

ABSTRACT

TRASK, JOSEPH LAKE. Optimization Methods for Calibration and Analysis of Congested Freeway Facilities. (Under the direction of John Baugh and Nagui Rouphail.)

Congestion and delays on freeways result in countless hours of frustration for drivers and cost the nation billions of dollars each year. Consequently, projects are constantly being undertaken to both improve the physical capacities of roadways and develop new strategies for demand management and congestion reduction. A variety of planning and analysis tools are used to aid in project planning and execution, and the Highway Capacity Manual is one such tool whose goal is to provide methodologies and algorithms to assess the mobility effects of highway projects. In particular, the manual includes a macroscopic simulation approach for analyzing congested freeway facilities methodology rooted in the hydrodynamic theory of traffic flow. The methodology has been shown to be an effective approach for planning and operational analyses, but despite being widely used, little has been done to utilize optimization techniques in conjunction with the method.

This work begins by utilizing classical linear programming to provide a novel optimization framework for conducting congestion analysis using the methodology. As it is currently constructed in the manual, the computational procedure is inherently limited to computing outputs from a specified set of inputs. By reformulating these relationships and embedding them in an optimization framework, the methodology can be used to characterize vehicle flows based on any number of optimality criteria. The resulting approach bears many similarities to system optimal dynamic traffic assignment techniques, and its effectiveness is explored with respect to peak demand spreading and ramp metering applications.

Next, the work takes a step back and focuses on a key challenge of the model creation process. Before any planning or congestion management analysis can be conducted, an analyst must first create a calibrated model that faithfully represents the conditions of the real-world facility. This work develops a new calibration framework using a genetic algorithm metaheuristic to automate the demand estimation and adjustment process. Two encoding approaches for the problem are proposed based on the quality of available data. Additionally, the genetic algorithm approach is expanded to include three key capacity parameters, allowing the metaheuristic to consider pre-breakdown capacity adjustments, queue discharge flow rate, and the facility-wide jam density alongside demand volumes. At each stage, the effectiveness of the metaheuristic approach is demonstrated for examples representing both ideal conditions and real-world case studies. As an example, a real world case study is presented for the proposed calibration approach that focuses on a section of I-540 outside of Raleigh, NC. The calibration process is shown to remove 70% of the error for lightly congested conditions, and as much as 90% of the error when speeds indicate flow is highly congested. Further, the balance of the approach in matching both facility travel times and individual

segment speeds provides an improvement over current methods focusing on just the former.

Finally, a principle benefit of both the linear programming formulation and the GA calibration framework is their inherent flexibility. This flexibility provides an ideal basis for further innovation in terms of both optimization applications and improved model calibration. For example, numerous additional single and multi-objective functions can be studied for each approach, only a small number of which are presented in this work. Further, the LP model provides a strong optimization base that can be greatly expanded through the incorporation of integer variables, or even nonlinear constraints and objective functions. Additionally, the calibration framework can be improved and tailored to specific instances, utilizing more complex genetic operators or by implementing model-specific intergenerational learning heuristics.

© Copyright 2017 by Joseph Lake Trask

All Rights Reserved

Optimization Methods for Calibration and Analysis of Congested Freeway Facilities

by
Joseph Lake Trask

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Operations Research

Raleigh, North Carolina

2017

APPROVED BY:

Yahya Fathi

Earl Brill

John Baugh
Co-chair of Advisory Committee

Nagui Roupail
Co-chair of Advisory Committee

DEDICATION

To my family, whose support made this possible.

BIOGRAPHY

The author was born on March 4th, 1989 in Myrtle Beach, South Carolina. He attended Davidson College and earned a bachelor of science in mathematics in 2011.

