit minimum and maximum parameter values and various control settings from a GUI that will generate the appropriate input file automatically. The interface consists of four dialogs: 1) File Management, 2) Calibration Control, 3) Vehicle Performance Parameters, and 4) Driver Parameters. Each of these is described in the sub-sections that follow.

3.1.1 File Management

The File Management dialog box allows the user to enter the locations of calibration data files on the user's computer. The user must also specify a CORSIM input

file containing the geometry and other characteristics of the network being calibrated, as illustrated in Figure 3-1. The lower portion of the dialog allows the user to specify a directory where all output files will be saved.

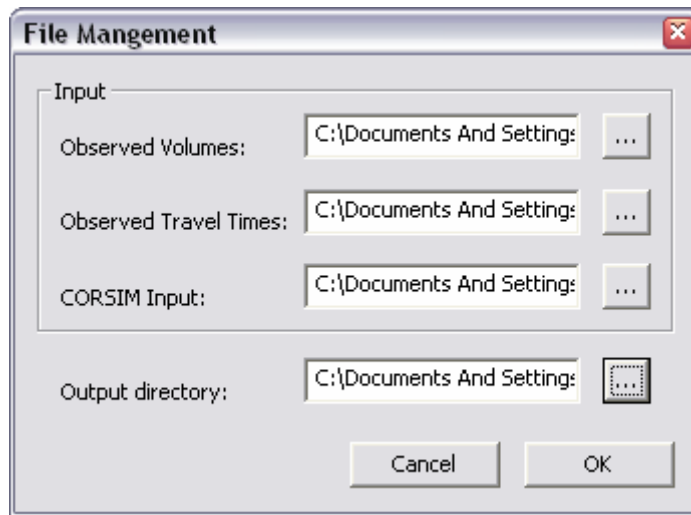


Figure 3-1: File Management dialog box.

The observed volume file lists one volume for each link where actual count data is available. The observed travel time file provides travel time data as either travel time for each individual link or as total travel times over groups of consecutive links. The beginning and ending node of such a link group must be flagged by a node number between 700 and 799. These node numbers must be manually adjusted by the user in the CORSIM model development. In order to optimize driver behavior and vehicle performance parameters, the CORSIM input file must contain three optional record types: 68 (Car-Following Sensitivity Factor), 69 (Pavement Friction Coefficients-Lag to Accelerate and Decelerate), and 70 (Lane Change Parameters, Minimum Separation for Vehicle Generation, Maximum Non-Emergency Deceleration, and HOV Lane Entry

Percentage). The genetic algorithm will not add cards containing calibrated parameters if these cards are not present in the original file. For a detailed discussion of CORSIM record types, refer to the CORSIM User's Guide (1).

At the bottom of the dialog the user can specify where the output files of the genetic algorithm will be written. The output files of the calibration program include a subdirectory for each member of the population. Each of these subdirectories contains three files. These subdirectories are overwritten every iteration and are primarily only used internally by the program. Five to seven different output files are also written (five if the model is being calibrated to volume or travel time, seven if the model is being calibrated to volume and travel time).

The five metadata output files written for a single MOE are named *Play_by_Play.txt*, *OBJECTIVE_output.txt*, *PARAMETER.txt*, *RUN_SITUATION.txt*, and either *volume_logfile.txt* or *time_logfile.txt* (depending on whether the program is calibrating to volume or travel time). If the program is calibrating to both volume and travel time, log files are written for each MOE, as well as an additional file called *both_logfile.txt*. *Play_by_Play.txt* was added to the original program for this research. The formats of the other files have been slightly modified from the version developed at Texas A&M University. These files are now written as tab delimited tables so that they can more easily be imported into spreadsheet programs such as Microsoft Excel. Each output file is briefly described below. Examples of each output file are included in Appendix B.

The *Play_by_Play.txt* file is a detailed log of the genetic algorithm operations taken by the program. The program writes out the chromosome bit strings after every

operation (mutation, crossover, selection) and specifies which chromosomes are being crossed over or mutated. This allows the user to follow how the population has changed over time and which operations have been the most influential in changing the characteristics of the population. Essentially, this file is a narration of a run of the calibration program.

The *OBJECTIVE_output.txt* file lists the minimum, maximum, and average fitness values for each iteration of the genetic algorithm. The user must refer to this file to determine the maximum fitness value that was found by the algorithm and at which iteration this value first occurred.

The *PARAMETER.txt* file lists the parameter sets used for each CORSIM run. The user may refer to this file to see how parameters are changing from iteration to iteration. Once the calibration is complete, the user must refer to this file to find the parameter set that corresponds to the greatest fitness value.

The *RUN_SITUATION.txt* file is a record of the time and date at which each CORSIM run is completed. This can be used to determine approximately how long a typical CORSIM run or iteration of the algorithm has taken. This file is updated after every CORSIM run and can be referred to in order to determine how much time remains before a test is completed.

The *volume_logfile.txt* and *time_logfile.txt* files contain the volume and travel time mean absolute error ratio (MAER) and fitness values for every CORSIM run performed by the program, often in the hundreds. In addition to the format change noted above, these file formats has also been modified to record the fitness values. The *both_logfile.txt* file lists the fitness values for each MOE as well as the combined fitness

value. The log files are updated after each iteration, so it may be referred to at any point during the program run to check the progress of the algorithm. Such checks are advisable early in the first iterations so that errors can be detected. If errors are detected early, the test can be aborted and errors corrected before beginning the test again.

3.1.2 Calibration Control

The Calibration Control dialog allows the user to specify several control variables for the genetic algorithm, as shown in Figure 3-2. The control variables include the maximum number of iterations, population size, mutation and crossover probabilities, acceptable error, and the fitness function parameters. The acceptable error parameter was added to the original algorithm for an additional stopping criterion. If the average error for a run falls below this value before the maximum number of iterations, the algorithm will stop and accept the parameters from that run as a final solution. The acceptable error is entered as a proportion between 0 and 1, where a value of 0 will force the algorithm to always perform the maximum iterations, a value of 0.1 will stop the algorithm when an error of less than ten percent is achieved, and a value of 1 always stop the algorithm after the first iteration.

In the study by Schultz (52), sensitivity analyses were performed to determine appropriate values for several of these control variables. Schultz concluded that mutation and crossover probabilities are not major determinants of overall calibration results. Schultz recommended a population size of at least 20 and at least 30 iterations.

The Calibration Control dialog also allows the user to specify the form of the fitness function. The fitness function employed by the calibration program can contain

three control variables: β values for both volume and travel time and an α value. The fitness function and the significance of these control variables are presented in Section 3.2.1.4.

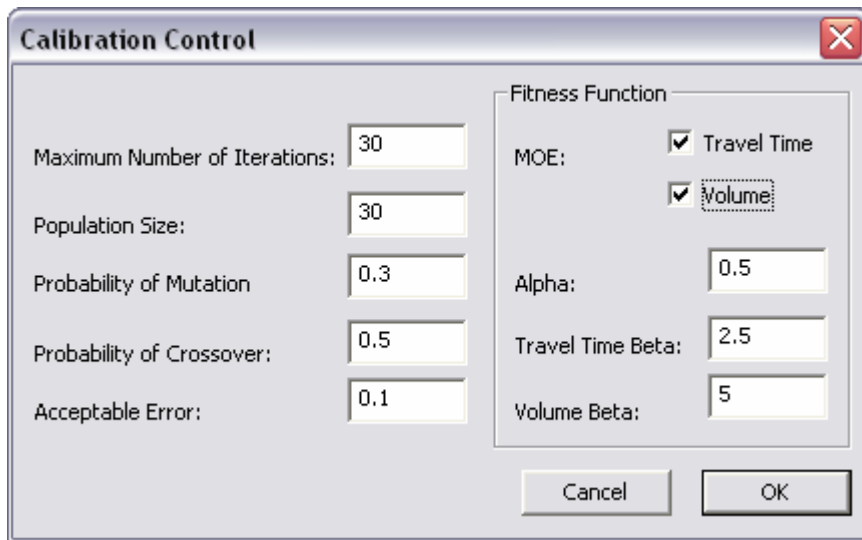


Figure 3-2: Calibration control dialog box.

3.1.3 Driver Behavior Parameters

The Driver Behavior Parameters dialog box is shown in Figure 3-3. The driver behavior parameters include car following sensitivity factors, PITT's car following constant, acceleration and deceleration lag times, time to complete a lane-change maneuver, minimum separation for generation of vehicles, mandatory lane-change gap acceptance, percent of drivers yielding right-of-way to lane-changing vehicles, multiplier for desire to make a discretionary lane-change, and advantage threshold for discretionary lane-change. These parameters are discussed in the following sections.

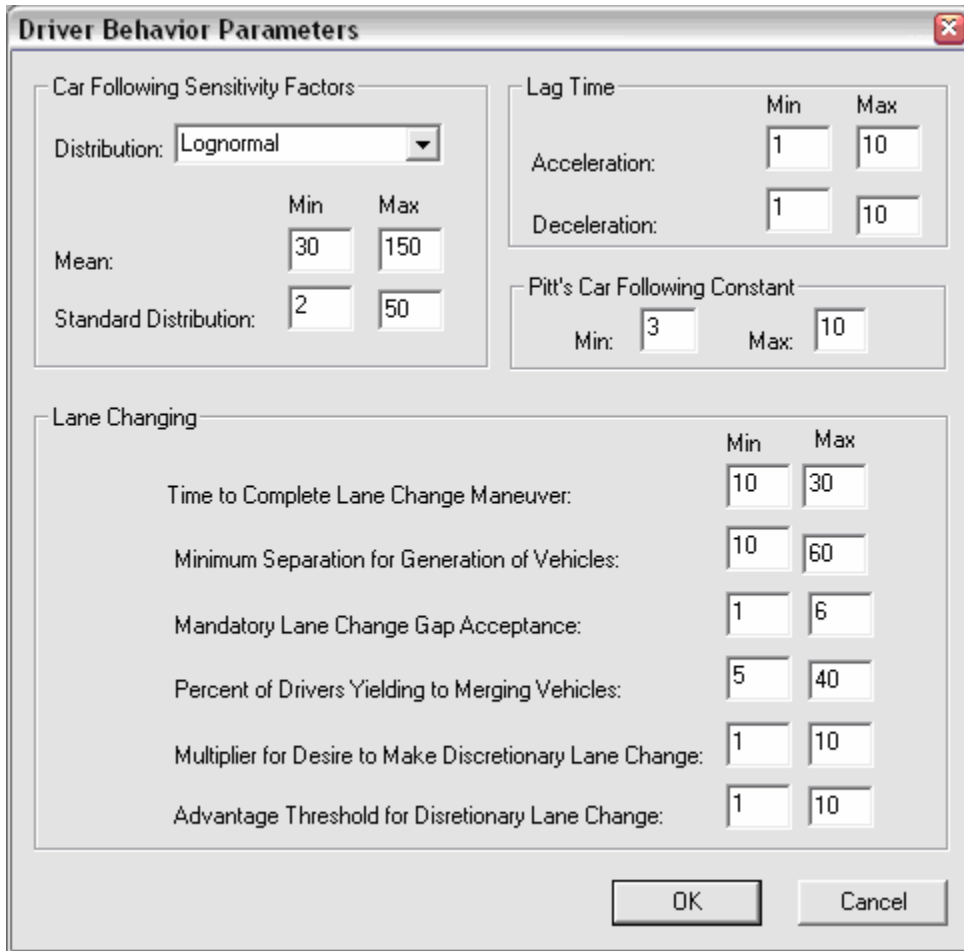


Figure 3-3: Driver Behavior Parameters dialog box.

3.1.3.1 Car-following Sensitivity Factors and PITT Constant

The FRESIM component of CORSIM uses the PITT car-following model (presented in Section 2.2.1.1), expressed previously by Equation 2.3. The user is allowed to define the car-following sensitivity factor, k , for each of 10 different driver types. A driver type with a lower value of k will follow the leading vehicle more closely than a driver type with a higher k value. These values are entered into CORSIM and into the

calibration program in of hundredths of units. Thus, a k value of 1.35 would be entered as 135.

The calibration program gives the user the option of specifying a normal or lognormal distribution for the car-following sensitivity factors and calibrating the mean and standard deviation rather than calibrating an individual factor for each of the 10 driver types, based on research by Schultz (52). If the 10 values are calibrated individually, only one pair of minimum and maximum values may be specified for the entire set.

The PITT car-following constant is defined as the space in feet between vehicles (i.e. gap) that the driver wishes to maintain. The CORSIM default is 10 feet. The CORSIM Users' Manual specifies an acceptable range of values from 3 to 10 feet (*1*). The 10 car-following sensitivity factors and the PITT car-following constant must all be entered on record type 68 in CORSIM. If this record type is not present, the default values shown in Table 3-1 are used.

Table 3-1: Default Car-Following Parameters

Car Following Sensitivity Factors										PITT	
Driver Type	1	2	3	4	5	6	7	8	9	10	Constant
Value	1.25	1.15	1.05	0.95	0.85	0.75	0.65	0.55	0.45	0.35	10 ft.

3.1.3.2 Lag Times for Acceleration and Deceleration

Lag times for acceleration and deceleration are expressed in units of tenths of a second. These parameters are intended to describe the delay motorists experience when

making required movements (*I*). The default values for both of these lag times are 0.3 seconds. Lag times can be specified on record type 69 to override the default values.

3.1.3.3 Lane Changing Parameters

The lane changing parameters in the bottom half of the dialog are written to record type 70 in CORSIM. Lane changing parameters include time to complete the lane-change maneuver, minimum separation for generation of vehicles, mandatory lane change gap acceptance, percent of drives willing to yield right-of way to merging vehicles, multiplier for desire to make a discretionary lane change, and the advantage threshold for making a discretionary lane change. This section discusses each of these.

During lane-changing, the lane-changing vehicle and the putative leading vehicle in the target lane are allowed to operate in unsafe conditions (*I*). These conditions are allowed to exist for the period of time defined by the time to complete lane-change maneuver which is entered in units of tenths of a second. Thus, a value of 30 would indicate that 3 seconds are required to complete a lane-change maneuver.

The minimum separation for generation of vehicles is the minimum rate at which vehicles enter the network in a given lane. While this is not strictly a lane-changing parameter, it affects lane-changing by controlling the supply of available gaps in the network. It is expressed as a time headway in units of tenths of a second.

Mandatory lane change gap acceptance is a unitless measure used to determine the acceptable gap for mandatory lane changes. It can be an integer value from 1 to 6, where 1 indicates the most aggressive drivers (who accept the smallest gap) and 6 indicates the least aggressive drivers (who require the largest gap) (*I*).

Percent of drivers desiring to merge right-of-way to lane changing vehicles represents the fraction of putative followers in the target lane who are willing to decelerate in order to increase the probability of a successful lane change. The program models this by allowing lane changes with a larger value of the lane changing risk factor when the putative follower is a cooperative driver (*I*).

The multiplier for desire to make a discretionary lane change controls how likely drivers are to change lanes when it is perceived to improve their ability to drive at their preferred speed or headway. The multiplier is a value between 0 and 1 and is expressed in tenths of units in CORSIM, and is therefore entered as an integer value between 0 and 10 (*I*).

The advantage threshold for a discretionary lane change controls how advantageous a driver must perceive a lane change maneuver to be in order to change lanes. Like the previous parameter, this is a value between 0 and 1 and is expressed in tenths of units in CORSIM, and is therefore entered as an integer value between 0 and 10 (*I*). Table 3-2 lists the default values for each of the lane changing parameters (*I*).

Table 3-2: Default Values for Lane Changing Parameters

Parameter	Default Value	
	Actual	CORSIM Entry
Minimum Separation for Generation of Vehicles	2 seconds	20
Time to Complete Lane Change Maneuver	1.6 seconds	16
Mandatory Lane Change Gap Acceptance	3	3
Percent of Drivers Yielding to Merging Vehicles	20%	20
Multiplier for Desire to Make Discretionary Lane Change	0.5	5
Advantage Threshold for Discretionary Lane Change	0.4	4

3.1.3.4 Vehicle Performance Parameters

The Vehicle Performance Parameters dialog box allows the user to set minimum and maximum values for the vehicle performance parameters to be calibrated, as shown in Figure 3-4. Maximum non-emergency deceleration indicates the rate at which a vehicle will decelerate in a non-emergency situation. It is expressed in tenths of a foot per second squared. Maximum non-emergency freeway deceleration is specified for each of nine vehicle types. With nine vehicle types and 10 driver types, there are 90 possible driver-vehicle combinations in a CORSIM simulation. The default maximum non-emergency deceleration rate for all nine vehicle types is 80 feet per second squared.

Maximum Non-emergency Freeway Deceleration:			
Vehicle Type 1:	Min:	70	Max: 120
Vehicle Type 2:	Min:	70	Max: 120
Vehicle Type 3:	Min:	40	Max: 100
Vehicle Type 4:	Min:	40	Max: 100
Vehicle Type 5:	Min:	40	Max: 100
Vehicle Type 6:	Min:	40	Max: 100
Vehicle Type 7:	Min:	40	Max: 100
Vehicle Type 8:	Min:	70	Max: 120
Vehicle Type 9:	Min:	70	Max: 120

Figure 3-4: Vehicle Performance Parameters dialog box.

3.2 Genetic Algorithm Implementation

Once the input file has been created, whether manually or by the GUI, it can be input into the calibration program. The program consists of an initialization step followed by many iterations illustrated in Figure 3-5. Each of the steps illustrated in the diagram is described in the sections that follow.

3.2.1 Initialization

During initialization, an initial population is created, the binary bit strings are converted to decimal values, each parameter set is run through CORSIM, a fitness function is evaluated for each chromosome, and the initial population goes through the selection procedure.

3.2.1.1 Creation of Initial Population

First, the program calculates the number of bits necessary to store the range of each parameter being optimized using Equation 3-1.

$$n_i = \text{int} \left[\log_2 \left(\frac{x_{\max} - x_{\min}}{p} + 1 \right) \right] \quad (3-1)$$

where: n_i = bits required for the range of values for parameter i ,

x_{\max} = maximum value for parameter i ,

x_{\min} = minimum value for parameter i , and

p = precision required for parameter i .

The total length of a chromosome is the sum of the required number of bits for each parameter being optimized, as given by Equation 3-2.

$$L_c = \sum_{i=1}^k n_i \quad (3-2)$$

where: L_c = length of the chromosome, and
 k = number of parameters to be optimized.

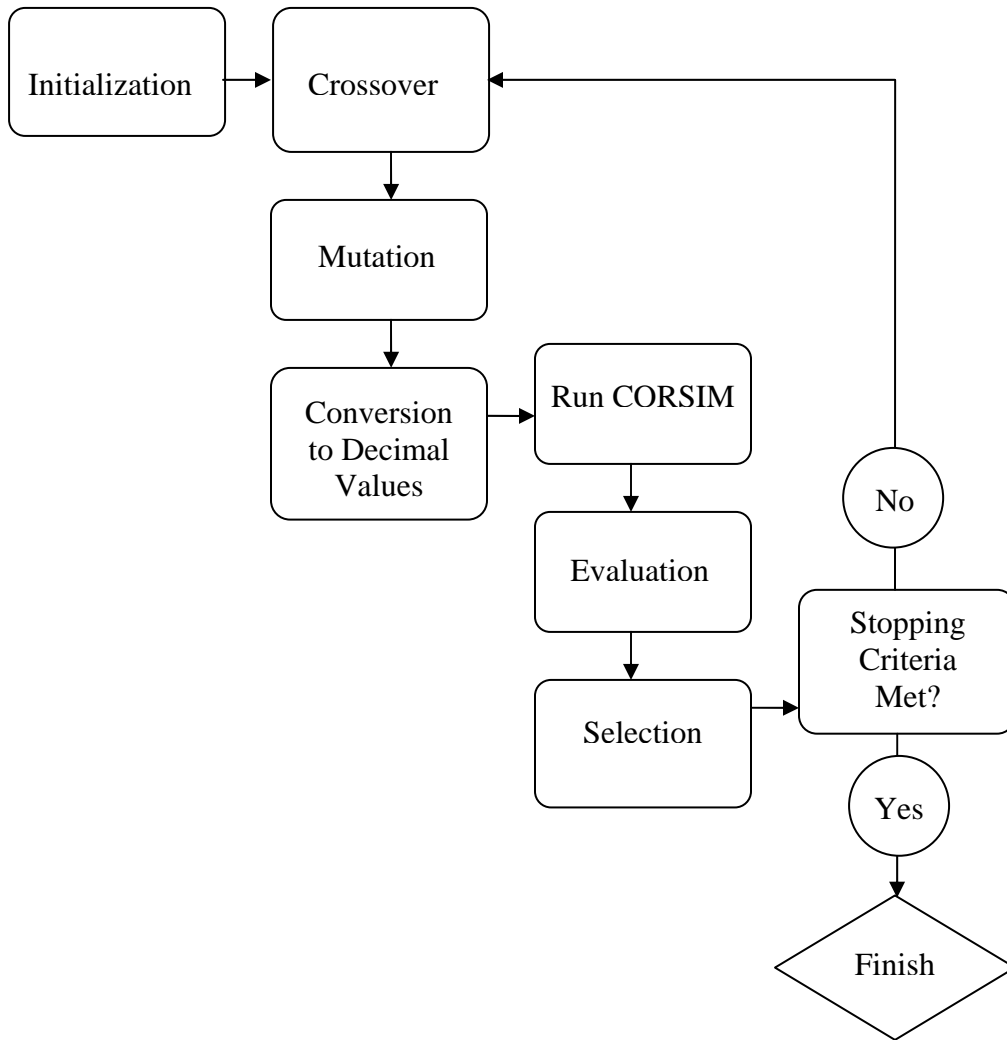


Figure 3-5: Genetic algorithm procedure.

Note that that these chromosomes do not store the actual values of the parameters, but rather a position within the range of possible values. The values stored in the bit string chromosomes will be used together with the minimum and maximum values to calculate actual parameter values, as described in Section 3.2.1.2.

Once the chromosome length has been calculated, an initial population can be created. The population is represented by a randomly generated P by L_c binary array, where P is the population size and L_c is the chromosome length.

3.2.1.2 Conversion to Decimal Values

The initial population is next used to generate parameter sets for the population of size, P . The portion of the chromosome representing each parameter is converted to a decimal value using the Equation 3-3.

$$x = \sum_{j=1}^{n_i} 2^{n_i-j} X_j \quad (3-3)$$

where: x = decimal value of the binary bit string,
 n_i = length of the bit string, and
 X_j = binary value of position j in the bit string.

An actual value for the parameter is calculated using Equation 3-4.

$$R = x_{\min} + \frac{x(x_{\max} - x_{\min})}{2^{n_i} - 1} \quad (3-4)$$

where: R = decimal value of the parameter being optimized.

The car following sensitivity factors for driver types 1 through 10 may be adjusted separately, or a distribution (normal or lognormal) may be specified. In the latter case, the chromosomes, minima, and maxima are not stored. Rather, means and standard deviations are stored and parameter values are calculated from these once they have been converted to decimal values.

3.2.1.3 Initial CORSIM Runs

Each of the parameter sets is used to update a copy of the original CORSIM input (*.trf) file and CORSIM is run on each of these new input files. Once all of these runs are completed, volume and travel time data are read from the CORSIM output files (*.out) and the fitness function is evaluated for each chromosome in the population.