ACKNOWLEDGEMENTS

I would like to thank the following people for their support:

- Dr. John Baugh for his guidance throughout my time at North Carolina State University, and for inspiring me to always give my best effort in all aspects of my work.
- Dr. Nagui Roupail for giving me the opportunity to work with ITRE, introducing me to the world of transportation engineering, and instilling in me a strong work ethic.
- Dr. Behzad Aghdashi for both his mentorship and constant willingness to help work through problems and discuss any aspects of my research.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
Chapter 1 Introduction	1
Chapter 2 A Linear Programming Formulation for Congested Freeway Facilities	4
2.1 Introduction and Motivation	4
2.2 Overview of the HCM Freeway Facilities Methodology	6
2.2.1 The Oversaturated Equations	7
2.3 Literature Review	8
2.3.1 Overview of Dynamic Traffic Assignment and the Cell Transmission Model ..	9
2.3.2 Optimization Extensions to the Core CTM Formulation	10
2.3.3 CTM Formulations and ATM Analysis	12
2.3.4 Overview the Transportation Network Design Problems (TNDP)	12
2.4 Formulation of Methodology as Dynamic Traffic Assignment	13
2.4.1 Flow Conservation	16
2.4.2 Mainline Flow	17
2.4.3 On-Ramp Flow	19
2.4.4 Off-Ramp Flow	21
2.4.5 Objective Function	22
2.5 Computational Results and Model Applications	23
2.5.1 System Optimal Results and Peak Spreading	24
2.5.2 Ramp Metering and Example	26
2.6 Conclusions and Future Work	30
Chapter 3 Calibrating Freeway Facilities with Genetic Algorithms and the HCM	32
3.1 Introduction and Motivation	32
3.2 Calibration through Demand Estimation	34
3.3 Literature Review	36
3.3.1 HCM Calibration and Hourly Demand Profiles	36
3.3.2 Calibration of the CTM and Related Models	38
3.3.3 System Identification and Metaheuristics	39
3.4 Genetic Algorithm System Identification Approach	41
3.5 Genetic Encoding of the Problem Space	42
3.5.1 Profile-Based Encoding	43
3.5.2 N-Distribution Encoding	44
3.6 Fitness Evaluation and Evolutionary Operators	50
3.6.1 Fitness Evaluation	50
3.6.2 Selection and Crossover	52
3.6.3 Mutation	53
3.6.4 Elitism and Preservation	53
3.7 Examples and Computational Results	54

3.7.1	Matching Hourly Demand Profiles	54
3.7.2	Simple Case Study	58
3.7.3	I-540 Case Study	67
3.8	Summary and Conclusions	76
3.9	Future Work	77
Chapter 4	A Two Stage Approach for Unified Demand and Capacity Calibration for Congested Freeway Facilities	79
4.1	Introduction	79
4.2	Literature Review	82
4.3	Capacity Calibration Parameters	83
4.4	Genetic Algorithm Overview	85
4.5	Unifying Demand and Capacity Calibration	87
4.6	Fitness Evaluation	88
4.7	Genetic Encoding of the Capacity Calibration Parameters	90
4.8	Selection and Crossover	93
4.9	Mutation	93
4.10	Computational Experiments and Numerical Results	94
4.11	Phase 1 Calibration Results	97
4.12	Phase 2 Calibration Results	102
4.13	Conclusions and Future Work	110
Chapter 5	Conclusions and Future Work	112
5.1	Summary and Conclusions	112
5.2	Future Work	115
	BIBLIOGRAPHY	117
	APPENDIX	123
	Appendix A Full Linear Programming Model	124
	A.1 Full Linear Programming Formulation	124
	A.2 Python and Gurobi Code	125

LIST OF TABLES

Table 2.1	Summary of key variables and parameters of the oversaturated methodology. .	15
Table 2.2	Summary of LP model size and solution for the first example facility.	25
Table 2.3	Ramp metering results for the I-40 facility with high congestion.	28
Table 3.1	Summary of the five parameters of each distribution representing the decision variables of the problem.	47
Table 3.2	Parameter ranges for the mixture distribution encoding.	55
Table 3.3	Results for each profile.	56
Table 3.4	Summary of the setup parameters for the profile-based encoding GA runs. . .	62
Table 3.5	Summary of the calibration results for the profile-based encoding GA runs. . .	62
Table 3.6	Summary of the setup parameters for the 3-distribution encoding GA runs. . .	64
Table 3.7	Summary of the demand estimation results for the 3-distribution encoding GA runs.	65
Table 3.8	Summary of the setup parameters for the profile-based encoding GA runs. . .	70
Table 3.9	Summary of the calibration results for the profile-based encoding GA runs. . .	71
Table 3.10	Breakdown of the mean absolute speed error for each segment in each time period for the three speed regimes (average of 5 GA runs).	73
Table 3.11	Summary of the setup parameters for the profile-based encoding GA runs. . .	74
Table 3.12	Summary of the setup parameters for the profile-based encoding GA runs. . .	74
Table 3.13	Breakdown of the mean absolute speed error for each segment in each time period for the three speed regimes (average of 5 GA runs).	76
Table 4.1	Recommended ranges for the capacity calibration parameters.	91
Table 4.2	Parameter minimum values and step sizes for the encoding example.	92
Table 4.3	Summary of the average absolute speed errors for the uncalibrated facility. . .	95
Table 4.4	Facility travel times based on the target speeds compared to those from the uncalibrated facility.	97
Table 4.5	Summary of the parameters used for the Phase 1 calibration.	98
Table 4.6	Summary results for the Phase 1 calibration run. All errors are the average absolute difference in speed.	99
Table 4.7	Facility travel times based on the target speeds compared to those from the facility after Phase 1 calibration.	101
Table 4.8	Summary of the parameters used for the Phase 2 calibration runs.	102
Table 4.9	Summary results for the Phase 1 calibration run. All errors are the average absolute difference in speed.	103
Table 4.10	Facility travel times based on the target speeds compared to those from facility after Phase 2 calibration.	105
Table 4.11	Summary results for the Phase 1 calibration run. All errors are the average absolute difference in speed.	108

LIST OF FIGURES

Figure 2.1	Diagram of the node-segment relationship. (Source: (<i>Highway Capacity Manual</i> 2016))	7
Figure 2.2	Example of the mainline flow (MF) and segment flow (SF) calculations with on-ramps and off-ramps.	8
Figure 2.3	Example fundamental flow-density diagram (<i>Highway Capacity Manual</i> 2016).	10
Figure 2.4	Dynamic Traffic Assignment representation of a network showing source and sink nodes added. Flow conservation can be preserved at these nodes.	14
Figure 2.5	24
Figure 2.6	Average absolute speed difference between solutions for each methodology for each segment along the facility over the course of the three hour study period	25
Figure 2.7	Geometry of the I-40 example facility.	27
Figure 2.8	Speed contours for the I-40 facility before and after ramp metering.	29
Figure 3.1	Example of a basic facility modeled in the HCM freeway facilities model (<i>Highway Capacity Manual</i> 2016).	33
Figure 3.2	Examples of well-known behavior exhibited in daily demand profiles (NCDOT, 2016).	36
Figure 3.3	Core freeway analysis calibration framework in the 6th edition of the HCM (<i>Highway Capacity Manual</i> 2016).	37
Figure 3.4	National default hourly demand profiles for weekdays and weekends on urban and rural interstates.	38
Figure 3.5	Flow chart detailing the process of a genetic algorithm.	42
Figure 3.6	Example of an input demand profile shape with a 15% search interval specified around it.	43
Figure 3.7	Visual example of the skew normal and skew Cauchy distributions. Both distributions are generated for a mean μ of 0 and a standard deviation σ of 1.	46
Figure 3.8	Example demonstrating the encoding of an n -distribution candidate profile into a binary string.	48
Figure 3.9	Full mixture distribution encoding example.	49
Figure 3.10	Visualization of the three crossover strategies.	52
Figure 3.11	Plotted results for matching the six unimodal subtypes. The true profile shape is represented by the red line, and the computed profile is shown as the blue dashed line.	57
Figure 3.12	Bimodal with PM peak known profiles (NCDOT, 2016). The red line represents the true profile, the blue dashed line shows the computed profile, and the green lines provide a 10% relaxation bandwidth of the profile.	58
Figure 3.13	Geometry of the simple example facility. The lane drop in segment 26 provides a bottleneck for the facility that creates two congested regimes.	59
Figure 3.14	Bimodal with AM peak 25th percentile demand profile (NCDOT, 2016).	59
Figure 3.15	Speed contour predicted by the HCM analysis for the contrived ground truth demand allocation.	60