3.2.1.4 Fitness Function

The fitness function used for this genetic algorithm is an exponential function of the MAER for volume, time, or a linear combination of the two exponentials. Thus the fitness function for either volume or time can be expressed Equation 3-5 (52).

$$F = 100e^{(\beta)(MAER)} \quad (3-5)$$

where: F = fitness of a chromosome for a given measure of effectiveness,

β = constant, and

$MAER$ = MAER value for the given MOE, calculated by Equation 3-6.

$$MAER = \frac{1}{N} \sum_{n=1}^N \left(\frac{|x_n^O - x_n^S|}{x_n^O} \right) \quad (3-6)$$

where: N = number of links or link groups (for aggregate MOEs),
 x_n^O = observed value of the MOE on link n , and
 x_n^S = simulation estimate of the value of the MOE on link n .

If the model is to be calibrated for both volume and travel time, the fitness function becomes a weighted average of the two fitness values, where the user specifies a control factor, α , to determine the relative weights of the measures of effectiveness. When α is 0.5, volume and fitness are weighted equally. A value of α less than 0.5 will favor volume in the calibration and a value greater than 0.5 will favor volume. Thus, the total fitness is given by Equation 3-7 (52).

$$F_{TOTAL} = \alpha(F_{TT}) + (1 - \alpha)(F_V) \quad (3-7)$$

where: F_{TOTAL} = total fitness,
 α = control factor,
 F_{TT} = travel time fitness, and
 F_V = volume fitness.

The exponential fitness function makes the genetic algorithm a maximization problem and favors smaller decreases in the MAER value as the magnitude of the fitness value increases.

3.2.1.5 Selection

Once fitness values have been calculated, the population is ready for the selection step of the algorithm. The selection method employed initially is a simple roulette wheel

selection, where the probability that a chromosome, i , will be selected to survive to the next iteration is given by Equation 3-8.

$$P_i = \frac{F_i}{\sum_{i=1}^I F_i} \quad (3-8)$$

where: P_i = the probability that chromosome i will survive,

F_i = fitness value of chromosome i , and

I = population size.

Chromosomes are randomly selected according to their respective probabilities and copied into a new population. The selection process continues until a new population of the same size of the initial population has been created. Because the more fit chromosomes have a higher probability of selection, this new population is likely to contain multiple copies of the most fit chromosomes and no copies of the least fit chromosomes.

3.2.2 Iterations

The steps described above comprise the initialization subroutine of the genetic algorithm. Once the creation, evaluation, and selection of the initial population is complete, the algorithm will iteratively perform the crossover, mutation, evaluation, and selection steps until it converges to a solution, or until a maximum number of iterations has been reached.

3.2.2.1 Crossover

The first iterative operation performed by the genetic algorithm is crossover, which is illustrated previously in Figure 2-2. In the crossover subroutine, each chromosome in the selected population has an equal probability of being selected, copied, and mated with another chromosome to create a new chromosome. This probability is defined by the user before the algorithm begins. When two chromosomes have been selected, a start position and crossover length are randomly selected and a simple two-point crossover is performed. The two new chromosomes created by this operation then replace the first two chromosomes in the existing population that were not created in this crossover subroutine. For example, if six chromosomes are selected for crossover, the operation will be performed three times. The offspring of the two selected chromosomes will replace the chromosomes in the first and second positions of the existing population. The offspring of the next two selected chromosomes will take the third and fourth positions and the offspring of the last two selected chromosomes will take the fifth and sixth positions.

3.2.2.2 Mutation

Once crossover is completed, the chromosomes left in the population that were not replaced by offspring may mutate. Chromosomes are selected for mutation based on a user-defined probability. The first chromosome selected for mutation is copied into the first position in the population not occupied by the offspring of the previous crossover (the seventh position, in the case of the previous example). A randomly chosen bit on this copy is then changed (from a 1 to a 0 or from a 0 to a 1). If further chromosomes are selected for mutation, their mutants will replace the next position in the population after

the previous mutant. Continuing with the previous example, if the first six positions in the population are occupied by the offspring of the crossover operation and three chromosomes are mutated, these mutants will occupy positions seven, eight and nine of the new population.

3.2.2.3 Evaluation

After crossover and mutation, the population contains new chromosomes in the first several positions, while the final positions are occupied by members of the previous iteration. The new members of the population are evaluated in the manner previously described. Binary bit strings are converted to decimal values, which are then used to calculate parameter values. A CORSIM input file (*.trf) is created with each of these new parameter sets and evaluated using CORSIM. Finally, fitness values are calculated for each new chromosome.

Note that additional CORSIM runs are not performed for members of the previous iteration which have survived to the current iteration. Rather, the fitness values that were calculated the first time these chromosomes were introduced to the population are retained for the entire life of the chromosome.

Once the new members of the population have been thus evaluated, the selection subroutine is called and implemented as described in Section 3.2.1.5. The iteration steps are then repeated until a stopping criterion has been met.

3.3 Concluding Remarks

This chapter presents a description of the parameters controlling the calibration program as well as how these parameters are entered into the GUI that was developed for

this research. The specific CORSIM parameters that can be modified by the calibration program were also described. Finally, the operations and underlying logic of the calibration program were described. Chapter 4 describes how data was collected for use as inputs to the calibration program described in this chapter.

4 Data Collection

The output of a microscopic traffic simulation model can only be as accurate as the data used for model input, calibration, and validation. The purpose of this study is to demonstrate a methodology that can be applied to any linear freeway network. The data collected for this study would not necessarily be appropriate for addressing site-specific design and analysis, but they do provide a basis for a model of a hypothetical network for demonstration purposes.

This chapter describes the sources of the data used for this study as well as the methodologies employed to collect and reduce the data. The location of the network being simulated is described first, followed by the procedures used to obtain geometry, volumes, and travel time data.

4.1 Test Site

The Wasatch Front comprises Weber, Davis, Salt Lake, and Utah Counties and is home to about three quarters of the population of the State of Utah (54). As the population between the Provo and Salt Lake City metropolitan areas increases, so will the traffic along the Interstate 15 (I-15) corridor as new residents commute to their destinations. The population in Utah County is projected to increase by 87 percent by the year 2030. During this same period, employment Salt Lake and Utah Counties is

expected to increase by 70 percent and housing units in Utah County by 99 percent. The I-15 corridor is currently operating near capacity and demand is projected to exceed capacity over almost all of I-15 by 2030 (55).

I-15 is the only major arterial connecting the Provo and Salt Lake City metropolitan areas. Construction of additional arterials to relieve congestion on I-15 is inhibited by the geography of the corridor, which is bound by the Uinta Mountains to the East and Utah Lake to the West.

The calibration methodology presented in this research was tested on a hypothetical network based on a 13-mile section of I-15, running through the cities of Orem, American Fork, Pleasant Grove, and Lehi, Utah, shown in Figure 4-1. This portion of the Utah County I-15 corridor is currently under study by UDOT for a Draft Environmental Impact Statement exploring alternatives to mitigate congestion (55). This section was chosen for this research because of the availability of data resulting from the priority UDOT has placed on the corridor. Additionally, its proximity to Brigham Young University made it possible to collect additional calibration data, as needed.

4.2 Data Collection and Reduction

The required data for any simulation model include inputs to the model, calibration data, and validation data. In order to ensure accurate evaluation of the model, it is important that these data sets are not correlated. For example, if the model is being calibrated to volume data, the calibration data should not be calculated from the model input data, or vice versa. One reason why it would not be appropriate to draw site-specific conclusions from this study is that this is exactly what was done, as will be

discussed in Section 4.2.2. For this study, model inputs included geometry and volume data. Volumes and travel time were used for calibration and validation.



Figure 4-1: Location of the I-15 test site.

4.2.1 Geometry Data

Geometry data for the test networks were taken from a CORSIM file provided by UDOT which was created by Wilbur Smith Associates in January 2002. This file was

modified to include only FRESIM elements (i.e., only on-ramps, off-ramps, and mainline freeway links). The original geometry file was subdivided into two separate networks, northbound and southbound. The northbound network was used for calibration and the southbound network for validation.

4.2.2 Volume Data

Volume data was obtained from counts recorded by UDOT on February 13, 2006. The count data included turning movements at all on- and off-ramps in the corridor. The complete count data are included in Appendix C. These were used as input entry volumes for the CORSIM model. The mainline entry volumes for both the northbound and southbound networks were calculated from published values of average annual daily traffic (AADT) (56). These AADTs were converted to directional hourly volumes using the directional (D) and peak hour (K) factors published by UDOT (57). These calculations are summarized in Table 4-1. Intermediate and exit mainline volumes were computed by adding and subtracting ramp volumes from the mainline entry volumes. Figure 4-2 is a schematic diagram of the network showing these network counts.

Table 4-1: Calculation of Mainline Entry Volumes

Location	AADT	K	D (Southbound)	Peak Hour Volume = AADT × K × D	
				Northbound	Southbound
South of Exit 269	99,830	8.6%	0.51	4,379	
North of Exit 284	123,605	8.6%	0.51		5,421

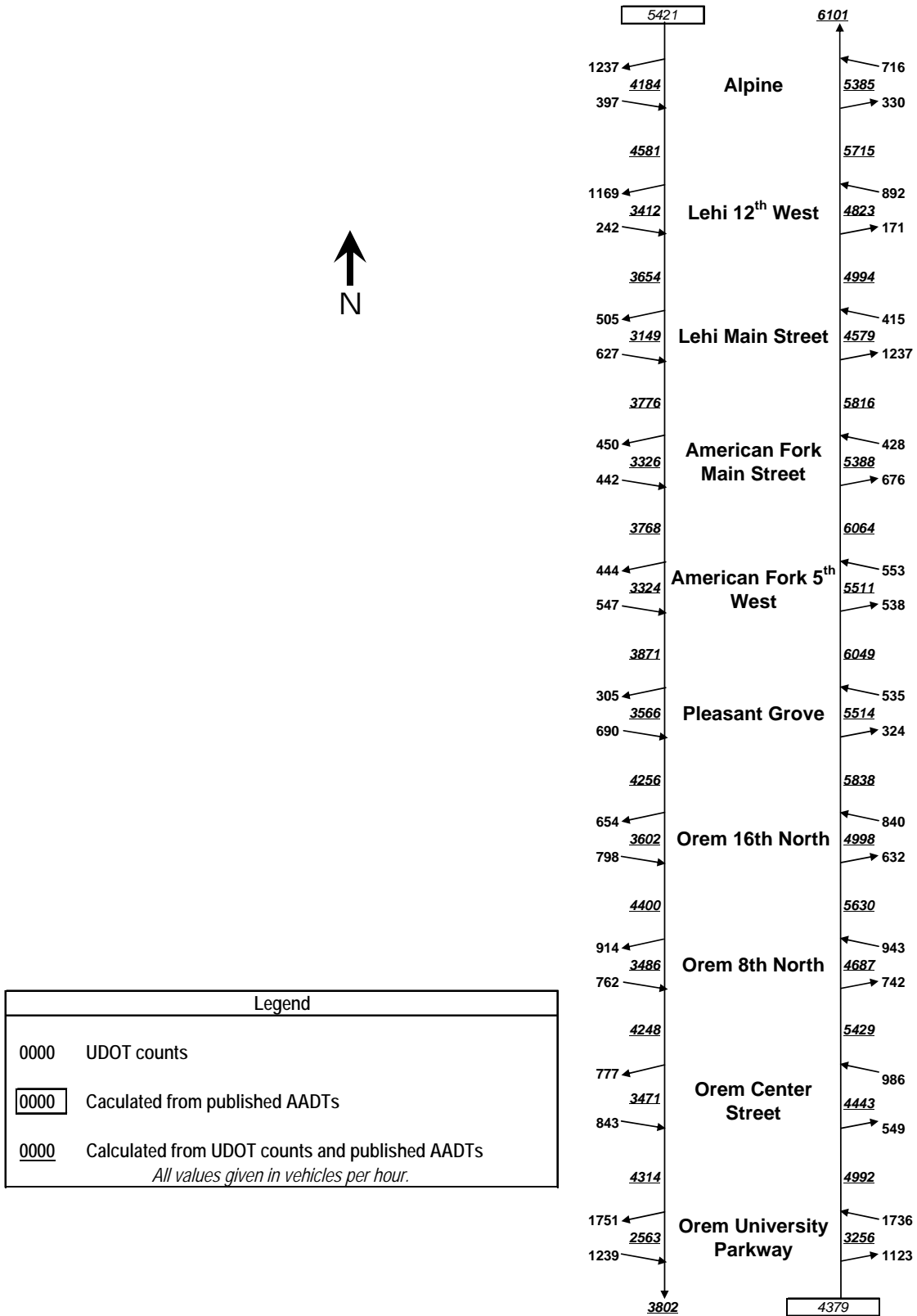


Figure 4-2: Observed ramp volumes with calculated mainline volumes.

4.2.3 *Travel Time Data*

This sub-section describes the methods used to collect travel time data in the field and reduced to obtain speed profiles and travel times.

4.2.3.1 Field Data Collection

Travel time data was collected using test cars. The average-car technique, as described by Roess et al. (58), was applied. This method was chosen because it minimized the dangers that would be associated with attempting to pass specific numbers of vehicles, as in the floating-car technique, or driving as fast as possible, as in the maximum car technique. A driver using the average car technique is instructed to drive at the approximate speed of the traffic stream (58).

Three vehicles were used as test cars and are referred to here as Vehicles 1, 2, and 3. These vehicles were a 1997 Nissan Altima, a 2004 Jeep Grand Cherokee, and a 2001 Saturn SL1, respectively. Vehicles 2 and 3 each made one northbound run and one southbound run. Vehicle 1 made three runs in each direction, for a total of five runs in each direction. A driver and a passenger were assigned to each vehicle and given a global positioning system (GPS) unit. The passenger was instructed to use the GPS unit to record the vehicle's position approximately once every second. Because of variations in signal strength, weather, and the attentiveness of the person operating the GPS unit, data was not successfully recorded for the entire length of every run.

4.2.3.2 Data Reduction

The latitude and longitude coordinates recorded by the GPS units were converted to Universal Transverse Mercator (UTM) coordinates using the spreadsheet developed by

Professor Steve Dutch at the University of Wisconsin at Green Bay (59). The calculations in this spreadsheet use the formulae given by Snyder (60). Once the coordinates had been converted to UTM coordinates, the approximate distance between each point was calculated using the distance formula given in Equation 4-1.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4-1)$$

where: d = distance between the two points,

x_i = easting of point i , and

y_i = northing of point i .

The approximate speed of the vehicle at each point (excluding the first) was then calculated based on the calculated distance and elapsed time between that point and the previous point. The speeds thus calculated are only rough estimates as the position data collected by the GPS units are accurate to within several meters. Those points with calculated speeds of greater than 150 miles per hour were eliminated from the data set. In order to further improve the reasonableness and smoothness of the speed profiles, the speeds were taken as the average of the previous 20 points. For the first 20 points in a data set, the average of all previous points was used.

4.2.3.3 Results

This subsection presents travel time estimates from the northbound and southbound runs.

Speed profiles were created by plotting the speeds thus calculated versus the distance traveled, as illustrated in Figure 4-3 through Figure 4-7, for runs 1 through 5, respectively. All vehicles appear to have experienced reductions in speed around mile six. The data points were also color-coded for speed plotted on a map of the corridor, as illustrated in Figure 4-8 through Figure 4-12, for runs 1 through 5, respectively. The data points on the speed maps range in color from green to red, with the highest speeds represented by green dots and the lowest speeds by red dots. Vehicle 1 was used for northbound runs 2, 3, and 4 and southbound runs 3, 4, and 5. Vehicle 2 was used for run 1, both northbound and southbound. Vehicle 3 was used for northbound run 5 and southbound run 2. Each data set collected by vehicles 2 and 3 was more complete than any collected by vehicle 1, but vehicle 1 performed more runs.

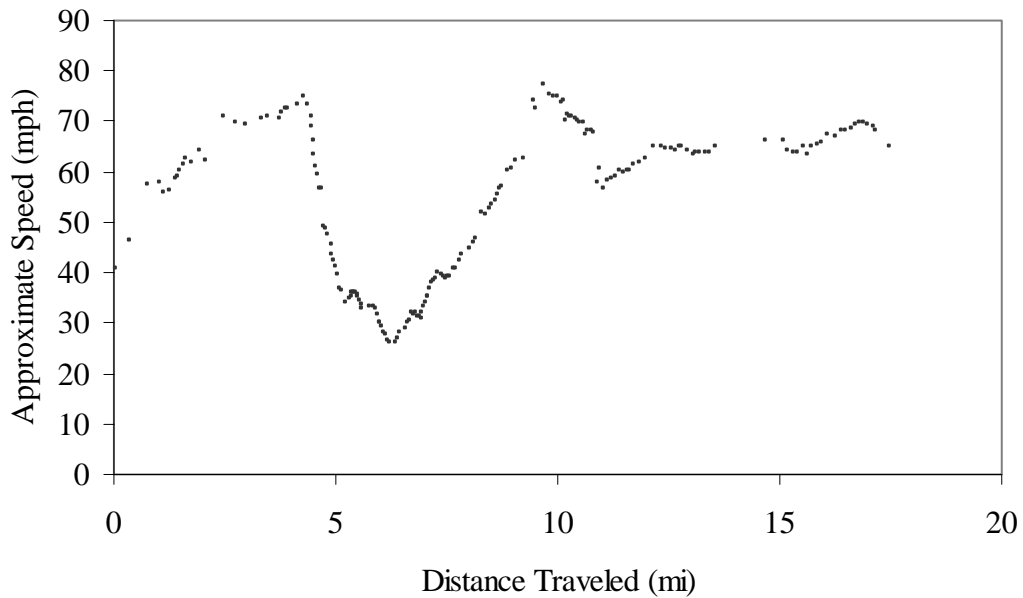


Figure 4-3: Speed profile for northbound run 1.

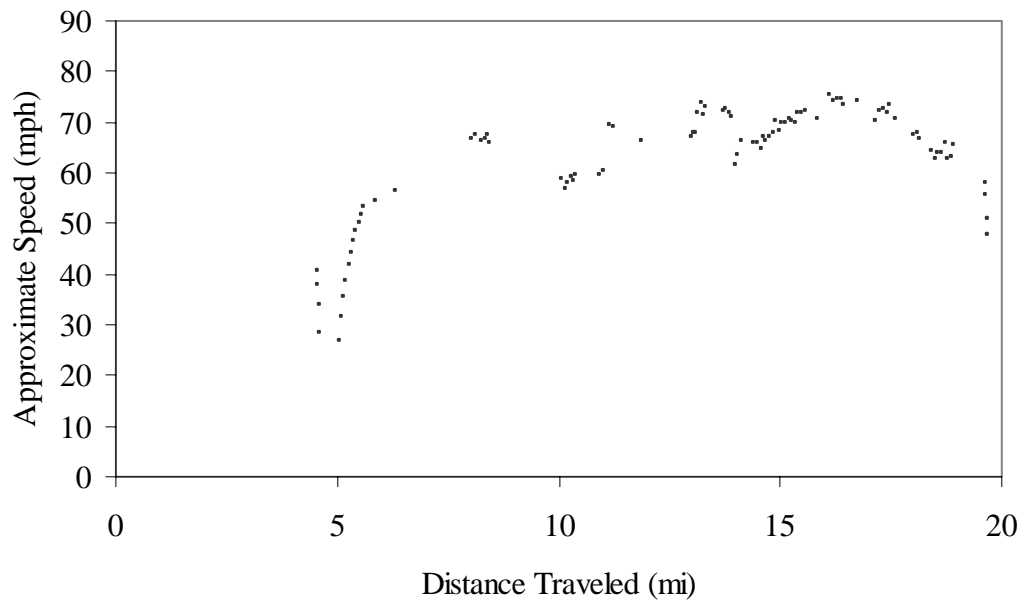


Figure 4-4: Speed profile for northbound run 2.

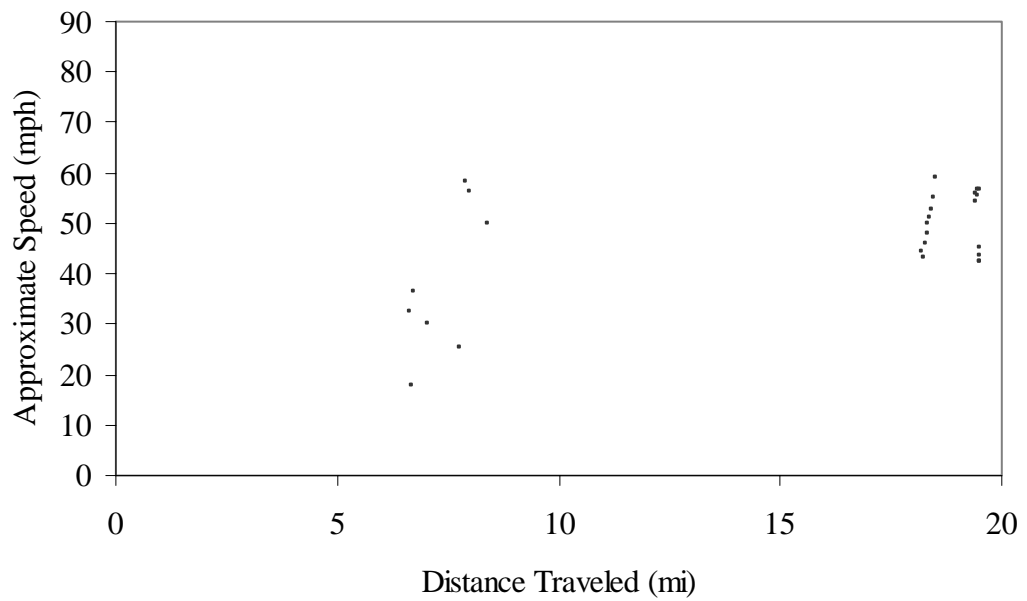


Figure 4-5: Speed profile for northbound run 3.

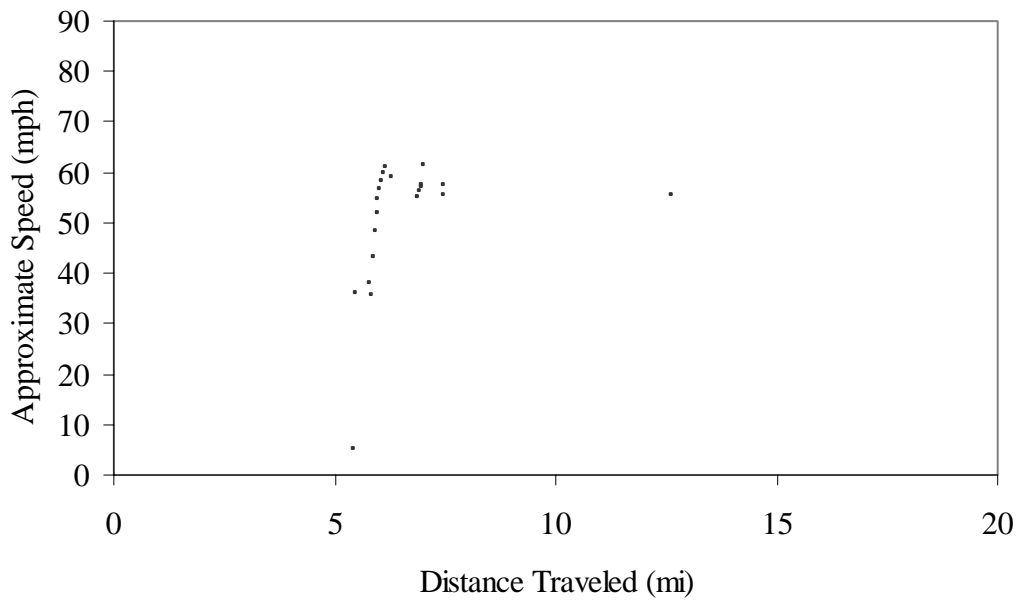


Figure 4-6: Speed profile for northbound run 4.