Figure 3.16	Demand profile comparison and speed contour of the initial guess demand volume inputs.	61
Figure 3.17	Contour highlighting the speed differences between the GA run with the lowest total speed error and the target data set. White cells represent minimal speed errors, while red cells indicate larger speed errors.	63
Figure 3.18	Comparison of profiles generated by the best and worst calibration of the GA framework. The true profile is represented by the blue dotted line, and the initial incorrect profile is shown as an orange dotted line. The black line represents the profile as calibrated by the GA framework.	64
Figure 3.19	Side by side comparison of the target speeds (left) and the speeds predicted by the best GA run (right).	66
Figure 3.20	Comparison of profiles generated by the best and worst calibration of the GA framework. The true profile is represented by the blue dotted line and the black line represents the profile as estimated by the GA framework.	66
Figure 3.21	Geometry and target speed data for the I-540 WB case study.	68
Figure 3.22	Speed contour predicted by the HCM analysis for the initial demand assignment.	69
Figure 3.23	Predicted speed contour of the profile-based demand calibration run with the lowest total speed error.	72
Figure 3.24	Predicted speed contour of the demand estimation run with the lowest total speed error.	75
Figure 4.1	Core freeway analysis calibration framework in the 6th edition of the HCM (<i>Highway Capacity Manual 2016</i>).	80
Figure 4.2	Effects of different pre-breakdown bottleneck capacities (<i>Highway Capacity Manual 2016</i>).	84
Figure 4.3	Effects of two different capacity drop percentages α_1 and α_2 (<i>Highway Capacity Manual 2016</i>).	85
Figure 4.4	Effects of varying jam density values $K_{j,1}$ and $K_{j,2}$ on the volume density diagram (<i>Highway Capacity Manual 2016</i>).	86
Figure 4.5	High-level process flow of the unified two-stage calibration process.	89
Figure 4.6	Example of encoding three pre-breakdown CAFs and the facility-wide jam density into binary strings based on the values of Table 4.2.	92
Figure 4.7	Demonstration of three common crossover operators.	94
Figure 4.8	HCM segmentation of the 14.5 mile section of I-540 westbound near Raleigh, NC.	94
Figure 4.9	Speeds obtained from the real-world sensor data for Tuesday August 19th 2014 (top) and those initially predicted by the FREEVAL computational engine for the uncalibrated I-540 case study facility.	96
Figure 4.10	Comparison of the target speed contour (top) and the speed contour as predicted by the partially calibrated facility after Phase 1.	100
Figure 4.11	Comparison of target speed contour (top) and the speed contour as predicted by the calibrated facility after Phase 2.	104
Figure 4.12	Chart showing the set of pre-breakdown CAFs for all segments of the facility as estimated during the Phase 2 calibration process.	106

Figure 4.13 Comparison of target speed contour (top) and the speed contour as predicted by the methodology after the special case Phase 2 calibration with a single CAF109

CHAPTER

1

INTRODUCTION

The 2015 Urban Mobility Scorecard reports that drivers nationwide suffer through more than 7 billion hours of delay per year due to traffic congestion, costing the nation as a whole at least \$160 billion dollars and wasting 3 billion gallons of fuel.¹ When pairing this knowledge with the frustration of sitting in bumper to bumper traffic, few would argue that improving the nation's transportation systems is not a massively important endeavor. There are many avenues through which improvements can be made, but one of the most obvious areas centers on reducing congestion and delay on freeways. In a perfect world with an unlimited budget, all delay could be eliminated by simply increasing the capacity of freeways far beyond the expected traffic demands. However, in reality, funding for infrastructure projects is often very limited, and great care must be taken to ensure that the projects that are undertaken are well planned and carried out efficiently.

A variety of planning and analysis tools are used to aid in project planning and execution. The Highway Capacity Manual is one such tool whose goal is to provide methodologies and algorithms to "assess the traffic and environmental effects of highway projects."² In particular, the manual includes a *freeway facilities* methodology rooted in the hydrodynamic theory of traffic flow which has been shown to be an effective approach for planning and operational analysis. Despite being

¹Schrank,David, et al. "2015 Urban Mobility Scorecard." (2015). <https://mobility.tamu.edu/ums/>

²Highway Capacity Manual 6th Edition: A Guide for Multimodal Mobility Analysis. Transportation Research Board, Washington, D.C., 2016.

widely used, little has been done to take advantage optimization techniques in conjunction with the methodology. Incorporating optimization techniques can greatly expand the scope of the analysis and provide novel approaches for a number of new planning, operational, and congestion management applications. Additionally, in order for these types of analyses to be useful, they must be conducted on models that have first been calibrated to accurately represent observed real-world conditions. The model calibration process, which represents a major barrier to entry for use of the methodology, can also be vastly improved with optimization techniques by providing a new automated framework to replace the current arduous manual procedure. Consequently, the goal of the work presented is to explore improvements to the existing methodology through the use of both classical and metaheuristic optimization approaches. The work is presented as three chapters, with each constituting an individual paper.

Chapter 2 utilizes linear programming to provide a new optimization framework for conducting congestion analysis with the methodology. The framework is intended for use with calibrated models and works with known inputs and parameters to improve the performance of a specific aspect of the system, or the system performance as a whole. The new framework is created by reformulating the analytical *oversaturated* models as a set of linear constraints. These constraints can then be paired with a variety of objective functions that tailor the framework for use in congestion management approaches. In this context, the methodology emulates a dynamic traffic assignment model, and the system optimal approach can be used for a number of different applications. Two specific applications are demonstrated in the chapter. First, an analysis that computes an optimal spreading of peak demand is presented. This approach uses an objective function that minimizes the total delay of the system while maintaining the core node-flow relationships of the model. Next, a system-optimal ramp metering approach is presented that relaxes flow constraints loading the system at on-ramps in order to improve the overall performance.

The following two chapters take a step back and address the process of calibrating a core facility model. The calibration process is a foundational aspect of using the methodology and a critical step of model creation that precedes all conducted analyses. Model calibration is essential because any analysis that uses the method, including that as proposed above, hinges on the core model correctly representing daily recurring congestion conditions. In fact, modeling existing real-world conditions for a single representative day is the first step for using the methodology in conducting reliability and active traffic management analyses. This requirement necessitates that every model must be calibrated to match a set of target real-world conditions. The manual's current guidance concerning calibration is very limited, and the process often presents a major barrier to entry for the methodology.

Much of the challenge of the calibration process stems from the fact that each model requires a

large number of inputs, with even moderate sized models requiring the input and adjustment of thousands of individual parameters. Further, many of the critical inputs are difficult to measure under real-world conditions. Calibration of both input demand volumes and capacity parameters is vitally important as the driving force of the methodology is the relationship between demand volumes and segment throughput capacities. However, vehicle demand volumes are largely unknown under real world conditions and are essentially impossible to measure when traffic flow is congested. The manual does provide default estimates for the key capacity parameters, but these should ideally be adjusted in conjunction with demand volumes for each specific facility location.

A genetic algorithm metaheuristic framework is first developed in Chapter 3 to address the challenges of demand estimation in the context of model calibration. Two encoding approaches for the problem are proposed based on the quality of available data. One approach makes use of existing knowledge of hourly demand profiles, while the second utilizes mixture distributions of random variables in order to allocate demand over the course of a study period when the behavior is unknown. A genetic algorithm is then used to manipulate the individual demand inputs to find a set of overall volumes that minimizes the errors between predicted outputs and a set of target real-world performance measures (e.g., segment speeds). The effectiveness of the approach is tested on both an ideal facility with known demands and a single bottleneck, as well as a case study in Raleigh, NC, where the underlying demands are unknown.

Chapter 4 expands upon the demand estimation approach by adding three key capacity parameters into the existing calibration framework. New genetic encodings for the inputs are developed that allow the metaheuristic to consider pre-breakdown capacity adjustments, the queue discharge flow rate, and the facility-wide jam density alongside demand volumes. These three capacity parameters are important to incorporate as each controls aspects of breakdown occurrence, queue formation and recovery speeds, and overall queue size in a unique way. The resulting framework consists of a two-phase calibration process. During the first phase, the GA adjusts demand volumes and the queue discharge rate, while at the second phase the GA manipulates the jam density and bottleneck segment pre-breakdown capacity adjustments. Further, unlike the initial work of Chapter 3, this expanded approach utilizes an updated objective function that considers errors in both facility travel times and individual segment speeds. The effectiveness of the framework is once again tested on the North Carolina case study facility, and an analysis of the calibration resulting from the metaheuristic approach is presented.