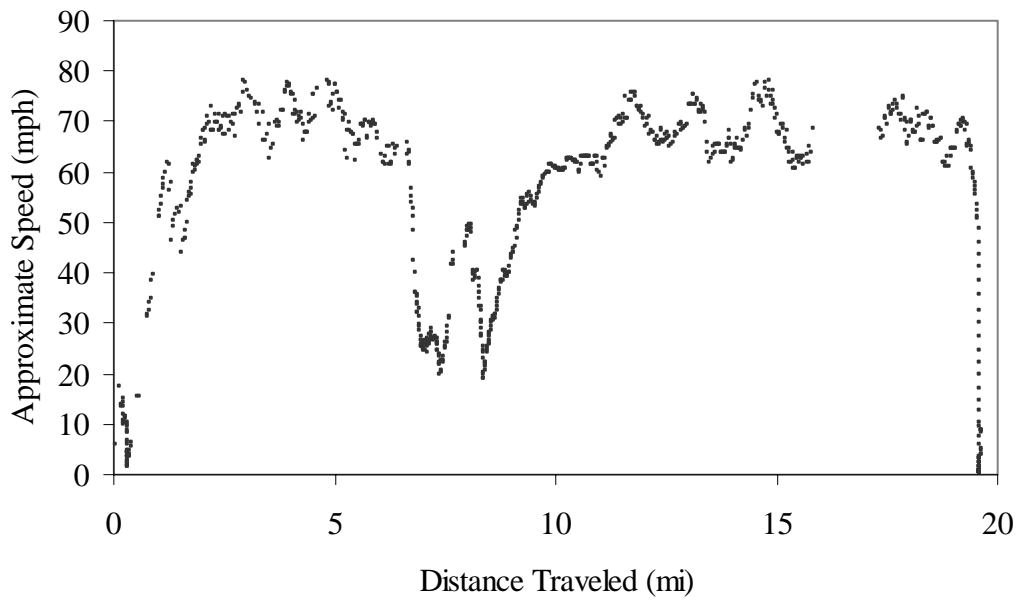


Figure 4-7: Speed profile for northbound run 5.

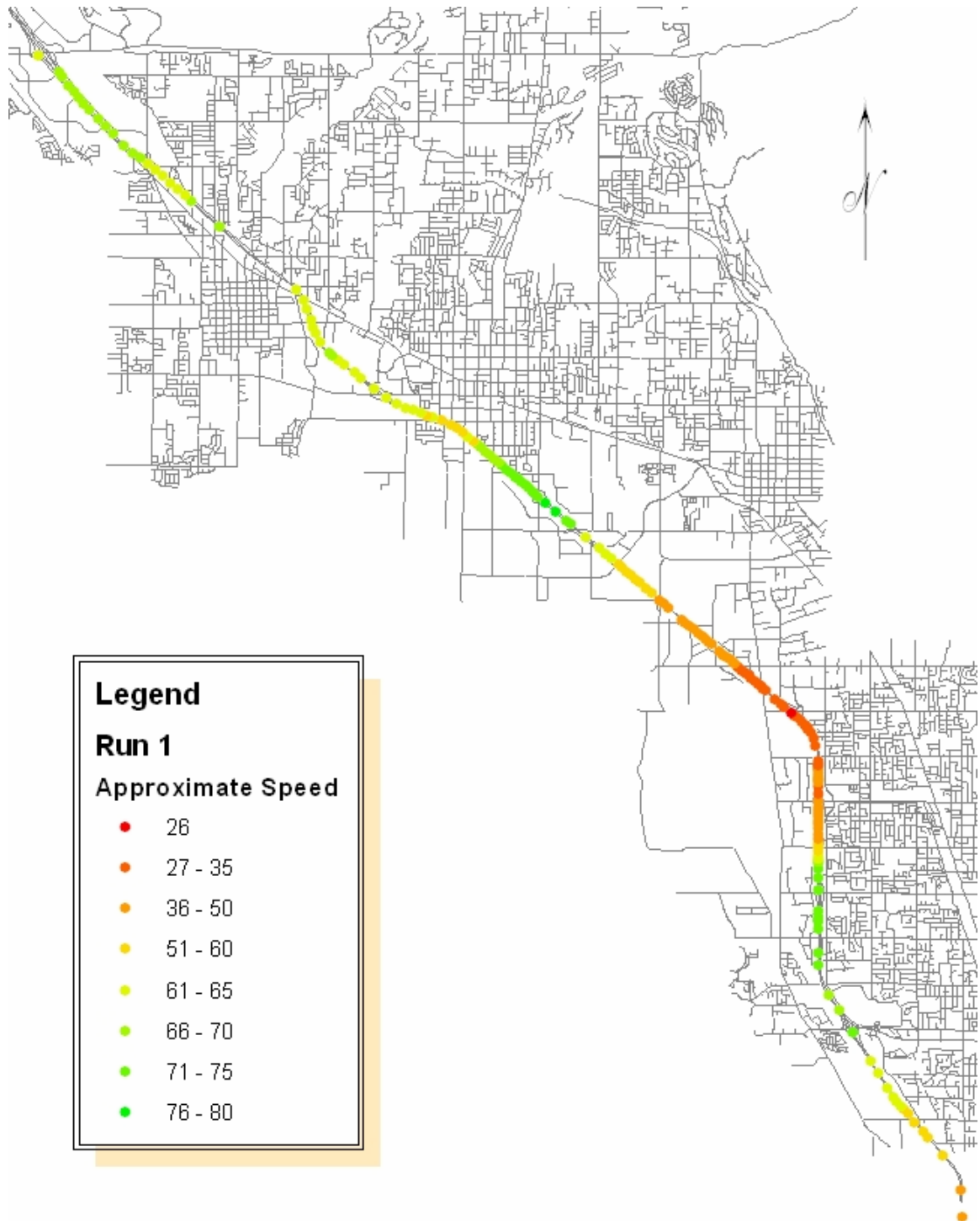


Figure 4-8: Speed map for northbound run 1.

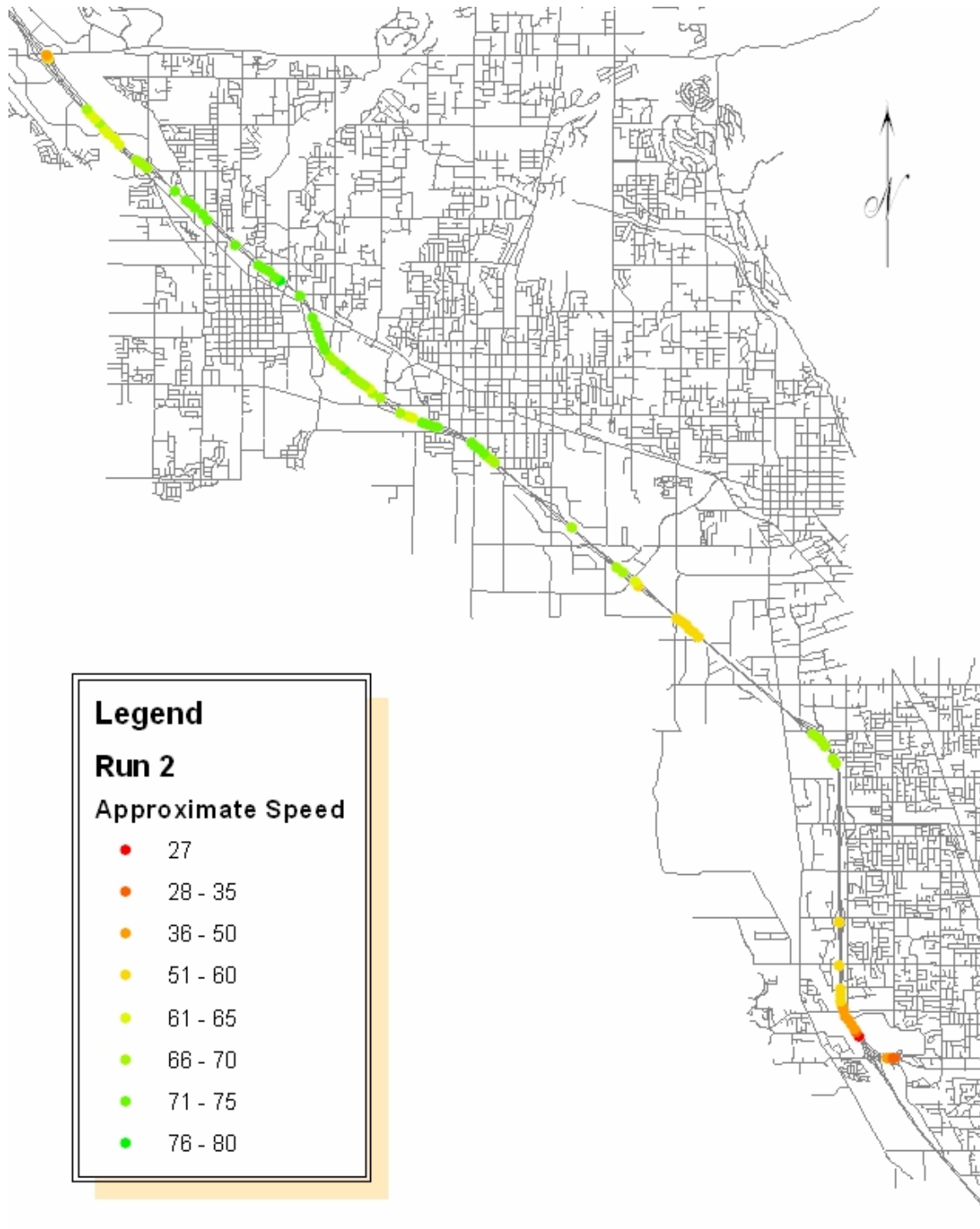


Figure 4-9: Speed map for northbound run 2.

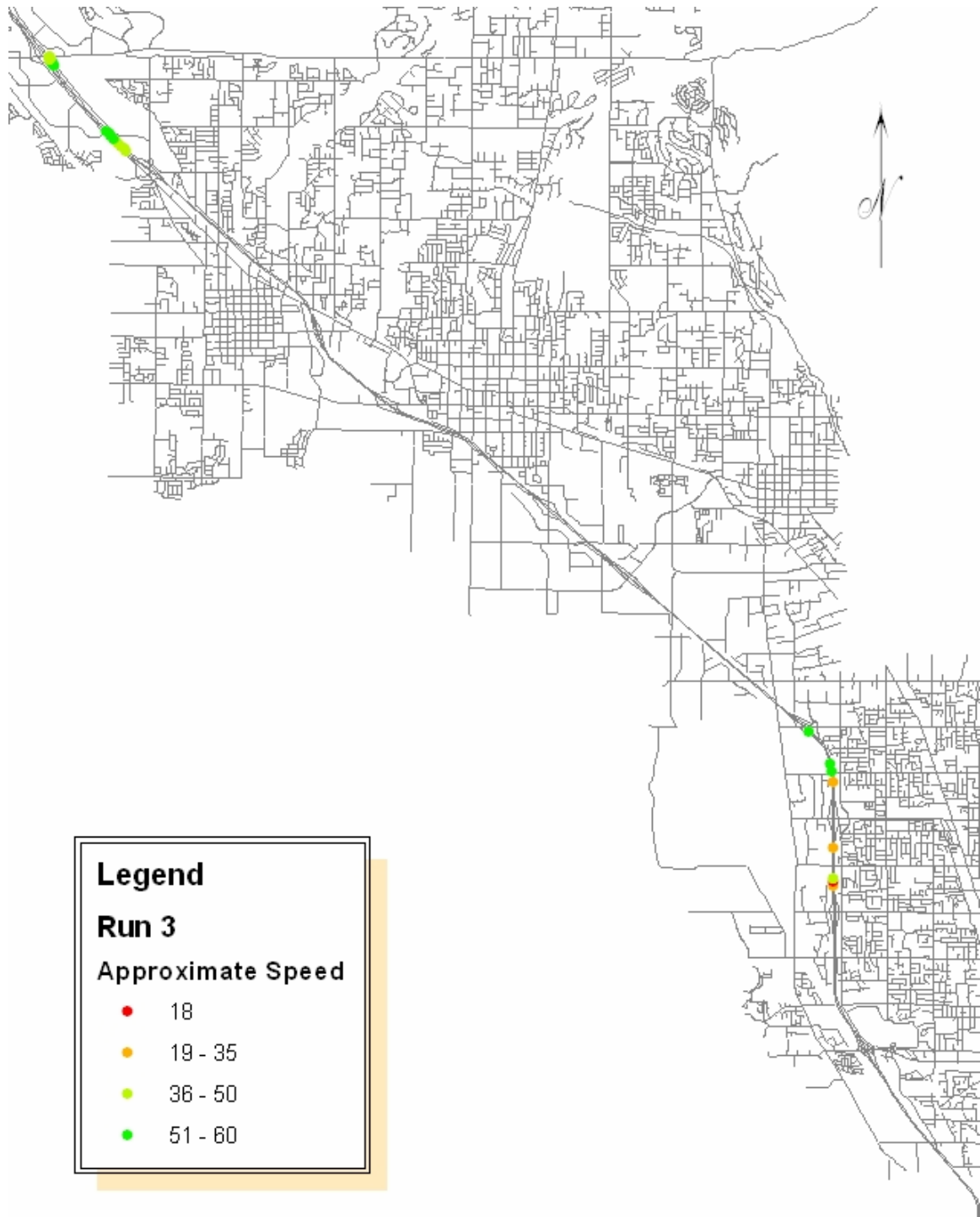


Figure 4-10: Speed map for northbound run 3.

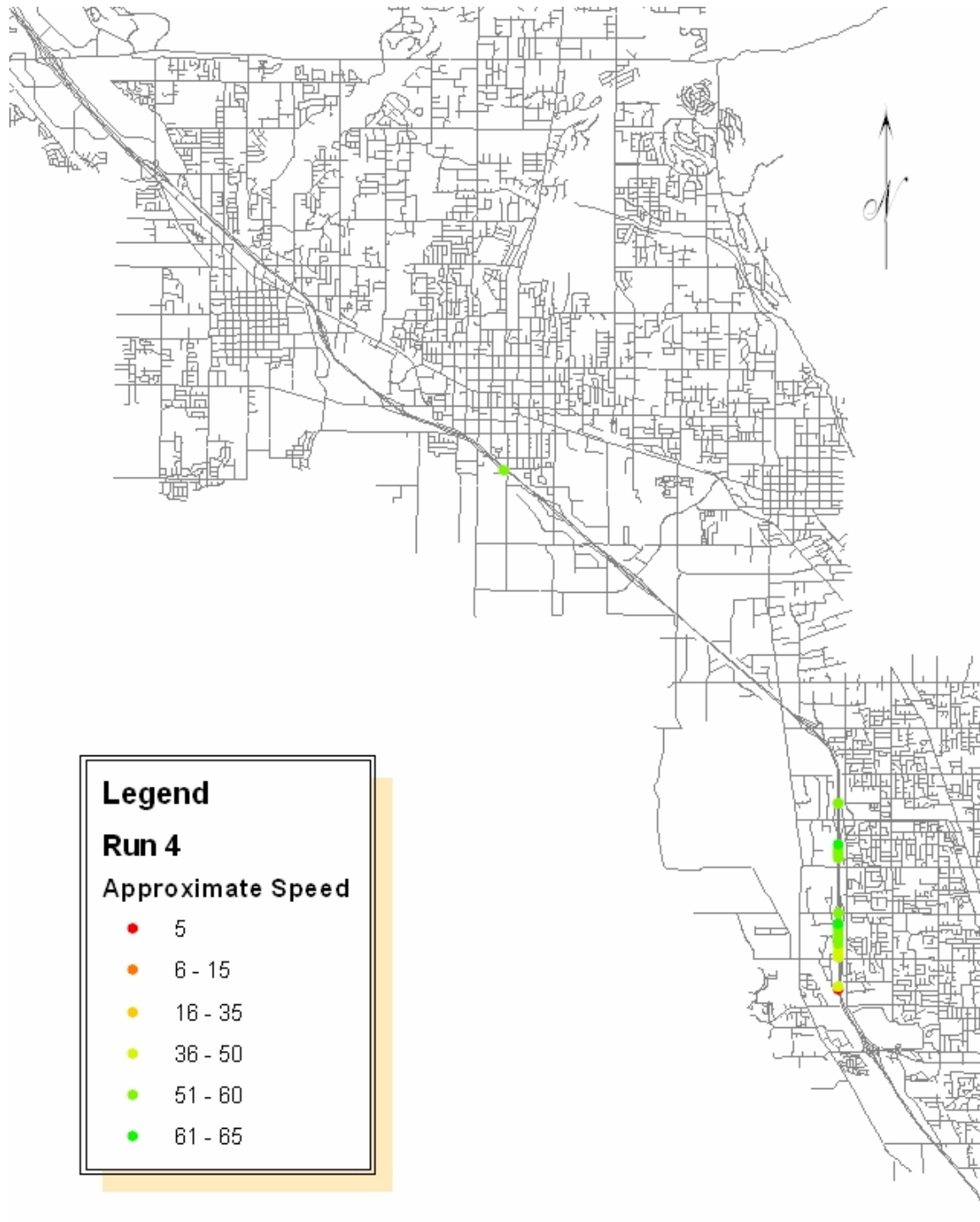


Figure 4-11: Speed map for northbound run 4.

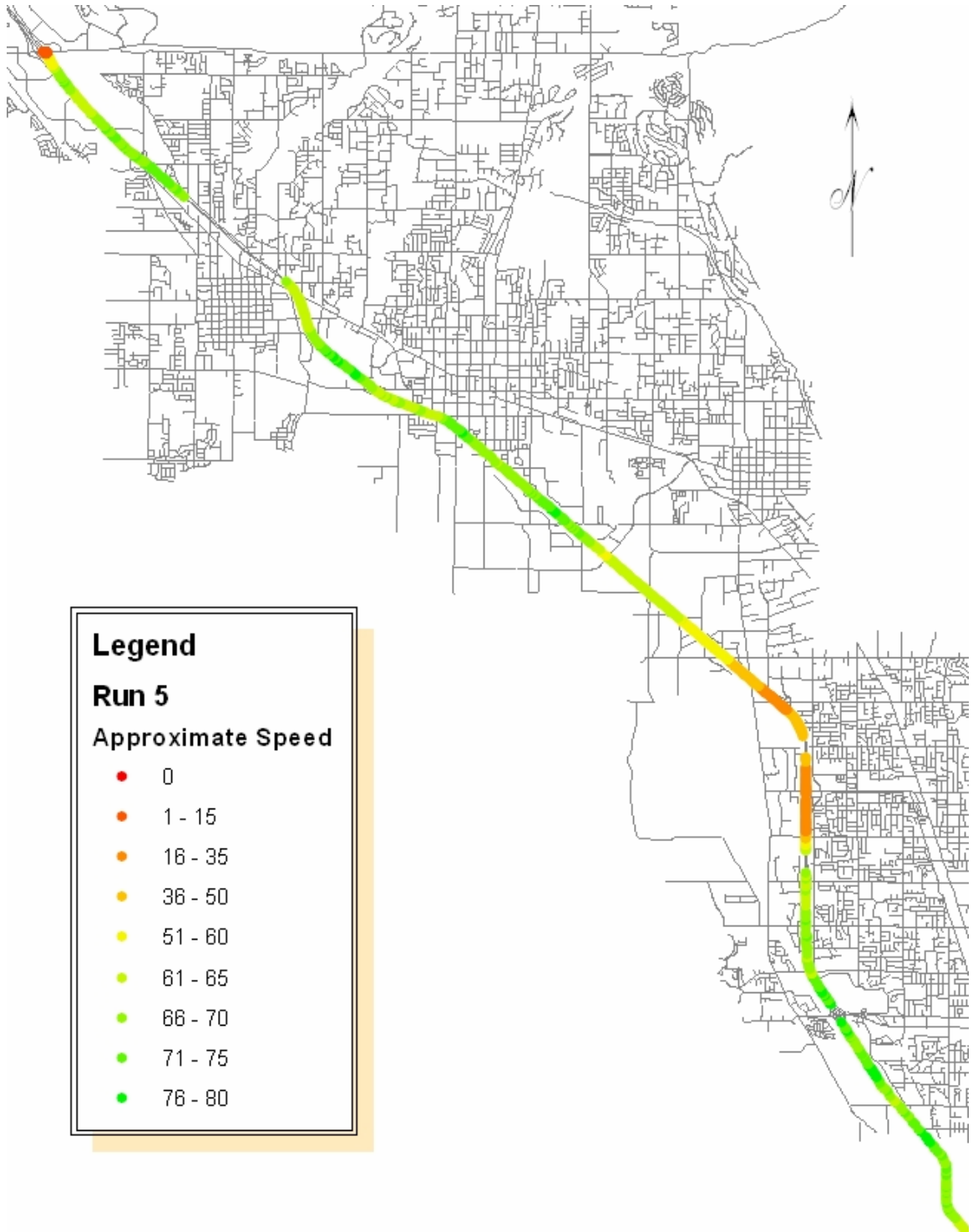


Figure 4-12: Speed map for northbound run 5.

The time recorded for data points collected at intersections of on- and off-ramps with the mainline freeway were used to calculate travel times between those points. Times for all data points collected at all on- and off-ramps are shown in Table 4-2. Blank cells in this table indicate that no data point was recorded close to that location for that run. Table 4-3 shows the travel times that were calculated from the data in Table 4-2. Segments were chosen such that travel times were available from at least three different runs, in order to have the largest possible sample size from which to compute an average.

Table 4-2: Times Recorded at Node Locations for Northbound Runs

Location		Run Number				
		1	2	3	4	5
Exit 269	<i>off</i>	17:28:24				17:20:16
University Parkway, Orem	<i>on</i>	17:28:52	18:33:47			17:20:47
Exit 271	<i>off</i>	17:29:47	18:34:52		17:35:17	17:21:49
Center St., Orem	<i>on</i>	17:30:19				17:22:24
Exit 272	<i>off</i>	17:31:16				
800 North, Orem	<i>on</i>	17:32:24				
Exit 273	<i>off</i>			18:07:50	17:36:03	17:23:10
1600 North, Orem	<i>on</i>			18:08:26	17:36:37	17:24:34
Exit 275	<i>off</i>	17:37:00	18:38:20 *			17:28:30
Pleasant Grove Blvd.	<i>on</i>	17:37:51				17:29:18
Exit 276	<i>off</i>	17:38:08	18:39:20			17:29:36
500 East, American Fork	<i>on</i>	17:38:56				17:30:12
Exit 278	<i>off</i>	17:40:25	18:41:22			17:31:31
Main St., American Fork	<i>on</i>	17:40:58	18:41:53			17:31:59
Exit 279	<i>off</i>	17:41:35	18:42:20			17:32:36
Main St., Lehi	<i>on</i>	17:41:58	18:42:43			17:33:00
Exit 282	<i>off</i>	17:43:56	18:44:24			17:34:56
1200 West, Lehi	<i>on</i>	17:44:32	18:45:03	18:17:35		17:35:36
Exit 284	<i>off</i>	17:45:12				17:36:22
Alpine	<i>on</i>					

*Interpolated from points on either side.