CHAPTER

2

A LINEAR PROGRAMMING FORMULATION FOR CONGESTED FREEWAY FACILITIES

2.1 Introduction and Motivation

The Highway Capacity Manual (HCM) was first published in 1950 with the goal to provide methodologies and algorithms to “assess the traffic and environmental effects of highway projects” (*Highway Capacity Manual* 2016). Of the manual’s many approaches, this research focuses on the *freeway facilities* methodology, which was first introduced into the manual in 2000. The HCM freeway facilities methodology is a macroscopic simulation technique and computational procedure to calculate flows and speeds on an extended length of freeway. The driving force of the methodology is the relationship between three key quantities: the demand flow rate (demand input), segment capacities, and the actual flow rate (volume served). The analysis is divided into two operational regimes. A facility is considered to be *undersaturated* when the demand input never exceeds the available throughput capacity. However, when demand input exceeds capacity at at least one point along a facility, the freeway is considered to be *oversaturated*.

The oversaturated methodology is based on the Cell Transmission Model (CTM), which is rooted in the hydrodynamic theory of traffic flow (Daganzo, 1994). The oversaturated methodology expands on the core CTM methodology by introducing additional relationships to better capture congestion dynamics, as well as capabilities to track the horizontal propagation of vehicle queues. The associated computational procedure for the *oversaturated* methodology is presented as an iterative method that progresses in a “forward” manner across both the spatial and temporal dimensions of an analysis. While this formulation is adequate for a direct application of the methodology, its use is very limited for any additional analysis of the core flow relationships.

The limitations of the procedure stem in part from the nature of the relationship between demand, capacity, and volume served. For each flow relationship, volume served is specified to be the exact minimum of demand and any constraints on throughput capacity. The “exactness” here serves a number of important purposes, including helping the analysis to approximate a “first-in, first-out” (FIFO) condition, as well as a User Optimum (UO) system utilization property. However, using just the minimum of a set of values at each step effectively funnels out and discards information that may still be relevant for additional analysis. The discarded information is potentially valuable for reconstructing inputs from outputs, and can be instrumental for sensitivity analysis of facility performance under varying conditions.

The iterative nature of the formulation also comes with its own set of inherent limitations. Each facility analyzed in the methodology is broken down spatially into connected *segments*, and the group of segments is analyzed over the course of a study period consisting of consecutive 15-minute analysis periods. Within each analysis period, the procedure iterates downstream along the facility (i.e., in the direction of traffic flow), computing the congestion conditions at each segment. However, as currently constructed, at each computational step the procedure can only make use of information known from the current or previous steps. This restriction substantially limits the usefulness of the methodology as part of a framework for system level optimization approaches.

This paper presents formal mathematical programming techniques that provide a novel linear programming analysis framework for the HCM’s freeway facilities methodology. This new framework helps to address many of the limitations of the current computational procedure. Each of the key flow relationships of the methodology is examined and reformulated as a set of linear constraints. By defining the relationships in this manner, the methodology is opened up beyond just the basic direct application and analysis allowed by the iterative approach. The type of sensitivity analysis allowed by linear programming can provide new insight into the relationships between inputs, outputs, and intermediate flow quantities that can help to better understand and predict facility performance. Further, the framework allows for optimization approaches that can be used for facility operational and design analysis, as well as for the development of new traffic management techniques.

2.2 Overview of the HCM Freeway Facilities Methodology

The HCM defines a *freeway facility* as an extended length of freeway composed of continuously connected basic freeway, weaving, merge, and diverge segments (*Highway Capacity Manual* 2016). In the context of the methodology, a facility is generally a 9-12 mile portion of a single freeway as opposed to an interconnected network of freeways. For a typical freeway facilities analysis, a study period falling within a 24-hour time period is specified and divided into 15 minute analysis periods.

In order to conduct traffic flow analysis, the HCM provides four core methodologies that can be used to analyze each type of segment on a individual basis. While these methodologies can be used for some operational and design analyses, they are inherently limited in scope as they cannot capture conditions that extend beyond the boundaries of the uniform segments. This is problematic as most analysis of interest involves traffic congestion and the formation of vehicle queues that can extend far beyond the allowable length of a single segment type. In order to accurately capture these types of conditions, the freeway facilities methodology was developed to unite and extend the four core segment methodologies (*Highway Capacity Manual* 2016).

For both the freeway facility and the segment methodologies, the analysis revolves around the relationship between the *demand flow rate* and the *actual flow rate*. The demand flow rate, or input demand, is defined to be the number of vehicles that wish to traverse the facility or segment during a given time period. The actual flow rate, or volume served, is the number of vehicles that are actually able to move through the segment after accounting for maximum link capacities and related congestion conditions. In the context of these two quantities, the methodologies treat the flow through each segment as the minimum of the desired input and the available output.