Table 4-3: Northbound Travel Times

Segment		Travel Time (seconds)					
Start	End	Run Number			Average	Standard Deviation	Required Runs
		1	2	3			
Exit 269, on	Exit 271, off	55	65	62	60.7	5.1	4
Exit 271, off	Exit 275, off	433	208	401	347.3	121.7	2277
Exit 275, off	Exit 276, off	68	60	66	64.7	4.2	3
Exit 276, off	Exit 278, off	137	122	115	124.7	11.2	19
Exit 278, off	Exit 279, off	70	58	65	64.3	6.0	6
Exit 279, off	Exit 282, on	177	163	180	173.3	9.1	13

The final column of Table 4-3 shows the number of runs that would be required for the computed average to be within five seconds of the actual travel time, with 95 percent confidence. This was computed using Equation 4-2 (58).

$$N \geq \frac{1.96^2 s^2}{e^2} \quad (4-2)$$

where: N = required sample size,
 s = sample standard deviation (seconds), and
 e = tolerance (5 seconds).

Because of the magnitude of the sample standard deviations, more runs would be required to obtain accurate travel time estimates. Additionally, run 2 would likely be removed from the computations because the data were recorded between 6:30 and 6:45 pm, which is slightly later than the evening peak. Hence, the travel time recorded for run 2 between the off-ramp of Exit 271 and the on-ramp of Exit 275 (shown in the second row of Table 4-3) is significantly less than for runs 1 and 3. For this reason, the computed value for the required number of runs is unusually high. In spite of these shortcomings,

the average values computed here are sufficient to demonstrate the calibration methodology. The travel time values shown in Table 4-3 were used as inputs to the calibration program.

Speed profiles were also created for the southbound runs and are shown in Figure 4-13 through Figure 4-17, for runs 1 through 5, respectively. Speed maps for the southbound runs are shown in Figure 4-18 through Figure 4-22, for runs 1 through 5, respectively.

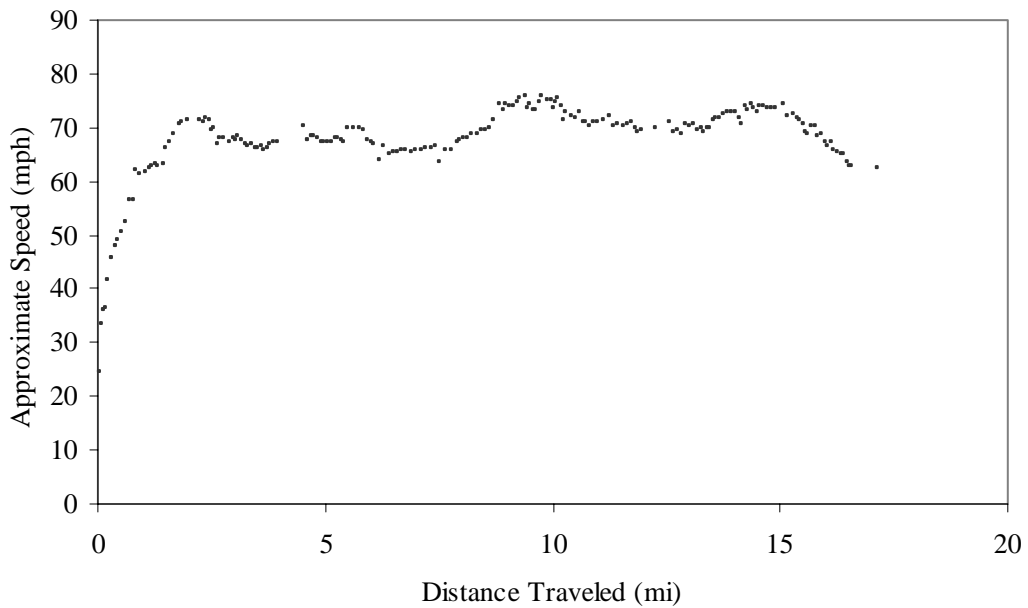


Figure 4-13: Speed profile for southbound run 1.

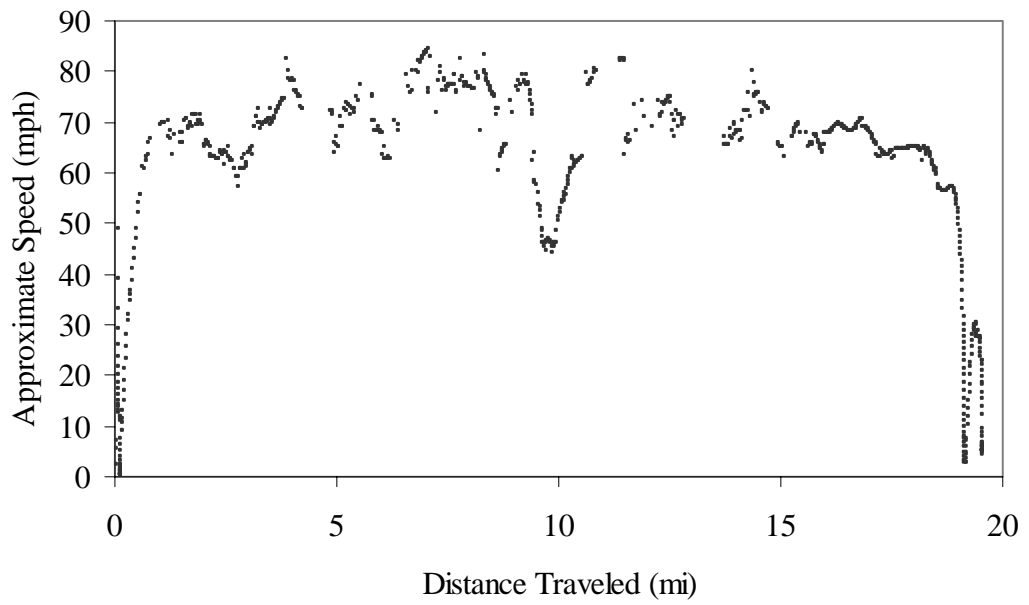


Figure 4-14: Speed profile for southbound run 2.

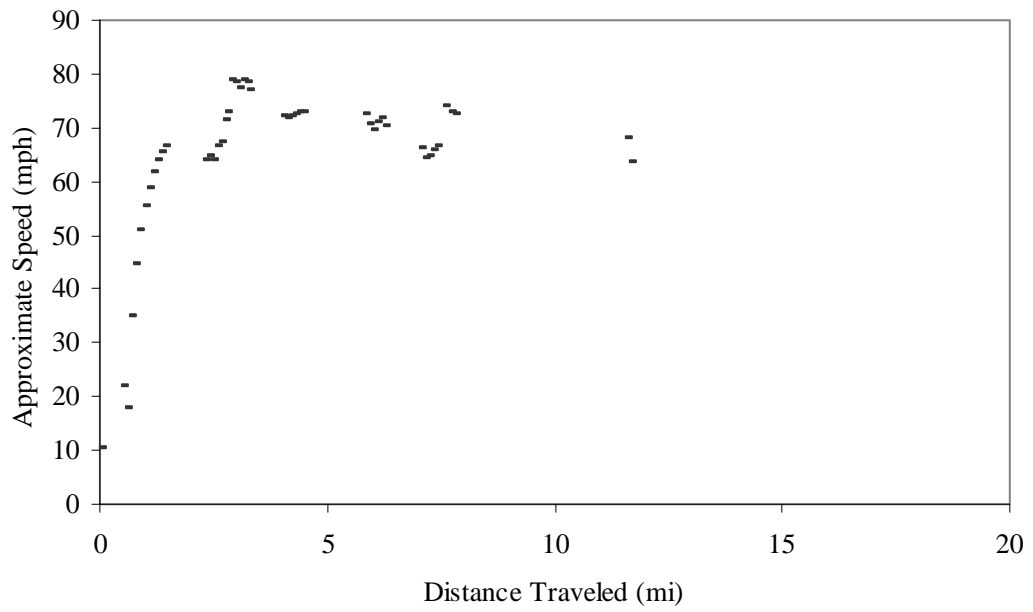


Figure 4-15: Speed profile for southbound run 3.

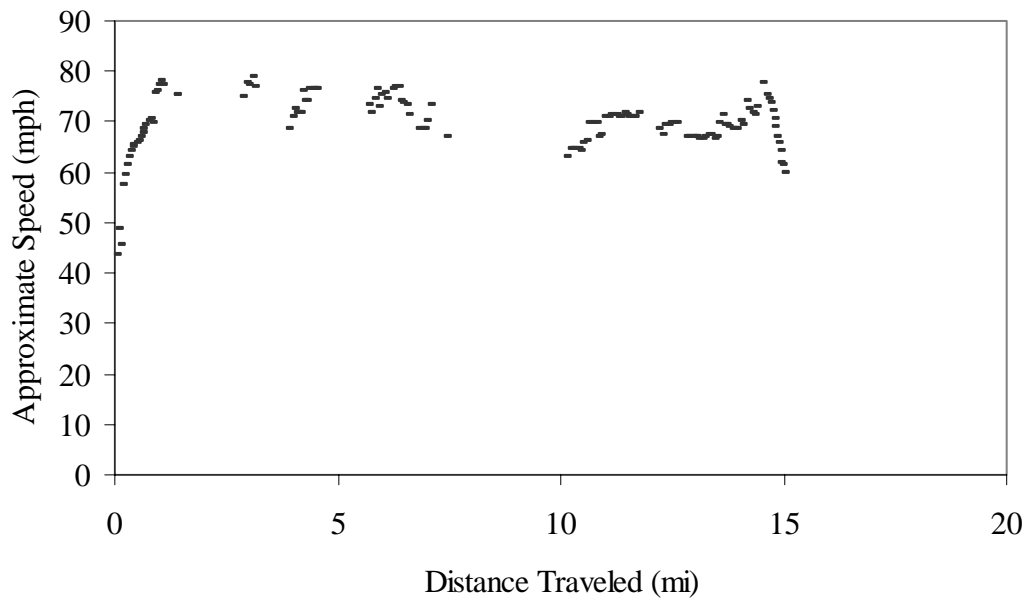


Figure 4-16: Speed profile for southbound run 4.

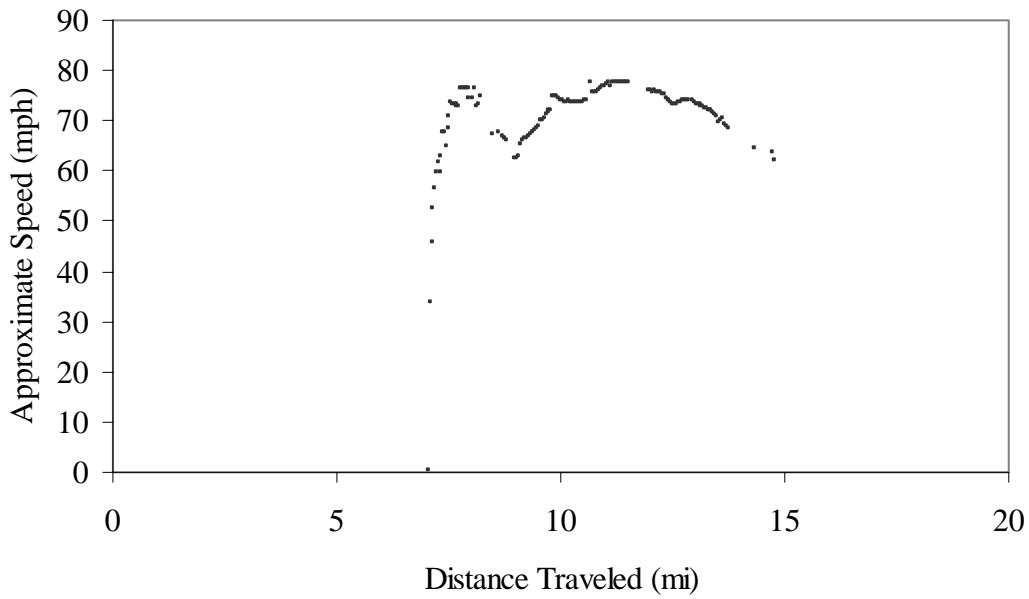


Figure 4-17: Speed profile for southbound run 5.

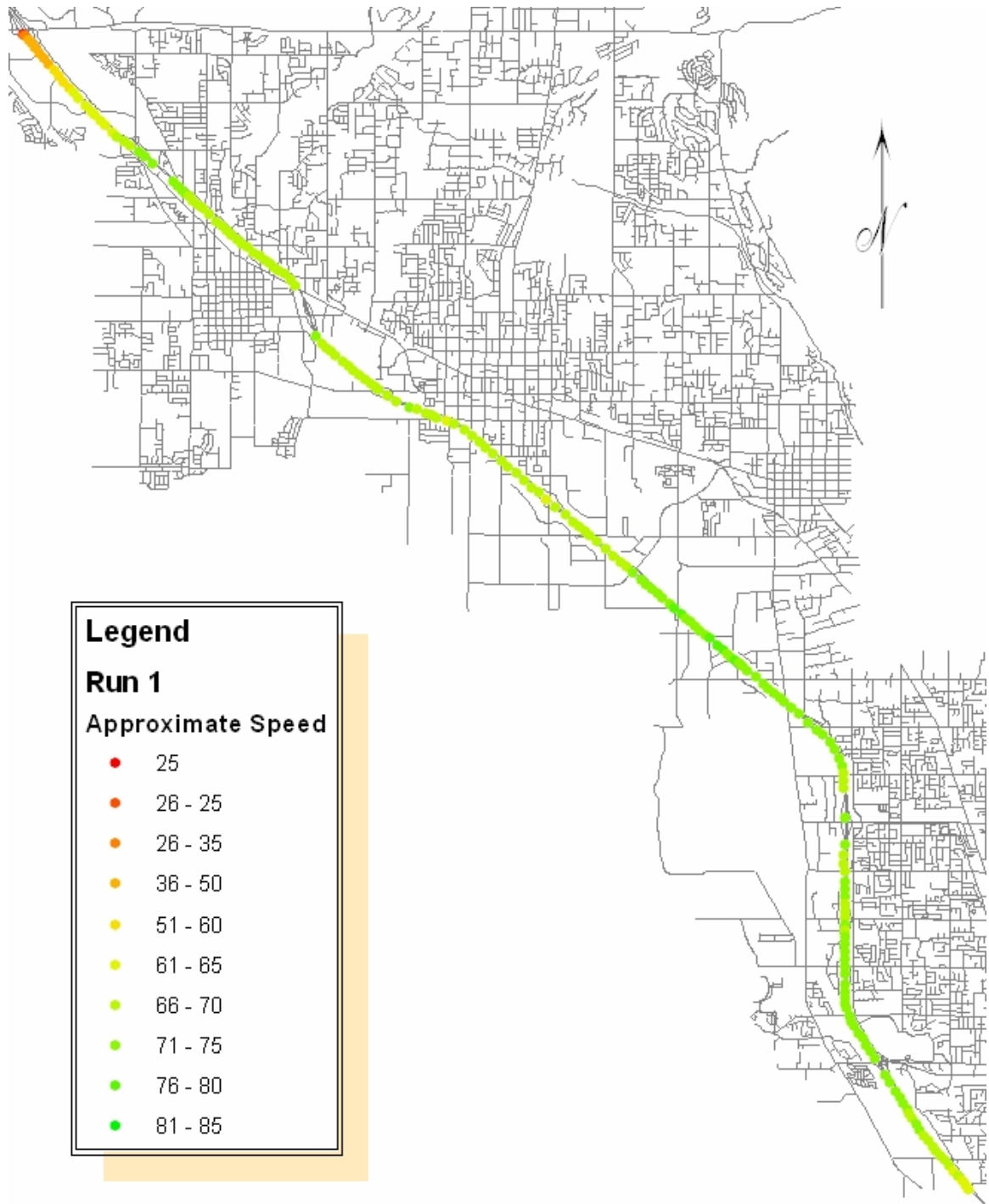


Figure 4-18: Speed map for southbound run 1.

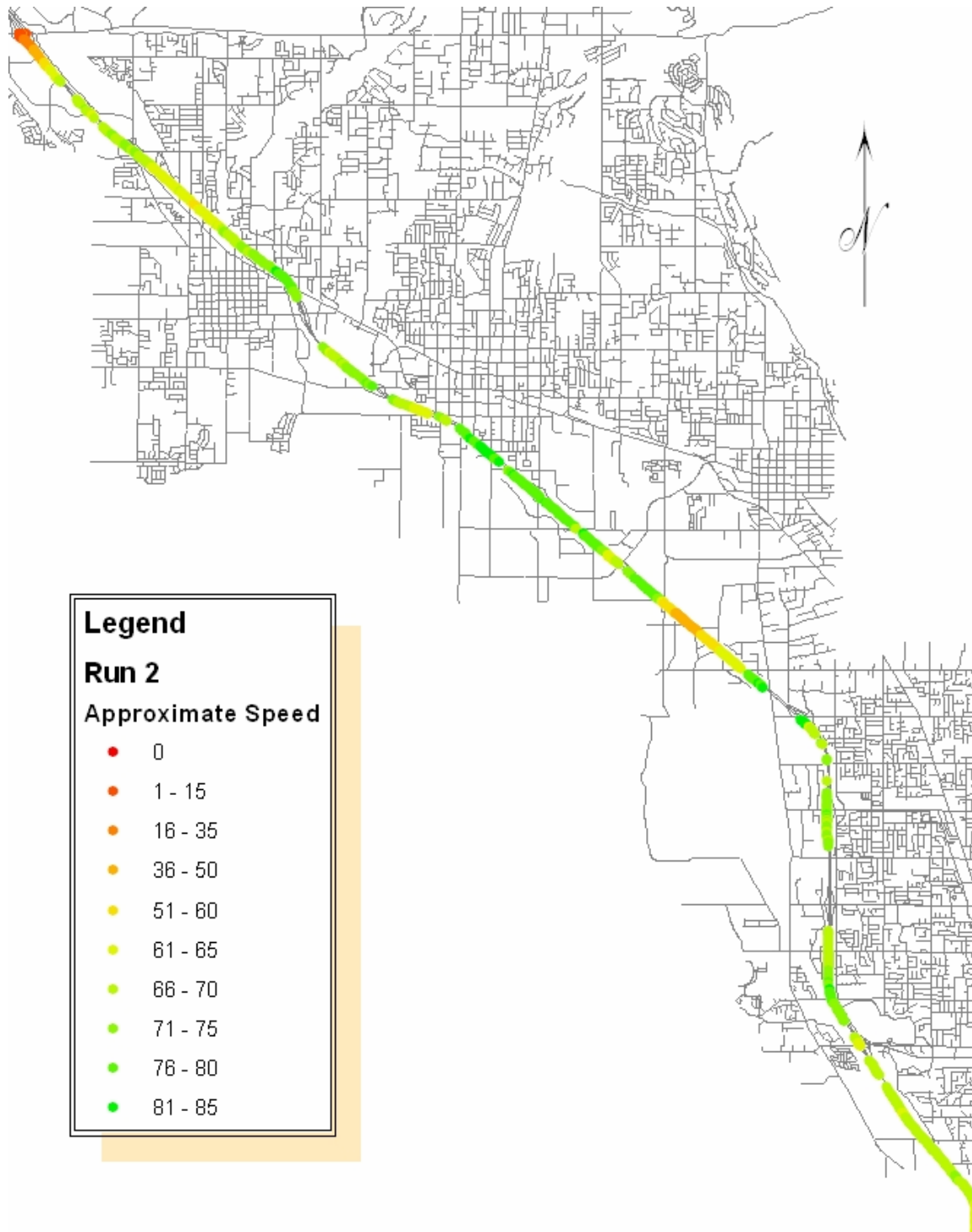


Figure 4-19: Speed map for southbound run 2.

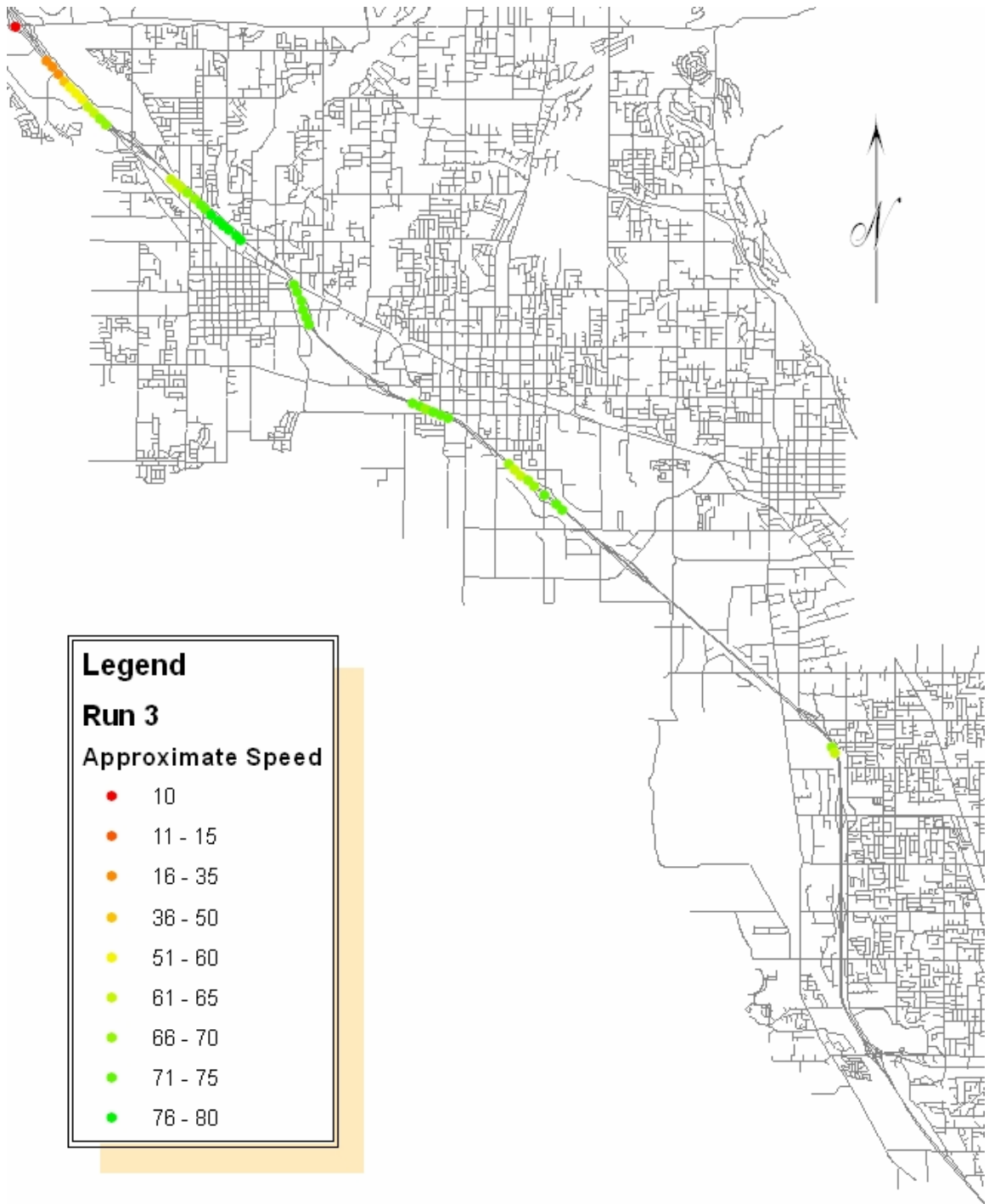


Figure 4-20: Speed map for southbound run 3.