When the input demand is less than the available capacity, the analysis is straightforward. The underlying hydrodynamic theory allows flow-density diagrams and speed-flow equations to be used to estimate segment and facility performance measures. Additionally, because all demand is being met in this case, the segments can be analyzed individually as their performance conditions are effectively independent. However, at times where the demand flow rate is greater than the allowable actual flow rate, the analysis can change drastically. In the event that demand exceeds capacity, and a *bottleneck* and resulting flow breakdown occur, the individual segment methodologies can no longer provide accurate results. The effects of unserved demand can extend upstream to additional segments, and the effects of restricted demand rates may be felt on flows downstream of the bottleneck.

Bottleneck and breakdown analysis are precisely the scenarios that users of the methodology need to analyze. In fact, this is a primary motivation for the HCM and other related methodologies. A principal difficulty for most approaches comes in ensuring that the dynamics of the congestion

can be fully captured in the spatial and temporal domain of the system. Some methods resort to tracking queues “vertically” on a single segment, but this solution is far from desirable when dealing with situations involving large queues that span multiple segments. The HCM’s freeway facilities methodology provides a solution to this issue by allowing a facility to be analyzed as a whole. This increased spatial extent can be used to accurately model the accumulation and upstream propagation of queues. Allowing queues to be tracked “horizontally” reflects real world behavior. Further, when the extent of the spatial domain is chosen well, even the effects of a bottleneck on downstream flow can also be modeled.

2.2.1 The Oversaturated Equations

The freeway facilities methodology considers a facility at all times to be in one of two states. Due to the existence of these two conditions, the freeway facilities methodology relies on two connected sets of equations for its analysis. The first is the undersaturated method, which utilizes the individual segment methodologies to analyze a facility. The second algorithm is the oversaturated method, which is invoked as soon as demand exceeds capacity in at least one segment along the facility.

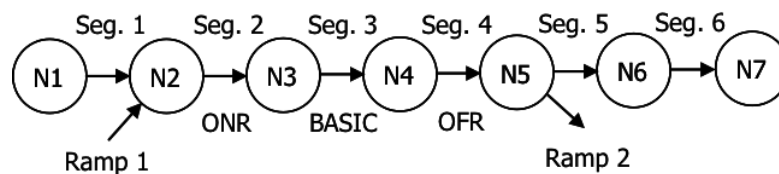


Figure 2.1 Diagram of the node-segment relationship. (Source: (*Highway Capacity Manual* 2016))

The oversaturated method introduces more complex flow and queuing equations in an effort to capture the congestion dynamics that may arise. The method also instantaneously increases the computational resolution by breaking each 15 minute analysis period into sixty 15 second time steps and slightly adjusting the spatial domain. Instead of just analyzing the conditions at each segment, the facility is viewed as a set of nodes and segments as is shown in Figure 2.1. Flows are computed at each node, and segments are used to track stored vehicles and flow densities. At its simplest, the flow through each node is the minimum of input demand and segment capacity. The presence of congestion and queuing can introduce new limits on the output available at a node. These new restrictions can arise from turbulence due to cars entering a congested mainline from an on-ramp, as well as due to the build up of queues during previous time steps. The oversaturated methodology accounts for these situations by introducing new quantities into the flow relationships

defined at each node type. These new relationships are discussed in Section 2.4.

The oversaturated equations evaluate the facility by computing flows for each node, starting at the first upstream node and moving down the facility. Traffic demand enters the facility at the upstream *mainline* segment, and flows down the facility until it exits either at a *diverge* (off-ramp) segment or at the end of the facility. In addition to the demand entering the facility at the upstream segment, demand can enter at *merge* (on-ramp) segments provided that the segment has enough capacity to contain both flows. Flows at each node consist of three quantities, mainline flow (MF), on-ramp flow (ONRF), and off-ramp flow (OFRF). Segment flows (SF) are calculated for each intermediate segment between every pair of nodes. Figure 2.2 shows how mainline and segment flows interact at merge and diverge segments.