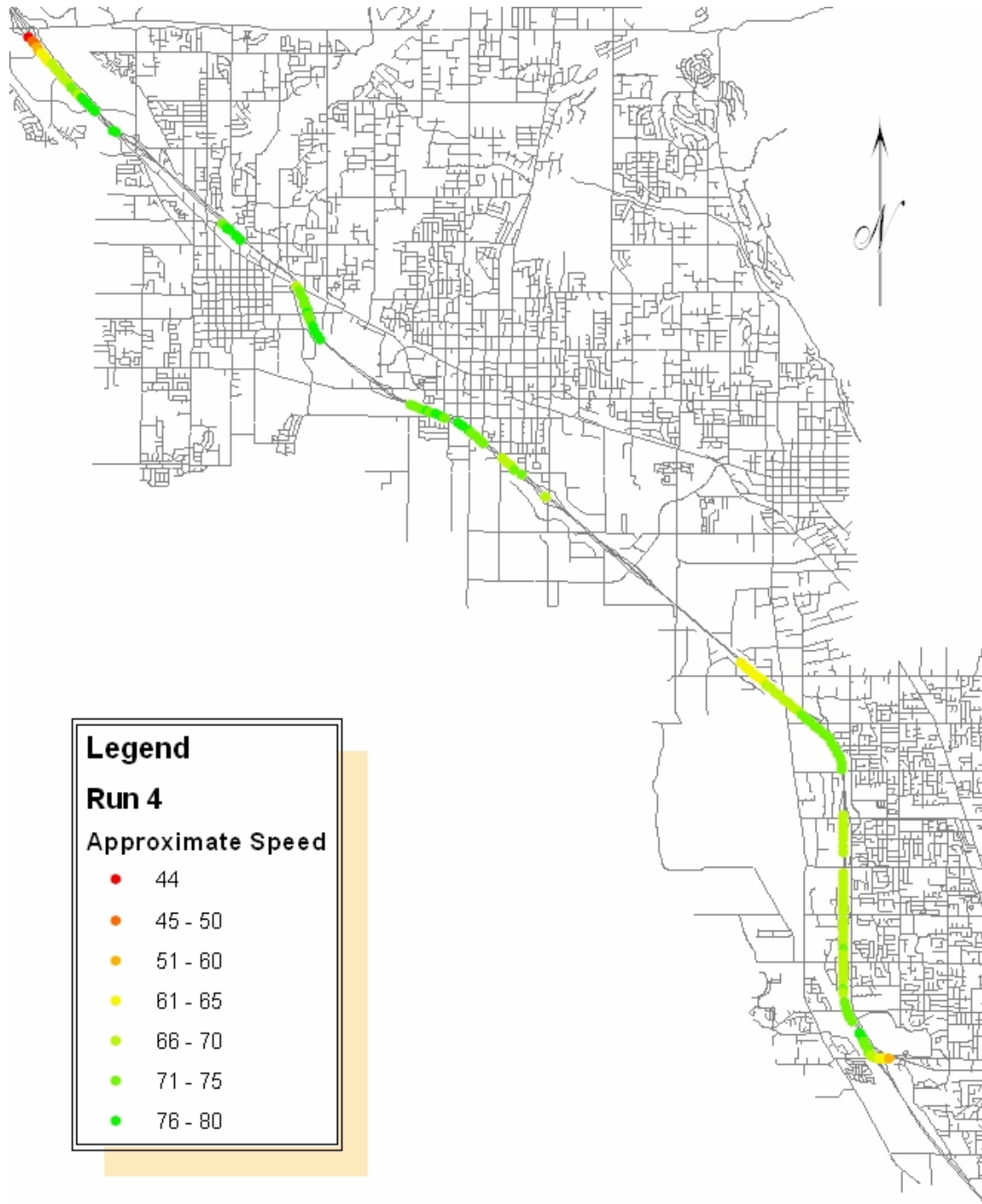


Figure 4-21: Speed map for southbound run 4.

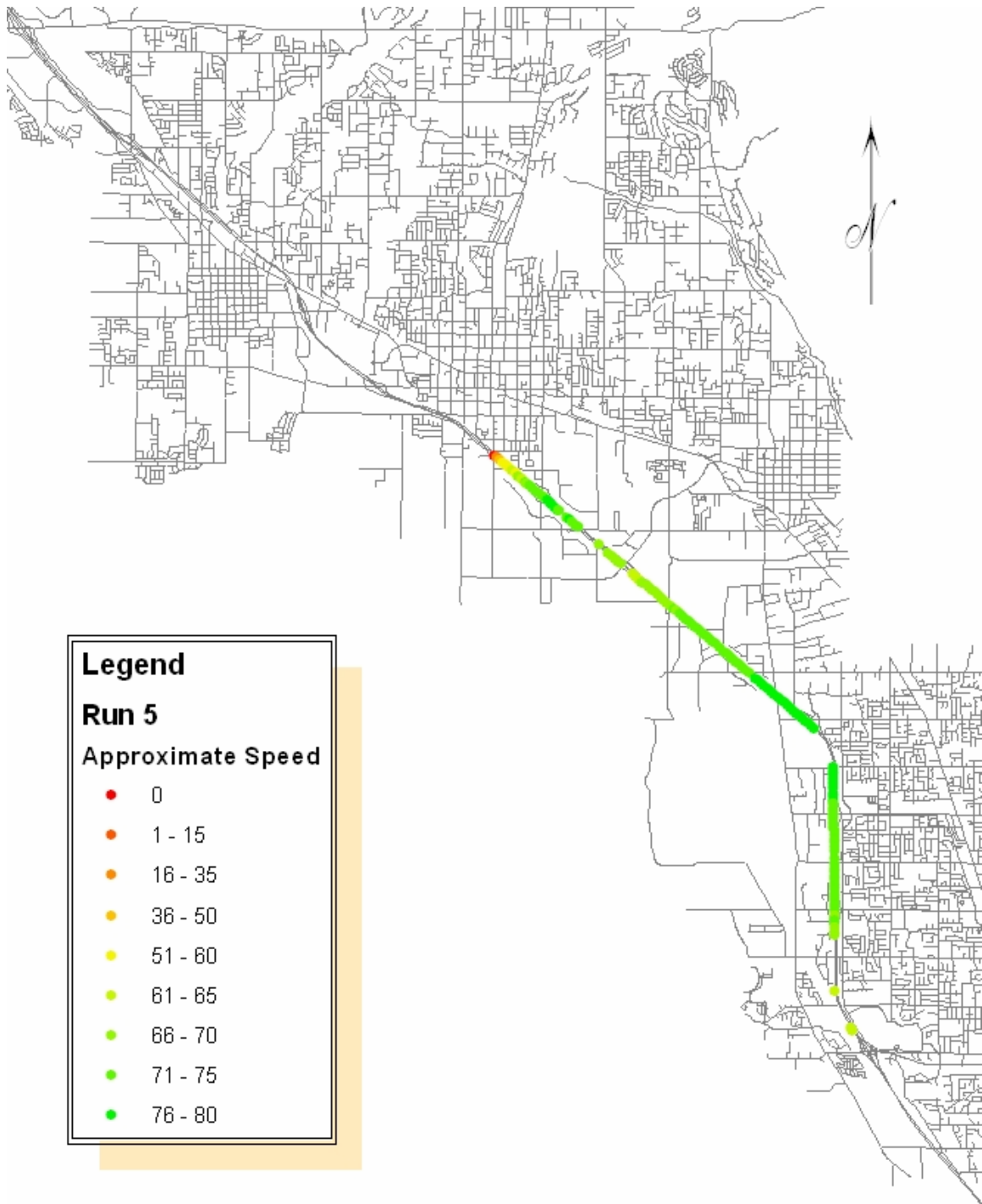


Figure 4-22: Speed map for southbound run 5.

The times of the data points collected at the on- and off-ramps are shown in Table 4-4. Larger sample sizes (four runs) were possible for the southbound data.

Table 4-4: Times Recorded at Node Locations for Southbound Runs

Location		Run Number				
		1	2	3	4	5
Exit 284	<i>off</i>					
<i>Alpine</i>	<i>on</i>	17:48:47	17:40:39	18:47:59	18:19:47	
Exit 282	<i>off</i>	17:49:28	17:41:25	18:49:00	18:20:23	
<i>1200 West, Lehi</i>	<i>on</i>	17:50:05	17:42:00			
Exit 279	<i>off</i>		17:43:51	18:51:13	18:22:43	
<i>Main St., Lehi</i>	<i>on</i>	17:52:11		18:51:30	18:23:03	
Exit 278	<i>off</i>	0.74515	17:44:45			
<i>Main St., American Fork</i>	<i>on</i>	0.74551	17:45:19	18:52:35	18:24:04	
Exit 276	<i>off</i>	17:54:54	17:46:38	18:53:40	18:25:11	17:57:27
<i>500 East, American Fork</i>	<i>on</i>	17:55:28	17:47:09	18:54:06		17:57:57
Exit 275	<i>off</i>	17:55:53	17:47:36			17:58:23
<i>Pleasant Grove Blvd.</i>	<i>on</i>	17:56:23	17:48:13			17:58:56
Exit 273	<i>off</i>	17:58:02			18:28:20	18:00:27
<i>1600 North, Orem</i>	<i>on</i>	17:58:35	17:50:50	18:56:47	18:28:51	
Exit 272	<i>off</i>	17:58:58	17:51:16			18:01:20
<i>800 North, Orem</i>	<i>on</i>	17:59:23	17:51:46		18:29:44	18:01:47
Exit 271	<i>off</i>	17:59:56			18:30:14	18:02:16
<i>Center St., Orem</i>	<i>on</i>	18:00:21	17:52:39		18:30:36	18:02:42
Exit 269	<i>off</i>	18:01:18	17:53:31			18:03:37
<i>University Parkway, Orem</i>	<i>on</i>	0.75118	0.74581			

Table 4-5 shows these calculated travel times. The final column indicates the required sample size for 95 percent confidence that the true average is within five seconds of the calculated average. Required sample sizes were computed using the formula given previously in Equation 4-2. While the southbound travel time data has a lower standard deviation than the northbound data, additional runs would still be required

for 95 percent confidence that the average of this data set represents the true average travel time for the evening peak on this network. The travel times presented in Table 4-5 were used for validation of the parameter sets generated by the calibration program.

Table 4-5: Southbound Travel Times

Segment		Travel Time (seconds)							
		Run Number					Average	Standard Deviation	Required Runs
Start	End	1	2	3	4	5			
Exit 284 on	Exit 282, off	41	46	61	36		46.0	10.8	18
Exit 282, off	Exit 278, on	244	234	215	221		228.5	13.0	26
Exit 278, on	Exit 273, on	303	331	252	287		293.2	33.0	167
Exit 272, on	Exit 271, on	58	53		52	55	54.5	2.6	1

4.3 Concluding Remarks

This chapter describes the sources of the volume and geometry data that were used as inputs to the CORSIM model being calibrated in this research. It also discussed the methodology that was utilized to collect and reduce the travel time data used to calibrate that model. Chapter 5 describes how these data were used with the calibration program presented in Chapter 3 to determine an optimal set of driver behavior parameters for the northbound network.

5 Calibration Procedure

The calibration procedure used for this analysis comprised three steps. First, sensitivity analyses were performed to determine which parameters should be included in the calibration and to find the appropriate population size. Next, the calibration was performed with the genetic algorithm using the parameters selected based on the sensitivity analysis. Finally, the best parameter set from the calibration was validated on a separate network. This chapter describes each of these steps in the calibration procedure.

5.1 Sensitivity Analysis

Two sensitivity analyses were performed prior to running the calibrating parameter sets with the genetic algorithm. First, test CORSIM runs were performed for a minimum and maximum value for each parameter that could be included in the calibration, with all others held constant at the default values. Next, tests were run to determine an optimum population size for the genetic algorithm. This section describes the methodology and results for both of these analyses.

5.1.1 Parameter Selection

The purpose of the first set of sensitivity analyses was to determine which parameters have a significant enough effect on the outcome of the simulation to be

included in the calibration. Nine MOEs for the network were examined as part of this analysis. These include: vehicle-miles, vehicle-minutes, the ratio of moving time to total trip time, average content, current content, speed, total delay, travel time per vehicle mile, and delay time per vehicle mile. Of these, only average content and speed are shown and discussed in this chapter. The results for all nine MOEs are presented in Appendix D.

Average content and speed were chosen as representative MOEs for two reasons. First, the parameters that failed the test based on both of these MOEs also failed for the remaining seven. Second, these two measures are closely related to the MOEs to which the model was to be calibrated (volume and travel time). Average content is the total number of vehicle seconds accumulated on all links over the course of the simulation, divided by the duration of the simulation in seconds. The relationship between density and average content is given in Equation 5-1 (1).

$$D = \frac{C}{Ln} \quad (5-1)$$

where: D = density (vehicles per lane-mile),
 C = average content (vehicles),
 L = length of segment (miles), and
 n = number of lanes.

Volume is related to density by Equation 5-2.

$$V = SD \quad (5-2)$$

where: V = volume (vehicles per hour), and
 S = speed (miles per hour).

Combining Equations 5-2 and 5-2 yields the relationship shown in Equation 5-3.

$$C = \frac{VLn}{S} \quad (5-3)$$

The sensitivity analysis used for parameter selection comprised three steps. First, the simulation was run 10 times with default values for all parameters. Each of these default runs used a different random number seed in order to determine the amount of variability that existed in the model based solely on random vehicle arrival. The results of these test runs are shown in Table 5-1.

Table 5-1: Selected MOEs Resulting from Varying Random Number Seeds

Run Number	Random Number Seed	Average Content (vehicles)	Speed (mph)
1	8871	1618.1	58.1
2	7981	1613.0	58.2
3	3641	1617.7	58.3
4	8351	1612.0	58.2
5	8151	1654.0	57.2
6	1151	1591.1	58.4
7	6487	1610.7	58.3
8	2887	1623.2	57.9
9	6723	1650.5	57.6
10	1467	1603.8	58.4
Average:		1619.4	58.1
Standard Deviation:		19.4	0.4
Range:		62.9	1.3

Next, the simulation was run with the minimum and maximum values of every parameter under consideration. For each of these runs, the values of all variables not being tested, including the random number seeds, were held constant at their default values. For the PITT car-following constant, mandatory lane change gap acceptance, multiplier for desire to make a discretionary lane change, and advantage threshold for a discretionary lane change, these values corresponded to the minimum and maximum allowed by CORSIM. The ranges for all other parameters were those determined by Schultz (52) based on reasonableness criteria.

Because the random number seeds were held constant for all of these runs, any variability in the model output between runs using the minimum and maximum values of a particular parameter was entirely the result of the difference in those parameter values. The focus of this analysis was not to determine whether the parameter sets yielded accurate results, but rather to determine whether significant changes in the value of a particular parameter might have a significant effect on the model output.

For parameters that can be described by a distribution (i.e. car-following sensitivity for 10 driver types and maximum non-emergency freeway deceleration for nine vehicle types), all values were set to the same extreme. For example, all 10 driver types were assigned a car following sensitivity factor of 30 for the minimum case and 150 for the maximum case. The results of these test runs are shown Table 5-2.

The final step of the parameter selection sensitivity analysis was to compare the observed ranges resulting from changing a single parameter value while holding the random number seed constant, to the observed range resulting from varying the random number seed while holding all parameters constant, shown in Table 5-3.

Table 5-2: Ranges of MOEs from Varying Parameter Values

		Network MOEs	
Parameter	Values	Average Content	Speed (mph)
Multiplier for desire to make discretionary lane change	1	1629.1	57.57
	10	1657.0	56.28
	<i>Range</i>	9	27.9
Advantage threshold for discretionary lane change	1	2580.7	29.76
	10	1640.5	57.67
	<i>Range</i>	9	940.2
Mandatory lane change gap acceptance	1	1615.5	58.19
	6	1640.2	57.90
	<i>Range</i>	5	24.7
Time to complete lane change maneuver	10	1690.9	54.74
	60	1598.2	58.70
	<i>Range</i>	50	92.7
Percent of drivers yielding to merging vehicles	5	1615.0	57.98
	40	1637.8	57.76
	<i>Range</i>	35	22.8
Minimum separation for generation of vehicles	10	1423.4	59.58
	30	1629.7	58.09
	<i>Range</i>	20	206.3
Pitt's car following constant	3	1590.1	59.35
	10	1613.0	58.21
	<i>Range</i>	7	22.9
Maximum non-emergency deceleration	40	1570.3	60.13
	120	1583.4	59.17
	<i>Range</i>	80	13.1
Acceleration lag time	1	1568.3	59.61
	10	1606.2	59.27
	<i>Range</i>	9	37.9
Deceleration lag time	1	1650.5	56.99
	10	1629.1	57.67
	<i>Range</i>	9	21.4
Car following sensitivity factors	30	2118.8	40.78
	150	1521.7	61.14
	<i>Range</i>	120	597.1

Table 5-3: Default Ranges and Results of Sensitivity Analysis

	Average Content (vehicles)	Speed (mph)
Default Range	62.9	1.3
Mandatory Lane Change Gap Acceptance	24.7	0.3
Time to Complete Lane Change Maneuver	92.7	4.0
Percent of Drivers Yielding to Merging Vehicles	22.8	0.2
Minimum Separation for Generation of Vehicles	206.3	1.5
Multiplier for Desire to Make Discretionary Lane Change	27.9	1.3
Advantage Threshold for Discretionary Lane Change	940.2	27.9
PITT Car Following Constant	22.9	1.1
Maximum Non-emergency Freeway Deceleration	13.1	1.0
Acceleration Lag Time	37.9	0.3
Deceleration Lag Time	21.4	0.7
Car Following Sensitivity Factors	597.1	20.4

Based on the results shown in Table 5-3, mandatory lane change gap acceptance, percent of drivers desiring to yield right-of-way to merging vehicles, the multiplier for desire to make a discretionary lane change, the PITT car following constant, maximum non-emergency freeway deceleration, and acceleration and deceleration lag times were all omitted from the calibration, as illustrated by the bold values for average content and speed. The variation in network MOEs that resulted from these parameters being set to their minimum and maximum values was no greater than the variation that resulted from

simply varying the random number seed. As mentioned previously, although only average content and speed are shown here, the parameters that failed based on both of these MOEs also failed based on all nine MOEs examined.

5.1.2 Determination of Population Size

A second analysis was performed to determine the optimum population size for the genetic algorithm. The results of prior research on this issue differ widely, with recommendations ranging from 4 (4) to 20 (52). The recommendation of a population size of 4 by Cheu et al. (4) is based on initial testing in which population sizes of 4, 10, and 40 were each tested in a genetic algorithm which included 400 total CORSIM runs, assuming that a run is made for every chromosome at every iteration. Thus, the tests included 100 iterations for a population of 4, 40 iterations for a population of 10, and 10 iterations for a population of 40. In these initial tests, the smallest population converged to the highest value by the end of 400 runs.

For Schultz's tests (52), the algorithm was run for a predetermined number of iterations rather than a predetermined number of runs. Populations of 10, 20, 30, and 40 chromosomes were each run for 30 iterations. In these tests, the larger populations reached better fitness values, although the algorithm took considerably more time to reach these values (considering that each iteration with 40 chromosomes takes approximately four times as long as a population of 10 chromosomes).

For this research, tests were run to further explore this question. Test runs of the genetic algorithm were performed with population sizes of 5, 10, 15, 20, 25, and 30. One hundred iterations were performed for each of these population sizes. The results of these

runs are shown in Figure 5-1. The larger populations tended to begin and end at higher fitness values. Smaller populations tended to begin and end at lower fitness values, but converged more quickly and with greater increases between the initial and final fitness values. It may be possible to capitalize on the advantages of both small and large populations by starting the algorithm with a large population size in order to achieve a high initial fitness value and reducing the population size after the first iteration.

In order to test this possibility, an algorithm was run with an initial population of 30, which was reduced to 5 after the first iteration. The results of this hybrid population size are shown in Figure 5-2, compared to the initial tests with population sizes of 5 and 30. In this case, populations of both 5 and 30 still converged to a higher fitness value than the hybrid population. This result is not necessarily conclusive because the initial populations are randomly generated, so initial fitness values are largely a matter of chance. Tests on each population size were repeated to obtain a larger sample size from which to draw a better conclusion.

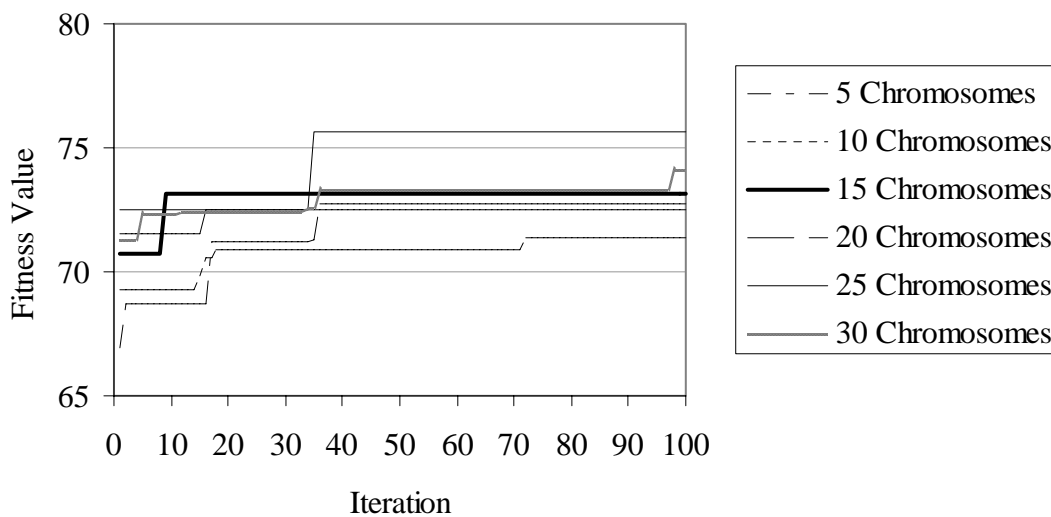


Figure 5-1: Comparison between population sizes.

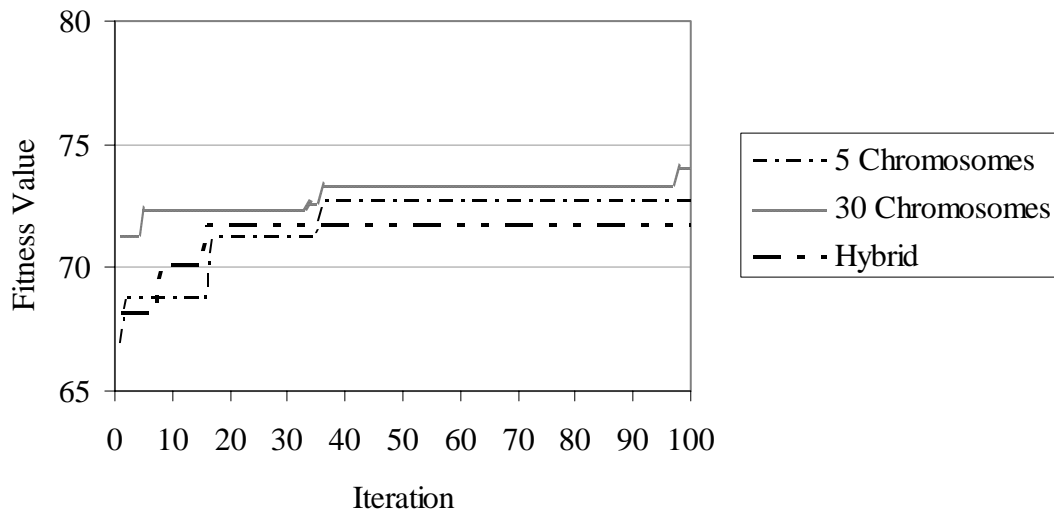


Figure 5-2: Comparison of hybrid, smallest, and largest populations.

The average results of both sets of runs are tabulated in Table 5-4. In these tests the hybrid population and the population of 5 chromosomes increased by the greatest percentage over 100 iterations, but population sizes between 20 and 30 still reached the highest final fitness values. The fitness values were computed using the fitness function given previously in Equations 3-5 through 3-7.

Table 5-4: Average Results of Population Size Analysis

	Initial Fitness	Final Fitness	Absolute Increase	Percent Increase
5 Chromosomes	67.57	71.73	4.16	6%
10 Chromosomes	69.40	72.34	2.93	4%
15 Chromosomes	71.05	73.19	2.14	3%
20 Chromosomes	70.93	74.07	3.15	4%
25 Chromosomes	71.14	74.65	3.50	5%
30 Chromosomes	71.51	74.17	2.66	4%
Hybrid	68.32	72.14	3.82	6%

In addition to the final value, the time required to reach this value is also an important consideration in selecting a population size. These average times are shown in terms of both number of iterations and hours in Figure 5-3.

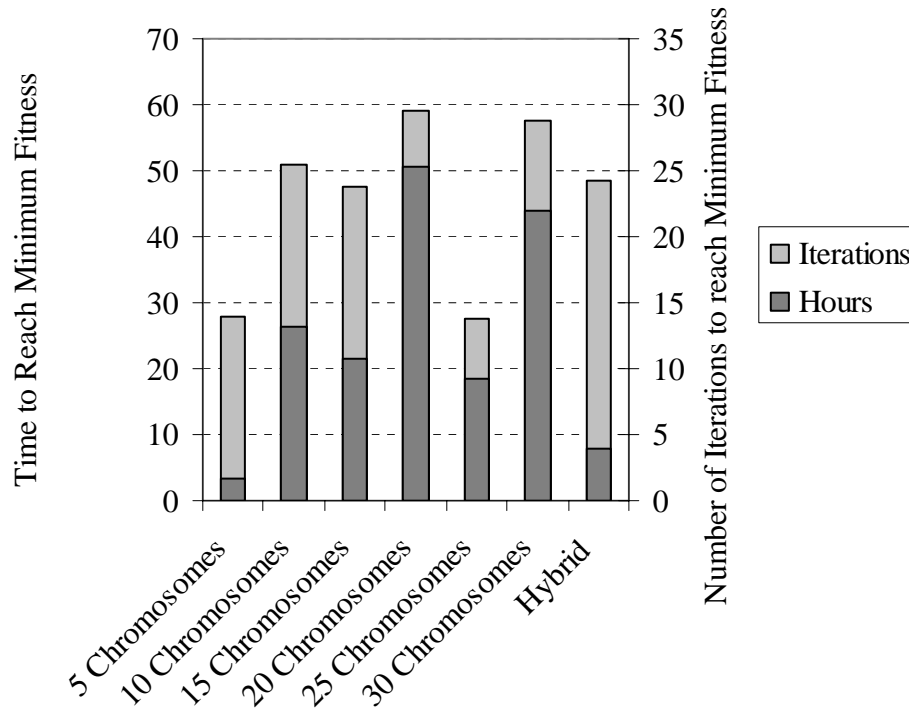


Figure 5-3: Comparison of average time required between population sizes.

Based on the results of this analysis, a population size of 25 was chosen for the calibration procedure with 30 iterations. Further research is required to determine whether population size is related to final fitness value and whether there is an advantage to varying the size of the population over the life cycle of the algorithm. Davis (50) has had success with varying the values of genetic algorithm parameters such as probabilities of crossover and mutation and population size using a technique called adaptive operator fitness. Applying this technique to the calibration algorithm may improve efficiency.

5.2 Calibration

Based on the results of the sensitivity analyses, the calibration program was run with a population of 25 chromosomes for 30 iterations. Car-following sensitivity factors, the advantage threshold for a discretionary lane change, the minimum separation for generation of vehicles, and the time required to complete a lane change maneuver were included in the calibration. All other parameters were held constant at default values.

As discussed in Section 3.2.1.4, the calibration program allows the user to calibrate based on observed volumes, observed travel times, or a weighted average of the two fitness values (Equation 3-7). The weight placed on each MOE is specified by the value of α in the calibration parameters dialog. An α value of 0.5 weights volume and travel time equally. An α value greater or less than 0.5 places more weight on travel time or volume, respectively. Normally, the value of α would be chosen based on the intended application of the model and whether it is more critical for the model to accurately predict traffic volumes or travel times. For example, if the purpose of the simulation is to evaluate travel demand management strategies that are intended to reduce the peak hour commute time between neighboring cities, greater emphasis should be placed on travel time and α should be set to a value greater than 0.5. If the purpose is to determine whether the capacity of a facility will fall below demand as result of a possible lane closure, greater emphasis should be placed on volume and α should be set to a value less than 0.5.

Section 2.3.5.4 suggests that selecting a fitness function that includes too many MOEs may unnecessarily complicate the algorithm. It is possible that there may be an advantage to calibrating a model to volume or travel time rather than a linear combination

of the two, even when both measures are of interest. In order to determine the effects of different combinations of these two measures, the calibration program was run five times with α values of 0.0, 0.25, 0.5, 0.75, and 1.0. Running the calibration with α values of 0 or 1 is equivalent to calibrating to only volume or travel time, but when both sets of field data were included as inputs, the program calculated fitness values for both, although only one of the two values was used in the selection process. The changes in MAER values between the parameter sets found by each of these calibration tests, compared to the MAER values resulting from the default parameters sets are illustrated in Figure 5-4.

When only travel time was included in the fitness function, the MAER value of volume actually increased. In all other cases, there was improvement in both MAER values.

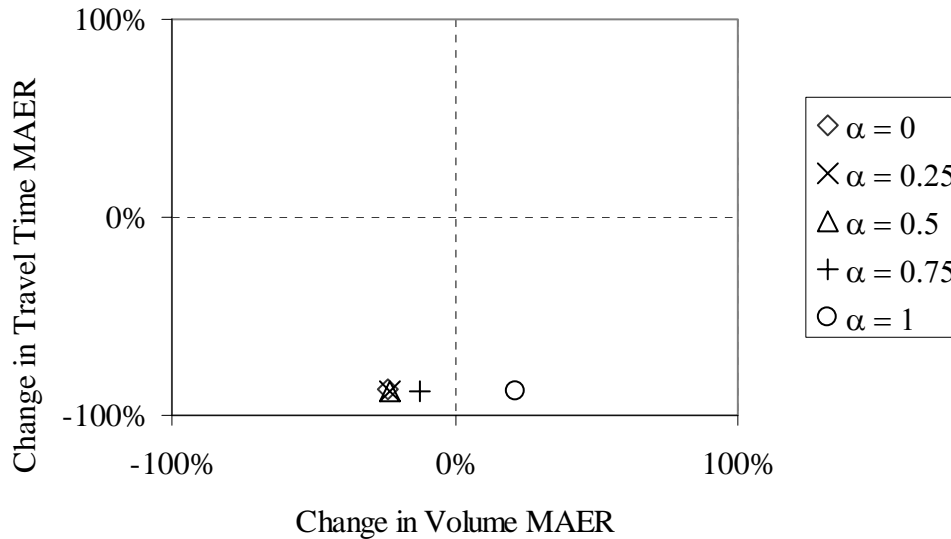


Figure 5-4: Change in volume MAER versus decrease in travel time MAER.

The results illustrated in Figure 5-4 are shown in more detail in Table 5-5, which shows the fitness and MAER values resulting from the default parameter sets and the calibrated parameter sets.

The upstream volume (the volume of vehicles exiting the network) and the total travel time for all mainline links are also shown in Table 5-5. Note that a decrease of approximately 87 percent in the travel time MAER value may represent a difference in total travel times that varies by 25 seconds.

The five resulting parameter sets are given in Table 5-6. The parameter sets shown in Table 5-6 were applied to a model of the southbound network as part of the validation procedure, which is described in the following section.

5.3 Validation

The final parameter set from each of the five calibration runs (shown in Table 5-6) were applied to a model of the southbound direction of the same section of I-15 that was used for calibration. For each parameter set, one CORSIM run was performed and the results were compared to field data collected as described in Chapter 4. MAER values were calculated for volumes and travel time.

The MAER values for each of the five validation runs are shown in Table 5-7. In these tests, the calibrated parameters sets did not present any consistent advantage over the default parameters for either volume or travel time. This could be because the southbound direction has very different driver behavior characteristics than the northbound network.

Table 5-5: Fitness and MAER Values of Default and Calibrated Parameter Sets

α		Value with Default Parameters	Value with Calibrated Parameters	Absolute Difference	Percent Difference
0	Total Fitness:	74.5397	79.5793	5.0396	6.8%
	Volume Fitness:	74.0479	79.5793	5.5314	7.5%
	Travel Time Fitness:	76.0152	74.5851	-1.4301	-1.9%
	Volume MAER:	0.0601	0.0457	-0.0144	-24.0%
	Travel Time MAER:	0.8776	0.1173	-0.7603	-86.6%
	Upstream Volume (veh/hr)	5130	5156	26	0.5%
	Total Travel Time (sec)	1063.1	1082.7	19.6	1.8%
0.25	Total Fitness:	74.5397	78.6103	4.0706	5.5%
	Volume Fitness:	74.0479	79.4058	5.3579	7.2%
	Travel Time Fitness:	76.0152	76.2236	0.2084	0.3%
	Volume MAER:	0.0601	0.0461	-0.014	-23.3%
	Travel Time MAER:	0.8776	0.1086	-0.769	-87.6%
	Upstream Volume (veh/hr)	5130	4995	-135	-2.6%
	Total Travel Time (sec)	1063.1	1062.9	-0.2	0.0%
0.5	Total Fitness:	74.5397	77.9244	3.3847	4.5%
	Volume Fitness:	74.0479	79.4258	5.3779	7.3%
	Travel Time Fitness:	76.0152	76.4229	0.4077	0.5%
	Volume MAER:	0.0601	0.0461	-0.014	-23.3%
	Travel Time MAER:	0.8776	0.1076	-0.77	-87.7%
	Upstream Volume (veh/hr)	5130	5039	-91	-1.8%
	Total Travel Time (sec)	1063.1	1069.0	5.9	0.6%
0.75	Total Fitness:	74.5397	77.0334	2.4937	3.3%
	Volume Fitness:	74.0479	76.7785	2.7306	3.7%
	Travel Time Fitness:	76.0152	77.1184	1.1032	1.5%
	Volume MAER:	0.0601	0.0528	-0.0073	-12.1%
	Travel Time MAER:	0.8776	0.1039	-0.7737	-88.2%
	Upstream Volume (veh/hr)	5130	5052	-78	-1.5%
	Total Travel Time (sec)	1063.1	1055.5	-7.6	-0.7%
1	Total Fitness:	74.5397	77.1775	2.6378	3.5%
	Volume Fitness:	74.0479	69.3531	-4.6948	-6.3%
	Travel Time Fitness:	76.0152	77.1775	1.1623	1.5%
	Volume MAER:	0.0601	0.0732	0.0131	21.8%
	Travel Time MAER:	0.8776	0.1036	-0.774	-88.2%
	Upstream Volume (veh/hr)	5130	5012	-118	-2.3%
	Total Travel Time (sec)	1063.1	1058.5	-4.6	-0.4%

Table 5-6: Parameter Sets Generated by Five Calibration Tests

Parameter	α				
	0	0.25	0.5	0.75	1
Car-following sensitivity factor for driver type 1	150	150	150	150	150
Car-following sensitivity factor for driver type 2	150	150	150	150	150
Car-following sensitivity factor for driver type 3	150	150	150	146	150
Car-following sensitivity factor for driver type 4	137	113	121	107	123
Car-following sensitivity factor for driver type 5	101	54	72	54	70
Car-following sensitivity factor for driver type 6	61	30	40	33	36
Car-following sensitivity factor for driver type 7	38	30	30	30	30
Car-following sensitivity factor for driver type 8	30	30	30	30	30
Car-following sensitivity factor for driver type 9	30	30	30	30	30
Car-following sensitivity factor for driver type 10	30	30	30	30	30
Time to complete lane-change maneuver	42	56	32	45	23
Minimum separation for generation of vehicles	13	26	10	26	16
Multiplier for desire to make a discretionary lane change	8	4	5	5	5
Advantage threshold for discretionary lane change	8	1	8	6	5

Table 5-7: Results of Validation Tests

	Error (%)	
	Volume	Travel Time
Default:	7	14
α		
0.00	9	16
0.25	20	13
0.50	9	15
0.75	22	13
1.00	8	14

The studies by Cheu et al. (4) and Schultz (52) suggest that drivers in the peak direction display different driving behavior than drivers in the off-peak direction.

However, this section of I-15 has a directional factor of 0.51, indicating that the peak direction is not well-defined. There may also be a bottleneck in the northbound direction

that does not exist in the southbound direction. Figure 4-12 shows significant slowing at the sharp curve in the roadway near 1600 North in Orem (Exit 273). Figure 4-18 shows that this slowing did not occur in the southbound direction.

Another possibility is that the parameters were over-fit to the northbound data. Over-fitting can occur when parameters are adjusted to force a model to fit calibration data too closely, masking any possible inaccuracies in the calibration data. In practice, the engineer should have an idea of how accurate the calibration data is and should not attempt to calibrate parameters to yield model outputs with less error than the model inputs are likely to contain; however, quantifying the degree of uncertainty in a calibration data set can be very difficult.

5.4 Concluding Remarks

The chapter described the procedure that was used to calibrate a CORSIM model to volume and travel time data. This was done in three steps: initial sensitivity analyses, calibration using the calibration program described in Chapter 3, and validation of the resulting parameter set on a separate network.

The initial sensitivity analysis demonstrated a simple method whereby it can be determined whether adjusting the values of a particular parameter can result in greater variability in model outputs than is caused by random variation within the CORSIM model. Those parameters which were thus determined not to have a large impact on the model outputs were excluded from the analysis. A second part of the sensitivity analysis suggested that, holding the number of iterations constant, larger populations (20 to 25

chromosomes) tend to reach higher fitness values than smaller populations (5 to 10 chromosomes).

The calibration step of the procedure was carried out with five different forms of the same fitness function, each assigning a different relative importance to volume and travel time. The fitness functions that included both of these MOEs, but did not weight them equally, showed the greatest overall reduction in the MAER values for both MOEs.

In the validation step, the parameter sets found by calibrating a model of the northbound network were applied to the southbound network. By and large, parameter sets calibrated to the northbound network provided little improvement over the default parameters when applied to the southbound network.

6 Conclusion

The first four chapters of this thesis introduced the goals and objectives of this research, summarized the methods used by the calibration program to find an optimal set of driver behavior parameters for a CORSIM model of a linear freeway network, and outlined methods used to collect and reduce volume and travel time data for use as a baseline in the calibration procedure.

Chapter 5 described how the procedures and data presented in the previous chapters were applied in the calibration of a hypothetical linear freeway network based on a 13-mile section of I-15. This section summarizes the main conclusions that can be drawn from the sensitivity analysis, calibration, and validation procedures and makes recommendations regarding possible areas of future research.

6.1 Sensitivity Analysis

The sensitivity analysis comprised two parts. The first examined the effects that extreme values of each calibration parameter have on the CORSIM model output. The second examined the effect that population size has on the convergence of the genetic algorithm. The following sections summarize some of the conclusions that can be drawn from these analyses, as well as questions that were raised that may warrant further research.

6.1.1 Sensitivity Analysis Conclusions

Based on the first part of the analysis, the parameters specifying mandatory lane change gap acceptance, percent of drivers willing to yield right of way to merging vehicles, multiplier for desire to make a discretionary lane change, the PITT car-following constant, maximum non-emergency freeway deceleration, and acceleration and deceleration lag times did not appear to have a significant effect on the model output and therefore were not included in the simulation. That is not to say that these parameters can be set to any arbitrary value, but as long as they are kept within a reasonable range, the accuracy of the model did not appear to be significantly affected.

The second part of the sensitivity analysis determined that larger population sizes (between 20 and 30 chromosomes) do tend to yield higher final fitness values than smaller sizes (between 5 and 15 chromosomes), when the number of iterations is held constant. If the time required to perform the calibration is held constant, using smaller population sizes or possibly decreasing the size of the population over the course of the calibration may present a distinct advantage.

6.1.2 Questions Raised by Sensitivity Analysis

Interactive effects of the calibration parameters were not considered in the analysis of the effects that extreme values have on the model outputs. For example, although several parameters were excluded because they did not have an effect when all other values were held constant at the CORSIM default values, the effect of these parameters may be mitigated or intensified when the values of the other parameters change. This is a question that should be further addressed.

Additionally, the question of whether larger population sizes may be more appropriate in earlier iterations than in later iterations was briefly addressed in the second part of the sensitivity analysis. This could be explored more fully in future research.

6.2 Calibration

Five calibration tests were run, each with a different fitness function, assigning relative weights of 100/0, 25/75, 50/50, 75/25, and 0/100 to volume and travel time fitness, respectively. The following sections summarize some of the conclusions that can be drawn from these calibration tests, as well as areas that may warrant further research.

6.2.1 Calibration Conclusions

When the model was calibrated to travel time alone, decreases in travel time error were accompanied by increases in volume error. When both MOEs were included in the calibration, a reduction in error was observed for both MOEs. This suggested that it may be advisable to include both MOEs in the calibration if possible, even when only one MOE is of interest for a particular study.

6.2.2 Questions Raised by Calibration

The above conclusions are based on only one test for each fitness function. Because of the stochastic nature of the genetic algorithm on which the calibration program is based, it is not apparent whether it would always be best to include both volume and travel time in the calibration procedure. While this question is beyond the

scope of this thesis, it would be an interesting question to explore through further research.

6.3 Validation

The validation step consisted of applying the calibrated parameters from the northbound network to the southbound network and comparing the accuracy of the results of the model with the calibrated parameters to those of the model with default values. The following sections summarize the conclusions that can be drawn from the validation runs, as well as questions that were raised that could be further explored.

6.3.1 Validation Conclusions

On the southbound network, the calibrated parameters did not represent any improvement over the default values. This could be because the southbound direction has very different driver behavior characteristics than the northbound network. This theory is supported by Cheu's (4) and Schultz's (52) research, which suggest that drivers exhibit very different behavior, depending on conditions of the network (such as peak direction). However, this section of I-15 has a directional factor of 0.51, indicating that the peak direction is not well-defined. The speed data shown in Figure 4-3 through Figure 4-22 suggest that there may be a bottleneck in the northbound direction that could affect driver behavior on the northbound network. Because the calibration data is of not as accurate as would be required for any site-specific conclusions, it is also likely that the parameters have been over-fit to the data used for calibration of the northbound network.

6.3.2 *Questions Raised by Validation*

It would be beneficial to determine a way to calibrate to an optimal level of error in order to find the parameter set that may not provide the best possible fit for one specific network, but will provide an acceptable fit for a large class of networks having similar characteristics. It would be useful to develop a method to find this optimal error to yield more widely applicable parameter sets.

6.4 **Concluding Remarks**

The importance of calibrating a microscopic traffic simulation model to local conditions cannot be overemphasized. One critical aspect of calibration is the determination of appropriate driver behavior parameters for a particular locality. This thesis addressed several issues with regards to calibrating driver behavior parameters that can aide engineers and researchers in creating models that more accurately reflect reality.

The purpose of this research was three-fold: 1) to develop a user-friendly GUI for an automated calibration program that could easily be utilized by a practicing engineer, 2) to determine which driver behavior parameters ought to be included in such a calibration procedure by performing necessary sensitivity analyses, and 3) to demonstrate an automated calibration procedure using local data. The GUI developed for this research suited these purposes very well and provides a framework that an engineer can use to quickly and easily implement an automated calibration procedure with any programming skills or familiarity with the format of the CORSIM input (*.trf) file. The sensitivity analyses performed in Chapter 5 determined that over half of the available calibration parameters could be omitted from these calibration tests without significantly

affecting the outcome. Finally, the calibration procedure demonstrated in this research was successful in finding parameter sets that improved the accuracy of the model, relative to the default parameter values.

References

1. *CORSIM User's Guide* (Software Help Menu). FHWA, U.S. Department of Transportation, Washington, DC, 2001.
2. *TSIS User's Guide* (Software Help Menu). FHWA, U.S. Department of Transportation, Washington, DC, 2001.
3. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing, Boston, MA. 1989.
4. Cheu, R. L., X. Jin, K. C. Ng, Y. L. Ng, and D. Srinivasan. Calibration of FRESIM for Singapore Expressway Using Genetic Algorithm. In *Journal of Transportation Engineering, Vol. 124, No. 6*. American Society of Civil Engineers, Reston, VA. 1998, pp. 526 – 535.
5. Alexiadis, V., K. Jeannotte, and A. Chandra. *Traffic Analysis Toolbox: Volume I: Traffic Analysis Tools Primer*. Publication FHWA-HRT-04-038. FHWA, U.S. Department of Transportation. 2004.
6. Meyer, M. D., and E. J. Miller. *Urban Transportation Planning: A Decision-Oriented Approach*. McGraw Hill, New York, 2001.
7. HDM-4 Product Information Brochure. HDM Global.
<http://www.hdmglobal.com/AboutHDM4.htm>. Accessed February 14, 2007.
8. Quickzone: A New Work Zone Delay Estimation Tool. Turner-Fairbank Highway Research Center, FHWA, U. S. Department of Transportation, Washington, DC.
<http://www.tfhrc.gov/its/quickzon.htm>. Accessed February 14, 2007.
9. Surface Transportation Efficiency Analysis Model (STEAM). FHWA, U. S. Department of Transportation, Washington, DC.
<http://www.fhwa.dot.gov/steam/index.htm>. Accessed February 14, 2007.
10. Product Specifications. Strong Concepts.
<http://www.strongconcepts.com/ProductSpecs.htm#SITE>. Accessed February 14, 2007.

11. Pas, E. I. Recent advances in activity-based travel demand modeling. *Presented at the TMIP Conference on Activity-Based Travel Forecasting*, New Orleans, LA, June 2-5, 1996.
12. Horowitz, A. J. *Reference Manual, Quick Response System II*. Version 7. Center for Urban Transportation Studies. University of Wisconsin, Milwaukee. 2004.
13. TransCAD Overview. Caliper Corporation, Newton, MA.
<http://www.caliper.com/tcovu.htm>. Accessed February 14, 2007.
14. MinUTP: The Classic 4-Step Forecasting Package for DOS. Citilabs.
<http://www.citilabs.com/minutp/index.html>. Accessed February 14, 2007.
15. TP+: Powerful Scripting for Unlimited Model Structures. Citilabs.
<http://www.citilabs.com/tpplus/index.html> accessed February 14, 2007.
16. What's New In VISUM 9.4. PTV Vision. http://www.english.ptv.de/cgi-bin/traffic/traf_visum.pl. Accessed February 14, 2007.
17. SIDRA Solutions. Akcelik and Associates, Ltd.
<http://www.aatraffic.com/SIDRA/aboutsidra.htm>. Accessed February 14, 2007.
18. CCG/CALC 2 – Signalized Intersection Analysis Software. BA Group.
<http://www.bagroup.com/Pages/software/CCGCALC.html>. Accessed February 14, 2007.
19. HCS+, Release 5.21. McTrans, University of Florida, Gainesville.
<http://mctrans.ce.ufl.edu/hcs/hcsplus/>. Accessed February 17, 2007.
20. TRAFFIX Online. Dowling Associates, Inc. <http://www.traffixonline.com/>. Accessed February 14, 2007.
21. *Synchro 6.0 User Guide*. Trafficware, Inc., Albany, CA. 2004.
22. Passer II-02, Version 1.0. Texas A&M University, College Station.
http://ttisoftware.tamu.edu/fraPasserII_02.htm. Accessed February 14, 2007.
23. TRANSYT-7F, United States Version. McTrans, University of Florida, Gainesville.
<http://mctrans.ce.ufl.edu/featured/TRANSYT-7F/>. Accessed February 14, 2007.
24. TSDWIN 3, The Latest Version of Fortran's Popular Time Space Diagram Application. Fortran. <http://www.fortrantraffic.com/whatsnew/new2.htm>. Accessed February 14, 2007.
25. Owen, L. E., Y. Zhang, L. Rao, and G. McHale. Traffic Flow Simulation Using CORSIM. In *Proceedings of the 2000 Winter Simulation Conference*. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds. ITT Industries, Dept. of Intelligent

- Transportation Systems; National University of Singapore, Dept. of Electrical Engineering, Singapore. 2000, pp. 1143-1147.
26. Druitt, S. An Introduction to Microsimulation. In *Traffic Engineering and Control*, Vol. 39, No 9. 1998.
 27. CORFLO. McTrans, University of Florida, Gainesville.
<http://mctrans.ce.ufl.edu/store/description.asp?itemID=441>. Accessed February 15, 2007.
 28. Kwon, E., and P. Michalopoulos. Macroscopic Simulations of Traffic Flows in Complex Freeway Segments on a Personal Computer. Institute of Electrical and Electronics Engineers, Minneapolis, MN.
<http://ieeexplore.ieee.org/iel3/3948/11424/00518859.pdf?arnumber=518859>. Accessed February 15, 2007.
 29. SATURN. WS Atkins. <http://www.saturnsoftware.co.uk/7.html>. Accessed February 15, 2007.
 30. Contram. <http://www.contram.com/>. Accessed February 15, 2007.
 31. Sundaram, S. *Development of a Dynamic Traffic Assignment System for Short-Term Planning Applications*. Master's Thesis, Massachusetts Institute of Technology, Cambridge, 2002.
 32. DYNASMART-P. McTrans, University of Florida, Gainesville.
<http://mctrans.ce.ufl.edu/featured/dynasmart/>. Accessed March 8, 2007.
 33. Figueiredo, L., J. A. T. Machado, and J. R. Ferreira. Dynamical Analysis of Freeway Traffic. *IEEE Transactions on Intelligent Transportation Systems, Volume 5, Issue 4*. Institute of Electrical and Electronics Engineers, Parma, Italy. 2004.
 34. Gettman, D., and L. Head. Surrogate Safety Measures from Traffic Simulation Models. *Presented at the 82nd TRB Annual Meeting (CD-ROM)*, TRB, National Research Council, Washington DC, 2003.
 35. Helsinki University of Technology, Laboratory of Transportation Engineering.
<http://www.tkk.fi/Units/Transportation/HUTSIM/>. Accessed February 15, 2007
 36. Husch, D., and J. Albeck. *SimTraffic 6 User Guide*. Trafficware. Albany, California. 1997.
 37. VISSIM. PTV Vision America. <http://www.ptvamerica.com/vissim.html>. Accessed February 15, 2007.
 38. AIMSUN Simulator. AIMSUN NG. <http://www.tss-bcn.com/aimsun.html>. Barcelona, Spain. Accessed February 15, 2007.

39. Lee, E. L., and R. B. Machemehl. The TEXAS Model for Intersection Traffic – A User-Friendly Microcomputer Version with Animated Graphics Screen Display. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1142. TRB, National Research Council, Washington DC, 1987, pp 1 – 5.
40. Van Aerde, M., B. Hellinga, M. Baker, and H. Rakha. INTEGRATION: An Overview of Traffic Simulation Features. *Presented at the 73rd Annual Meeting of the Transportation Research Board (CD-ROM)*, TRB, National Research Council, Washington DC, 1996.
41. Gardes, Y., A. D. May, J. Dahlgren, and A. Skabardonis. Freeway Calibration and Application of the Paramics Model. *Presented at the 81st Annual Meeting of the Transportation Research Board (CD-ROM)*, TRB, National Research Council, Washington DC, 2002.
42. Kikuchi, S., N. Uno, and M. Tanaka. Impacts of Shorter Perception-Reaction Time of Adapted Cruise Controlled Vehicles on Traffic Flow and Safety. In *ASCE Journal of Transportation Engineering. Volume 129, Issue 2*. American Society of Civil Engineers, Reston, VA. 2003, pp. 109–221.
43. May, A. D. *Traffic Flow Fundamentals*. Prentice Hall. Upper Saddle River, NJ. 1990.
44. Halati, A., H. Lieu, and S. Walker. CORSIM – Corridor Traffic Simulation Model, In *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*. American Society of Civil Engineers, Reston, VA. 1997, pp. 570–576.
45. Mitchell, M. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, MA. 1997.
46. Fraser, A. S. Simulation of Genetic Systems by Automatic Digital Computers. In *Australian Journal of Biological Sciences, Vol. 10*. 1957, pp. 484-491.
47. Bagley, J. D. *The Behavior of Adaptive Systems which Employ Genetic and Correlation Algorithms*. Ph.D. dissertation, University of Michigan, Ann Arbor, 1967.
48. Holland, J. H. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor. 1975.
49. Haupt, R. L., and S. E. Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, Inc., Hoboken, NJ, 1998.
50. Davis, L. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
51. Park, B., and H. Qi. *Development and Evaluation Procedure for Microscopic Traffic Simulation Models*. Virginia Transportation Research Council, Charlottesville, VA.

- http://www.virginiadot.org/vtrc/main/online_reports/pdf/05-cr1.pdf. 2004. Accessed February 17, 2007.
52. Schultz, G. G. *Developing a Methodology to Account for Commercial Motor Vehicles Using Microscopic Traffic Simulation Models*. Ph.D. dissertation, Texas A&M University, College Station, 2003.
 53. Schultz, G. G., and L. R. Rilett. Analysis of Distribution and Calibration of Car-Following Sensitivity Parameters in Microscopic Traffic Simulation Models. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1876*, TRB, National Research Council, Washington DC, 2004, pp 41–51.
 54. *Utah: 2000. Summary Population and Housing Characteristics*. Issued September 2002. United States Census Bureau, Washington, DC.
<http://www.census.gov/prod/cen2000/phc-1-46.pdf>. Accessed February 17, 2007.
 55. I-15 Corridor Problem, Purpose, and Goals. Utah Department of Transportation, Salt Lake City, UT.
<http://www.udot.utah.gov/i15utahcounty/downloads/ProblemPurposeAndGoals.pdf>
Accessed January 31, 2007.
 56. Utah Department of Transportation. *Traffic on Utah Highways 2005*. Salt Lake City, UT. 2006.
 57. Utah Department of Transportation. *1993 Traffic Volume Report: Average Weekday*. Salt Lake City, UT. 1994.
 58. Roess, R. P., E. S. Prassas, and W. R. McShane. *Traffic Engineering, Third Edition*. Prentice Hall, Upper Saddle River, NJ. 2004.
 59. Dutch, S. UTM Conversion Spreadsheet. Green Bay, Wisconsin.
<http://www.uwgb.edu/dutchs/UsefulData/UTMConversions1.xls>. Accessed January 31, 2007.
 60. Snyder, John P., *Map Projections – A Working Manual*. U.S. Geological Survey Professional Paper 1395. United States Government Printing Office, Washington, DC. 1987.

Appendix A. Sample Input File for Calibration Program

```
Input File for Genetic Algorithm Calibration of CORSIM
Date Created; Saturday, April 29, 2006

MOE;volume

Files;
Observed Volumes;C:\Documents and Settings\ct224\Desktop\Thesis\nb_no_ramp_link.txt
Observed Links;
Observed Travel Times;
CORSIM Input;C:\Documents and Settings\ct224\Desktop\Thesis\nb_nohov_OD.trf

Output Directory;C:\Documents and Settings\ct224\Desktop\15_pop

Calibration Control;
Maximum Number of Iterations;100
Population Size;15
Probability of Mutation;0.3
Probability of Crossover;0.5

Fitness Function;
Alpha;0.5
Travel Time Beta;2.5
Volume Beta;5

Calibration Parameters;
Car Following Sensitivity Distribution;lognormal
      prec. Min   Max   No
Car Following mean           ;0.1 30 150 1
Car Following standard distribution ;0.1 2 50 1
Pitt Constant                ;1 3 10 1
Lag to accelerate            ;1 1 10 1
Lag to decelerate           ;1 1 10 1
Time to complete lane change ;1 10 60 1
Vehicle generation separation ;1 10 30 1
Collision Avoidance time     ;1 1 6 1
Percent Yielding to Mergers  ;1 5 40 1
Lane Change Desire          ;1 1 10 1
Lane change advantage        ;1 1 10 1
Max non-emergency deceleration for veh 1 ;1 70 120 1
Max non-emergency deceleration for veh 2 ;1 70 120 1
Max non-emergency deceleration for veh 3 ;1 40 100 1
Max non-emergency deceleration for veh 4 ;1 40 100 1
Max non-emergency deceleration for veh 5 ;1 40 100 1
Max non-emergency deceleration for veh 6 ;1 40 100 1
Max non-emergency deceleration for veh 7 ;1 40 100 1
Max non-emergency deceleration for veh 8 ;1 70 120 1
Max non-emergency deceleration for veh 9 ;1 70 120 1
```


Appendix B. Sample Files from Calibration Program

B.1: Sample CORSIM Input File

```

Created by TSIS Fri Feb 02 15:41:57 2007 from TNO Version 61
Created by TSIS Wed Jan 17 17:55:14 2007 from TNO Version 52
Created by TSIS Thu Nov 02 17:56:47 2006 from TNO Version 52
Created by TSIS Wed Jan 30 01:11:28 2002 from TNO Version 52
Created by TSIS Tue Jan 29 22:06:08 2002 from TNO Version 52
Created by TSIS Tue Jan 29 15:15:04 2002 from TNO Version 52
12345678 1 2345678 2 2345678 3 2345678 4 2345678 5 2345678 6 2345678 7 234567
12 262001Wilbur Smith Associates 0 1
1 0 0 15 7981 0606 1 8 445 7781 7581 2
3600 3
60 4
0 0 0 0 0 0 0 0 0 0 1 CORSIM 5
266 15 8 32550 3 91 1627 92 1628 1 9 19
765 266 15 20880 3 1 19
264 765 266 35580 3 92 400 1 9 19
263 264 765 10400 3 1 19
7 266 15 3541 1 9 19
762 263 264 17550 3 91 1730101 874 1 19
261 762 263 19000 3 1 19
1 762 263 4791 3 10 19
260 261 762 40340 3 92 1500 1 9 19
15 8 14 24670 3 1 19
10 8 14 10881 1 9 19
15 98006 10671 1 1 19
8 14 17 26530 3 91 500 92 200 1 9 19
14 17 727 25840 3 1 19
18 17 727 9861 1 9 19
17 727 29106350 3 91 500 92 800 1 9 19
727 29 739 31970 3 1 19
727 358012 15081 1 1 19
37 29 739 12611 1 9 19
29 739 59 30940 3 91 800 92 500 1 9 19
739 59 68 27470 3 1 19
739 648017 12901 1 1 19
63 59 68 12261 1 9 19
59 68 777 42500 3 91 500 1 19
68 777 72 40920 3 92 500 1 9 19
777 72 774 29140 3 1 19
777 678020 12821 1 1 19
53 72 774 14471 1 9 19
72 774 76 33930 3 91 500 92 300 1 9 19
774 76 793 25740 3 1 19
774 798026 12611 1 1 19
78 76 793 11751 1 9 19
76 793 92107860 3 91 300 92 500 1 9 19
793 92 98 36560 3 1 19
793 898033 19071 1 1 19
87 92 98 13991 1 9 19

98 97 99 34970 3 1 19
98 948029 20631 1 1 19

```

85	97	99	12371	1			9					19
97	998040	74950	3	91	200		1					19
14	198008	10791	1				1					19
765	28005	4001	1				1					19
261	38001	3541	1				1					19
8004	260	261	0	3			1					19
8007	10	8	1	1			1					19
8009	18	17	1	1			1					19
8013	37	29	1	1			1					19
8018	63	59	1	1			1					19
8019	53	72	1	1			1					19
8025	78	76	1	1			1					19
8034	87	92	1	1			1					19
8024	85	97	1	1			1					19
8003	7	266	1	1			1					19
8002	1	762	1	3			1					19
266	15	0	0	41065	1375	1375			431500	100		20
765	266	0	0	41065						100		20
264	765	0	0	41065	2500	2500				100		20
263	264	0	0	41065						100		20
7	266	0	0	11060						100		20
762	263	0	0	41065					431500	100		20
261	762	0	0	41065						100		20
1	762	0	0	11055						100		20
260	261	0	0	41065	2500	2500				100		20
15	8	0	0	41065						100		20
10	8	0	0	11035						100		20
15	9	0	0	11035						100		20
8	14	0	0	41065	1085	1085			431500	100		20
14	17	0	0	41065						100		20
18	17	0	0	11035						100		20
17	727	0	0	41065	1182	1182			431500	100		20
727	29	0	0	41065						100		20
727	35	0	0	11035						100		20
37	29	4	0	11035						100		20
29	739	0	0	41065	1004	1004			431500	100		20
739	59	0	0	41065						100		20
739	64	0	0	11065						100		20
63	59	0	0	11065						100		20
59	68	0	0	41065					431500	100		20
68	777	0	0	41065	1092	1092				100		20
777	72	0	0	41065						100		20
777	67	0	0	11065						100		20
53	72	0	0	11065						100		20
72	774	0	0	41065	1056	1056			431500	100		20
774	76	0	0	41065						100		20
774	79	0	0	11065						100		20
78	76	0	0	11065						100		20
76	793	0	0	41065	1151	1151			431500	100		20
793	92	0	0	41065						100		20
793	89	0	0	11065						100		20
87	92	0	0	11065						100		20
92	98	0	0	41065	1121	1121			431500	100		20
98	97	0	0	41065						100		20
98	94	0	0	11065						100		20
85	97	0	0	11065						100		20
97	99	0	0	41065					431500	100		20
14	19	0	0	11035						100		20
765	2	0	0	11065						100		20
261	3	0	0	11065						100		20
8004	260	0	0	11065								20
8007	10	0	0	11065								20
8009	18	0	0	11065								20
8013	37	0	0	11065								20
8018	63	0	0	11065								20
8019	53	0	0	11065								20
8025	78	0	0	11065								20
8034	87	0	0	11065								20
8024	85	0	0	11065								20
8003	7	0	0	11065								20

22	16	3	5	2	4	80	80	80	80	80	80	80	80	80	80	15	70
0																	170
8006	57799		26572														195
8012	45070		39719														195
8017	39575		42843														195
8020	29877		47769														195
8026	24197		50098														195
8033	16080		61196														195
8029	10993		67516														195
8040	3400		77800														195
8008	55641		31544														195
8005	57947		20504														195
8001	60027		13004														195
8004	63980		6007														195
8007	57862		26939														195
8009	55202		32011														195
8013	44238		40303														195
8018	38765		43161														195
8019	28554		48168														195
8025	23635		50927														195
8034	15353		62014														195
8024	10695		68769														195
8003	57635		21864														195
8002	59135		13767														195
1	58979		13807														195
2	57597		20479														195
3	59752		12904														195
7	57420		21899														195
8	57304		27966														195
9	57434		26557														195
10	57448		26887														195
14	55844		30181														195
15	57300		25499														195
17	53907		31893														195
18	54835		31558														195
19	55275		31098														195
727	45306		38149														195
29	42691		39991														195
35	44400		39355														195
37	43903		39641														195
739	40005		41526														195
53	28592		46849														195
59	37527		42715														195
63	38729		42468														195
64	38992		42326														195
67	28827		46740														195
68	33761		44685														195
72	27154		47021														195
774	24183		48660														195
76	22467		50579														195
777	29894		46028														195
78	23422		49894														195
79	23582		49769														195
793	15458		58778														195
94	9984		67758														195
97	8743		68725														195
98	10685		65817														195
99	4800		75100														195
260	61868		9165														195
261	59683		12557														195
762	58642		14147														195
263	57679		15614														195
264	57403		16598														195
765	57361		20156														195
266	57340		22244														195
263	264	01															196
1	0	0															210

B.2: Sample OBJECTIVE_output.txt

```
Itr Avg_Fit Max_Fit Min_Fit Time_Recorded
1 75.055377.369273.3506 Sat Feb 10 12:34:48 2007
2 75.681077.369274.5545 Sat Feb 10 12:48:48 2007
3 75.828177.369274.6448 Sat Feb 10 13:04:09 2007
4 76.265777.369275.4096 Sat Feb 10 13:23:23 2007
5 76.268177.369275.4583 Sat Feb 10 13:41:06 2007
6 76.268177.369275.4583 Sat Feb 10 13:57:19 2007
7 76.277177.369275.5979 Sat Feb 10 14:10:49 2007
8 76.284277.369275.6377 Sat Feb 10 14:21:40 2007
9 76.327277.369275.7487 Sat Feb 10 14:43:12 2007
10 76.562977.369275.8969 Sat Feb 10 15:05:57 2007
11 76.655977.369275.9280 Sat Feb 10 15:19:13 2007
12 76.683177.369275.9280 Sat Feb 10 15:35:27 2007
13 76.721077.369275.9280 Sat Feb 10 15:54:09 2007
14 76.721077.369275.9280 Sat Feb 10 16:04:53 2007
15 76.721077.369275.9280 Sat Feb 10 16:25:20 2007
16 76.792977.369276.1189 Sat Feb 10 16:42:38 2007
17 76.897977.369276.2374 Sat Feb 10 17:06:21 2007
18 76.963677.369276.3071 Sat Feb 10 17:17:03 2007
19 76.963677.369276.3071 Sat Feb 10 17:32:57 2007
20 76.963677.369276.3071 Sat Feb 10 17:54:36 2007
21 76.980177.369276.3071 Sat Feb 10 18:06:16 2007
22 77.025477.369276.4721 Sat Feb 10 18:20:14 2007
23 77.025477.369276.4721 Sat Feb 10 18:37:48 2007
24 77.025477.369276.4721 Sat Feb 10 18:54:49 2007
25 77.098177.924476.4721 Sat Feb 10 19:13:08 2007
26 77.098177.924476.4721 Sat Feb 10 19:19:41 2007
27 77.163377.924476.4721 Sat Feb 10 19:36:50 2007
28 77.163377.924476.4721 Sat Feb 10 19:58:32 2007
29 77.163377.924476.4721 Sat Feb 10 20:15:51 2007
30 77.167077.924476.4721 Sat Feb 10 20:29:00 2007
31 77.288377.924476.9080 Sat Feb 10 20:48:51 2007
32 77.310977.924476.9080 Sat Feb 10 21:08:02 2007
33 77.310977.924476.9080 Sat Feb 10 21:22:10 2007
34 77.310977.924476.9080 Sat Feb 10 21:39:04 2007
35 77.334377.924476.9080 Sat Feb 10 21:53:11 2007
36 77.334377.924476.9080 Sat Feb 10 22:09:34 2007
37 77.334377.924476.9080 Sat Feb 10 22:28:28 2007
38 77.357477.924477.1341 Sat Feb 10 22:42:28 2007
39 77.357477.924477.1341 Sat Feb 10 23:03:19 2007
40 77.357477.924477.1341 Sat Feb 10 23:25:03 2007
```

B.3: Sample RUN_SITUATION.txt

```
ITR CHROM TIME_FINISHED
0 1 Sat Feb 10 11:57:02 2007
0 2 Sat Feb 10 11:58:01 2007
0 3 Sat Feb 10 11:58:57 2007
0 4 Sat Feb 10 11:59:54 2007
0 5 Sat Feb 10 12:00:53 2007
0 6 Sat Feb 10 12:01:51 2007
0 7 Sat Feb 10 12:02:47 2007
```

```

0 8 Sat Feb 10 12:03:45 2007
0 9 Sat Feb 10 12:04:42 2007
0 10 Sat Feb 10 12:05:38 2007
1 1 Sat Feb 10 12:16:13 2007
1 2 Sat Feb 10 12:17:09 2007
1 3 Sat Feb 10 12:18:06 2007
1 4 Sat Feb 10 12:19:04 2007
2 1 Sat Feb 10 12:37:14 2007
2 2 Sat Feb 10 12:38:09 2007
2 3 Sat Feb 10 12:39:06 2007
3 1 Sat Feb 10 12:54:16 2007
3 2 Sat Feb 10 12:55:15 2007
3 3 Sat Feb 10 12:56:14 2007
3 4 Sat Feb 10 12:57:13 2007
3 5 Sat Feb 10 12:58:11 2007
3 6 Sat Feb 10 12:59:10 2007
3 7 Sat Feb 10 13:00:08 2007
4 1 Sat Feb 10 13:07:59 2007
4 2 Sat Feb 10 13:08:57 2007
4 3 Sat Feb 10 13:09:55 2007
4 4 Sat Feb 10 13:10:54 2007
5 1 Sat Feb 10 13:24:23 2007
5 2 Sat Feb 10 13:25:23 2007
5 3 Sat Feb 10 13:26:22 2007

```

B.4: Sample time_logfile.txt

Genetic Algorithm Program was started at Sat Feb 10 12:08:59 2007

ITR	Chrom	Sum	Error	Pct_Error	Fitness
0 1	0.9177	0.1147	11.471775	0.0668	
0 2	0.9067	0.1133	11.333775	0.3262	
0 3	1.1098	0.1387	13.872370	0.6941	
0 4	0.8442	0.1055	10.552176	0.8125	
0 5	0.8656	0.1082	10.819976	0.2999	
0 6	0.8864	0.1108	11.080075	0.8054	
0 7	0.8116	0.1014	10.144577	0.5993	
0 8	0.8636	0.1079	10.794976	0.3476	
0 9	1.3671	0.1709	17.089265	0.2313	
0 10	0.9401	0.1175	11.751674	0.5434	
0 11	0.8520	0.1065	10.650076	0.6248	
0 12	0.8895	0.1112	11.118575	0.7325	
1 1	0.8498	0.1062	10.622176	0.6783	
1 2	0.8953	0.1119	11.190775	0.5959	
1 3	0.9401	0.1175	11.751674	0.5434	
1 4	0.8953	0.1119	11.190775	0.5959	
1 5	0.9177	0.1147	11.471775	0.0668	
1 6	0.8304	0.1038	10.380177	0.1435	
1 7	0.9316	0.1164	11.644774	0.7427	
1 8	0.8304	0.1038	10.380177	0.1435	
1 9	0.9401	0.1175	11.751674	0.5434	
1 10	1.3671	0.1709	17.089265	0.2313	
2 1	0.8636	0.1079	10.794976	0.3476	
2 2	0.8272	0.1034	10.339777	0.2215	
2 3	0.8116	0.1014	10.144577	0.5993	
2 4	0.9592	0.1199	11.989974	0.1005	

2 5	0.9177	0.1147	11.4717	75.0668
2 6	0.8272	0.1034	10.3397	77.2215
3 1	0.9113	0.1139	11.3915	75.2174
3 2	0.8388	0.1049	10.4853	76.9409
3 3	0.9113	0.1139	11.3915	75.2174
3 4	0.9536	0.1192	11.9203	74.2295
3 5	0.8596	0.1075	10.7455	76.4420
3 6	0.8442	0.1055	10.5521	76.8125
3 7	0.8715	0.1089	10.8939	76.1590
4 1	0.9421	0.1178	11.7763	74.4973
4 2	0.8773	0.1097	10.9664	76.0211
4 3	0.9014	0.1127	11.2676	75.4508
4 4	0.9067	0.1133	11.3337	75.3262
5 1	0.9401	0.1175	11.7516	74.5434
5 2	1.1098	0.1387	13.8723	70.6941

B.5: Sample volume_logfile.txt

Genetic Algorithm Program was started at Sat Feb 10 12:08:59 2007

ITR	Chrom	Sum	Error	Pct_Error	Fitness
0 1	1.1182	0.0589	5.8852	74.5084	
0 2	1.3367	0.0704	7.0351	70.3452	
0 3	0.9617	0.0506	5.0615	77.6411	
0 4	1.1832	0.0623	6.2274	73.2443	
0 5	1.2308	0.0648	6.4780	72.3324	
0 6	1.0640	0.0560	5.5999	75.5789	
0 7	1.1766	0.0619	6.1925	73.3723	
0 8	1.1313	0.0595	5.9541	74.2521	
0 9	1.1900	0.0626	6.2630	73.1139	
0 10	0.9101	0.0479	4.7899	78.7025	
0 11	1.0689	0.0563	5.6261	75.4800	
1 1	0.9576	0.0504	5.0402	77.7236	
1 2	1.1112	0.0585	5.8483	74.6461	
1 3	0.9101	0.0479	4.7899	78.7025	
1 4	1.1112	0.0585	5.8483	74.6461	
1 5	1.1182	0.0589	5.8852	74.5084	
1 6	1.2192	0.0642	6.4170	72.5533	
1 7	1.2028	0.0633	6.3304	72.8680	
1 8	1.2192	0.0642	6.4170	72.5533	
1 9	0.9101	0.0479	4.7899	78.7025	
2 1	1.1313	0.0595	5.9541	74.2521	
2 2	1.0112	0.0532	5.3223	76.6353	
2 3	1.1766	0.0619	6.1925	73.3723	
2 4	0.9710	0.0511	5.1103	77.4517	
2 5	1.1182	0.0589	5.8852	74.5084	
2 6	1.0112	0.0532	5.3223	76.6353	
2 7	1.0744	0.0565	5.6548	75.3717	
3 1	1.0799	0.0568	5.6839	75.2622	
3 2	0.9917	0.0522	5.2197	77.0294	
4 1	1.1568	0.0609	6.0885	73.7548	
4 2	1.0005	0.0527	5.2659	76.8513	
4 3	1.2619	0.0664	6.6416	71.7429	
4 4	1.3367	0.0704	7.0351	70.3452	
5 1	0.9101	0.0479	4.7899	78.7025	
5 2	0.9617	0.0506	5.0615	77.6411	

5 3	1.0043	0.0529	5.2857	76.7753
5 4	1.1225	0.0591	5.9079	74.4236
5 5	1.0675	0.0562	5.6183	75.5091
5 6	1.1551	0.0608	6.0792	73.7890
5 7	1.1313	0.0595	5.9541	74.2521
6 1	0.9710	0.0511	5.1103	77.4517
6 2	1.3540	0.0713	7.1261	70.0258

B.6: Sample both_logfile.txt

Genetic Algorithm Program was started at Sat Feb 10 12:08:59 2007

ITR Chrom Fit_Time Fit_VolFit_Both

0 1	75.0668	74.5084	74.5084
0 2	75.3262	70.3452	70.3452
0 3	70.6941	77.6411	77.6411
0 4	76.8125	73.2443	73.2443
0 5	76.2999	72.3324	72.3324
0 6	75.8054	75.5789	75.5789
0 7	77.5993	73.3723	73.3723
0 8	76.3476	74.2521	74.2521
0 9	65.2313	73.1139	73.1139
0 10	74.5434	78.7025	78.7025
0 11	76.6248	75.4800	75.4800
0 12	75.7325	72.8712	72.8712
1 1	76.6783	77.7236	77.7236
1 2	75.5959	74.6461	74.6461
1 3	74.5434	78.7025	78.7025
1 4	75.5959	74.6461	74.6461
1 5	75.0668	74.5084	74.5084
1 6	77.1435	72.5533	72.5533
1 7	74.7427	72.8680	72.8680
1 8	77.1435	72.5533	72.5533
2 1	76.3476	74.2521	74.2521
2 2	77.2215	76.6353	76.6353
2 3	77.5993	73.3723	73.3723
2 4	74.1005	77.4517	77.4517
2 5	75.0668	74.5084	74.5084
2 6	77.2215	76.6353	76.6353
2 7	76.5010	75.3717	75.3717
2 8	75.8467	73.3616	73.3616
3 1	75.2174	75.2622	75.2622
3 2	76.9409	77.0294	77.0294
3 3	75.2174	75.2622	75.2622
3 4	74.2295	75.9198	75.9198
4 1	74.4973	73.7548	73.7548
4 2	76.0211	76.8513	76.8513
4 3	75.4508	71.7429	71.7429
4 4	75.3262	70.3452	70.3452
4 5	76.6507	72.0145	72.0145
5 1	74.5434	78.7025	78.7025
5 2	70.6941	77.6411	77.6411
5 3	75.6179	76.7753	76.7753
5 4	76.8291	74.4236	74.4236
5 5	74.0762	75.5091	75.5091

Genetic Algorithm Program was started at Thu Feb 8 15:38:35 2007

```
##### ITERATION: 0 #####
Chrom P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22 P23 P24 P25
      P26 P27 P28
1 150 150 150 150 150 128 74 35 30 30 30 10 3 3 42 17 3 20 5 6 80 80 80 80 80 80 80 80
2 150 150 150 150 121 74 43 30 30 30 30 10 3 3 21 28 3 20 1 4 80 80 80 80 80 80 80 80
3 150 150 150 150 150 135 84 38 30 30 30 10 3 3 13 26 3 20 1 9 80 80 80 80 80 80 80 80
4 150 150 150 150 118 61 32 30 30 30 30 10 3 3 13 16 3 20 5 5 80 80 80 80 80 80 80 80
5 150 150 150 150 150 142 94 41 30 30 30 10 3 3 40 26 3 20 2 5 80 80 80 80 80 80 80 80
6 150 150 150 150 150 149 115 63 37 30 30 10 3 3 40 19 3 20 5 10 80 80 80 80 80 80 80 80
7 150 150 150 150 150 127 75 38 30 30 30 10 3 3 50 26 3 20 8 1 80 80 80 80 80 80 80 80
8 150 150 145 103 47 30 30 30 30 10 3 3 24 25 3 20 4 8 80 80 80 80 80 80 80 80
9 150 150 150 150 150 114 56 31 30 30 30 10 3 3 39 16 3 20 4 10 80 80 80 80 80 80 80 80
10 150 150 150 150 150 138 88 39 30 30 30 10 3 3 57 23 3 20 1 4 80 80 80 80 80 80 80 80
```

```
##### ITERATION: 1 #####
Chrom P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22 P23 P24 P25
      P26 P27 P28
1 150 150 150 150 123 70 36 30 30 30 10 3 3 33 24 3 20 6 6 80 80 80 80 80 80 80 80
2 150 150 150 150 127 75 38 30 30 30 10 3 3 50 26 3 20 8 3 80 80 80 80 80 80 80 80
3 150 150 150 150 127 75 38 30 30 30 10 3 3 56 28 3 20 8 1 80 80 80 80 80 80 80 80
4 150 150 150 150 138 88 39 30 30 30 10 3 3 51 21 3 20 1 4 80 80 80 80 80 80 80 80
5 150 150 150 150 127 75 38 30 30 30 10 3 3 50 26 3 20 8 1 80 80 80 80 80 80 80 80
6 150 150 150 150 114 56 31 30 30 30 10 3 3 39 16 3 20 5 10 80 80 80 80 80 80 80 80
7 150 150 150 134 84 40 30 30 30 30 10 3 3 16 25 3 20 4 1 80 80 80 80 80 80 80 80
```

```
##### ITERATION: 2 #####
Chrom P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22 P23 P24 P25
      P26 P27 P28
1 150 150 150 124 69 35 30 30 30 30 10 3 3 50 23 3 20 6 2 80 80 80 80 80 80 80 80
2 150 150 150 150 138 88 39 30 30 30 10 3 3 57 23 3 20 1 4 80 80 80 80 80 80 80 80
3 150 150 150 150 114 56 31 30 30 30 10 3 3 39 16 3 20 9 10 80 80 80 80 80 80 80 80
4 150 150 150 150 148 103 44 30 30 30 10 3 3 47 14 3 20 5 10 80 80 80 80 80 80 80 80
5 150 150 150 150 130 89 53 34 30 30 10 3 3 29 12 3 20 1 4 80 80 80 80 80 80 80 80
```

B.8: Sample Play_by_Play.txt (excerpt)

```
Entering selection sub, Fitness values are: 1: 73.0460537548437
2: 74.4661833895158
3: 73.5425125442163
4: 76.1462854828598
5: 73.0460537548437
```

```
After selection, fitness values are: 1: 74.4661833895158
2: 73.0460537548437
3: 73.0460537548437
4: 73.0460537548437
5: 73.0460537548437
```

```
After initialization:
111110101000000100111000100100010001011
110110111100000101110101111101100101001
110110111100000101110101111101100101001
110111010000101001111000110000101011001
110110111100000101110101111101100101001
```

Just started the Crossover sub.

```
Chromosome 1 not selected.
Chromosome 2 not selected.
After Crossover:
111110101000000100111000100100010001011
110110111100000101110101111101100101001
110110111100000101110101111101100101001
110111010000101001111000110000101011001
110110111100000101110101111101100101001
```

```
Starting to mutate!.
Chromosome 1 not selected.
Chromosome 2 not selected.
Chromosome 3 not selected.
Chromosome 4 not selected.
Chromosome 5 not selected.
Size new set to 0.
```

```
After Mutation:
111110101000000100111000100100010001011
110110111100000101110101111101100101001
110110111100000101110101111101100101001
110111010000101001111000110000101011001
110110111100000101110101111101100101001
```

```
Entering Evaluation subroutine!.
Calculated a fitness function for the first 0 guys.
They are:
Now, the entire fitness array is:
73.0460537548437
74.4661833895158
73.5425125442163
76.1462854828598
73.0460537548437
```

Make an array of only the last 0 guys
Sorted the values of FITNESS_FINAL. Now they are:
76.1462854828598
74.4661833895158
73.5425125442163
73.0460537548437
73.0460537548437

Changed the order of the chromosomes according to fitness values.
After evaluation:

```
110111010000101001111000110000101011001
110110111100000101110101111101100101001
110110110001100010010110001101011110101
100110101000001111001010001001001100010
110111010000101001111000110000101011001
```

Entering selection sub, Fitness values are: 1: 76.1462854828598
2: 74.4661833895158
3: 73.5425125442163
4: 73.0460537548437
5: 73.0460537548437

After selection, fitness values are: 1: 73.0460537548437
2: 73.5425125442163
3: 73.0460537548437
4: 76.1462854828598
5: 76.1462854828598

After selection:
110111010000101001111000110000101011001
110110111100000101110101111101100101001
110110110001100010010110001101011110101
100110101000001111001010001001001100010
110111010000101001111000110000101011001

Just started the Crossover sub.

Chromosome 1 not selected.
Chromosome 2 selected.
Copied chromosomes 1 and 4 into 1 and 2.
Crossed over bits 31 through 33 between 1 and 2.
After Crossover:

```
110111010000101001111000110000101011001
110111010000101001111000110000101011001
110111010000101001111000110000101011001
110111010000101001111000110000101011001
110111010000101001111000110000101011001
```

Starting to mutate!.
Chromosome 1 not selected.
Chromosome 2 not selected.
Chromosome 3 not selected.

Appendix C. Traffic Volume Counts from UDOT

Table C.1: Traffic Volume Counts from UDOT

Peak Hour Traffic Volumes	Northbound			Southbound			Eastbound			Westbound		
	Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru	Right
284 Alpine												
SB Ramps	0	0	0	1160	0	77	0	312	187	210	324	0
NB Ramps	214	0	116	0	0	0	112	1359	0	0	321	604
E. Frontage Rd & SR-92	0	0	0	24	0	242	225	1247	0	0	683	30
282 Lehi 1200 West												
SB Off Ramp & 1200 W	0	0	917	219	925	25	0	177	171	540	109	0
SB On Ramp & US-89	99	917	49	193	1441	2	0	0	0	0	0	0
NB Ramps & 1200 W.	82	0	89	0	0	0	796	517	0	0	567	0
2100 N & 1200 W	107	5	9	0	0	0	0	418	188	65	460	91
279 Lehi Main Street												
850 E & Main St	154	78	222	139	179	35	54	908	119	274	1205	250
SB Ramps & Main St	0	0	0	313	3	192	0	781	487	137	1538	0
NB Ramps & Main St	1024	8	213	0	0	0	222	872	0	0	653	185
278 American Fork Main Street												
SB Ramps & Main St	0	0	0	400	4	50	0	376	179	259	587	0
NB Ramps & Main St	248	11	428	0	0	0	51	725	0	0	600	366
US-89 & Main St	28	13	28	846	14	257	333	798	21	12	682	593
276 American Fork 500 East												
SB Ramps & 500 East	0	0	0	231	118	213	0	420	91	365	313	0
NB Ramps & 500 East	14	0	524	0	0	0	164	487	0	0	665	389
620 South & 500 East	246	17	113	24	36	34	64	903	43	111	775	31
275 Pleasant Grove												
SB Ramps & P. G. Blvd	0	0	0	273	7	25	0	177	200	483	72	0
NB Ramps & P. G. Blvd	40	5	279	0	0	0	111	339	0	0	516	419
273 1600 North Orem												
SB Ramps & 1600 N.	0	0	0	539	0	115	0	306	289	509	345	0
NB On Ramp & 1600 N.	0	0	0	0	0	0	116	728	0	0	854	724
NB Off Ramp & 1600 N.	203	107	322	109	0	386	86	642	0	0	980	102
272 800 North Orem												
SB Ramps & 800 N.	0	0	0	856	0	58	0	468	204	558	388	0
NB Ramps & 800 N.	120	0	622	0	0	0	113	1211	0	0	817	830
1200 West & 800 N.	187	210	115	62	152	100	125	1458	250	83	1362	74
271 Orem Center Street												
SB Ramps & Center St.	0	0	0	657	6	114	0	538	241	596	283	0
NB Ramps & Center St.	52	3	494	0	0	0	176	1019	0	0	828	807
1200 West & Center St.	258	288	107	85	196	198	143	1129	240	103	1179	124
269 University Parkway												
Ramps & University Pkwy	222	0	901	1521	0	230	260	551	241	998	476	1476
Sandhill Rd & University Pkwy	414	256	342	180	241	558	554	1867	551	380	1979	159

Table D.1: Default Ranges and Results of Sensitivity Analysis

	Average Content	Speed (mph)	Vehicle-Miles	Vehicle-Minutes	Moving Time/Total Trip Time	Current Content	Total Delay (veh-min)	Travel Time (min/veh-mile)	Delay Time (min/veh-mile)
Default Range	62.9	1.3	2171.7	3773.9	0.021	106.0	2351.8	0.02	0.03
Mandatory Lane Change Gap Acceptance	24.7	0.3	973.0	1483.5	0.004	26.0	538.8	0.01	0.00
Time to Complete Lane Change Maneuver	92.7	4.0	1251.9	5560.8	0.060	59.0	6611.8	0.08	0.07
Percent of Drivers Yielding to Merging Vehicles	22.8	0.2	948.8	1364.7	0.003	6.0	436.4	0.01	0.00
Minimum Separation for Generation of Vehicles	206.3	1.5	9857.7	12376.5	0.024	243.0	3250.1	0.02	0.03
Multiplier for Desire to Make Discretionary Lane Change	27.9	1.3	532.6	1674.1	0.018	77.0	1972.1	0.03	0.02
Advantage Threshold for Discretionary Lane Change	940.2	27.9	17805.7	56412.2	0.427	1893.0	71919.7	0.98	0.96
Pitt's Car Following Constant	22.9	1.1	483.6	1373.4	0.019	74.0	2017.9	0.02	0.02
Maximum Non-emergency Freeway Deceleration	13.1	1.0	728.8	788.6	0.013	55.0	1245.2	0.01	0.01
Acceleration Lag Time	37.9	0.3	1714.7	2271.1	0.004	11.0	536.8	0.00	0.01
Deceleration Lag Time	21.4	0.7	113.8	1284.8	0.010	67.0	1096.0	0.01	0.01
Car Following Sensitivity Factors	597.1	20.4	6622.4	35828.2	0.311	1024.0	41342.6	0.49	0.48

Table D.2: Ranges of MOEs from Varying Parameter Values

Parameter	Parameter Values	Network MOEs								
		Average Content	Speed (mph)	Vehicle Miles	Vehicle Minutes	Time/Total Trip Time	Current Content	Delay (veh-min)	(min/veh-mile)	Delay Time (min/veh-mile)
Multiplier for desire to make discretionary lane	1	1629.1	57.57	93795.3	97748.8	0.896	1644	10155.79	1.04	0.11
	10	1657.0	56.28	93262.7	99422.9	0.878	1721	12127.86	1.07	0.13
	<i>Range</i>	9	27.9	1.29	532.6	1674.1	0.018	77	1972.07	-0.03
Advantage threshold for discretionary lane change	1	2580.7	29.76	76800.3	154841.1	0.471	3642	81925.27	2.02	1.07
	10	1640.5	57.67	94606.0	98428.9	0.898	1749	10005.55	1.04	0.11
	<i>Range</i>	9	940.2	27.91	17805.7	56412.2	0.427	1893	71919.72	0.98
Mandatory lane change gap acceptance	1	1615.5	58.19	93997.7	96929.2	0.906	1675	9146.19	1.03	0.10
	6	1640.2	57.90	94970.7	98412.7	0.902	1649	9684.97	1.04	0.10
	<i>Range</i>	5	24.7	0.29	973.0	1483.5	0.004	26	538.78	-0.01
Time to complete lane change maneuver	10	1690.9	54.74	92561.3	101455.3	0.854	1739	14861.59	1.10	0.16
	60	1598.2	58.70	93813.2	95894.5	0.914	1680	8249.77	1.02	0.09
	<i>Range</i>	50	92.7	3.96	1251.9	5560.8	0.060	59	6611.82	0.08
Percent of drivers yielding to merging	5	1615.0	57.98	93646.8	96901.1	0.903	1654	9430.36	1.03	0.10
	40	1637.8	57.76	94595.6	98265.8	0.900	1660	9866.74	1.04	0.10
	<i>Range</i>	35	22.8	0.22	948.8	1364.7	0.003	6	436.38	-0.01
Minimum separation for generation of vehicles	10	1423.4	59.58	84805.2	85405.2	0.928	1492	6133.62	1.01	0.07
	30	1629.7	58.09	94662.9	97781.7	0.904	1735	9383.72	1.03	0.10
	<i>Range</i>	20	206.3	1.49	9857.7	12376.5	0.024	243	3250.10	-0.02
Pitt's car following constant	3	1590.1	59.35	94373.7	95405.7	0.924	1583	7215.49	1.01	0.08
	10	1613.0	58.21	93890.1	96779.1	0.905	1657	9233.37	1.03	0.10
	<i>Range</i>	7	22.9	1.14	483.6	1373.4	0.019	74	2017.88	-0.02
Maximum non-emergency deceleration	40	1570.3	60.13	94420.6	94215.3	0.935	1587	6151.96	1.00	0.07
	120	1583.4	59.17	93691.8	95003.9	0.922	1642	7397.16	1.01	0.08
	<i>Range</i>	80	13.1	0.96	728.8	788.6	0.013	55	1245.20	-0.01
Acceleration lag time	1	1568.3	59.61	93483.4	94100.6	0.927	1610	6843.52	1.01	0.07
	10	1606.2	59.27	95198.1	96371.7	0.923	1621	7380.28	1.01	0.08
	<i>Range</i>	9	37.9	0.34	1714.7	2271.1	0.004	11	536.76	0.00
Deceleration lag time	1	1650.5	56.99	94055.0	99029.5	0.888	1709	11058.09	1.05	0.12
	10	1629.1	57.67	93941.2	97744.7	0.898	1642	9962.09	1.04	0.11
	<i>Range</i>	9	21.4	0.68	113.8	1284.8	-0.010	67	1096.00	0.01
Car following sensitivity factors	30	2118.8	40.78	86408.7	127128.7	0.639	2568	45905.67	1.47	0.53
	150	1521.7	61.14	93031.1	91300.5	0.950	1544	4563.04	0.98	0.05
	<i>Range</i>	120	597.1	20.36	6622.4	35828.2	0.311	1024	41342.63	0.49