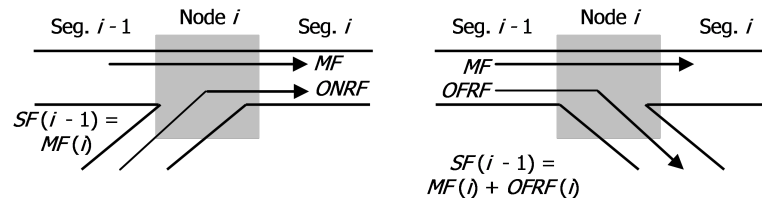


Figure 2.2 Example of the mainline flow (MF) and segment flow (SF) calculations with on-ramps and off-ramps.

For this research, the assumption is made that computation for an entire study period can just be confined to the set of oversaturated equations. This assumption is made for two reasons. First, it is possible to analyze undersaturated conditions using the oversaturated equations with a similar degree of accuracy. Second, analysis periods in which a facility is undersaturated have limited usefulness. As such, this research can assume study periods are almost completely restricted to times where the facility is oversaturated. All that is required is that the study period include a small number of undersaturated “warmup” and “cooldown” periods at the beginning and the end of the analysis. These ensure that the congestion and queues have time to build correctly and fully dissipate, and can generally consist of a single undersaturated period at both the beginning and end of a study period.

2.3 Literature Review

The following sections present a review of the literature as it relates to this work that has been published on Dynamic Traffic Assignment (DTA), Transportation Network Design Problems (TNDP),

and extensions and updates to the Cell Transmission Model (CTM). Most of the work presented in the following sections is built either on top of the CTM or a simplified set of flow and traffic assignment equations.

2.3.1 Overview of Dynamic Traffic Assignment and the Cell Transmission Model

An important analysis technique used in planning and evaluating freeway transportation networks is Dynamic Traffic Assignment (DTA). DTA seeks to model “time-varying network and demand interaction using a behaviorally sound approach” (Chiu et al., 2011). This research is primarily interested in a subset of DTA consisting of equilibrium-seeking *mesoscopic* models, i.e., models whose resolution falls between microscopic and macroscopic models, but make use of properties of both.

DTA models are often used to evaluate travel times and costs of network performance by modeling them as optimization problems. In these formulations, travel time and cost become the objectives to be minimized and are subject to flow conservation and traffic assignment constraints. Throughout the literature reviewed, two different traffic assignment conditions were generally used for the DTA formulations. The first and simpler of the two conditions is the System Optimum (SO) condition. Under SO assignment, the goal is to minimize the travel time or cost of the system as a whole (Ziliaskopoulos, 2000). The second condition, User Equilibrium (UE) or User Optimum (UO) assignment, differs from this in that each individual driver seeks to minimize his/her travel time, even if it comes at the cost of total system performance (Ukkusuri & Waller, 2008). This dynamic can be more difficult to capture in mathematical models, but is generally thought of to be a more realistic representation. It should be noted that under free flow conditions with little to no congestion, the two conditions will converge to the same answer, but when congestion is high, the optimal solutions can differ greatly. An in depth survey on DTA is given by Chiu et al. (2011).

One of the more widely used DTA models is the Cell Transmission Model (CTM), which was first proposed by Carlos Daganzo (1994). The CTM primarily consists of a set of difference equations that simulate traffic flow on highways and use shockwaves to model the backward (upstream) propagation of traffic congestion. The equations are presented as a discrete approximation of the differential equations proposed by Lighthill, Witham, and Richards (Lighthill & Whitham, 1955a,b; Richards, 1956). These original differential equations (often referred to as the LWR equations) were developed by applying hydrodynamic flow theory to traffic flow and analytically account for important concepts such as the Flow-Density diagram (Figure 2.3) and the shockwave propagation of congestion. In his initial paper, Daganzo asserts that the CTM is the discrete equivalent of the underlying hydrodynamic theory. The equations that make up the CTM allow for four distinct

inputs: the free flow speed (FFS) of the highway, the maximum flow in a cell (segment capacity), the jam density, and the wave speed of the propagation shockwaves. Work was later done by Daganzo (1995) to expand the model to represent traffic networks by allowing for basic, merge (on-ramp), and diverge (off-ramp) segments to be represented in the equations. Due to its accuracy (as shown in (Lin & Ahanotu, 1995; Smilowitz & Daganzo, 1999)) and relative computational simplicity, vast amounts of research have been conducted to expand the core methodology of the CTM.

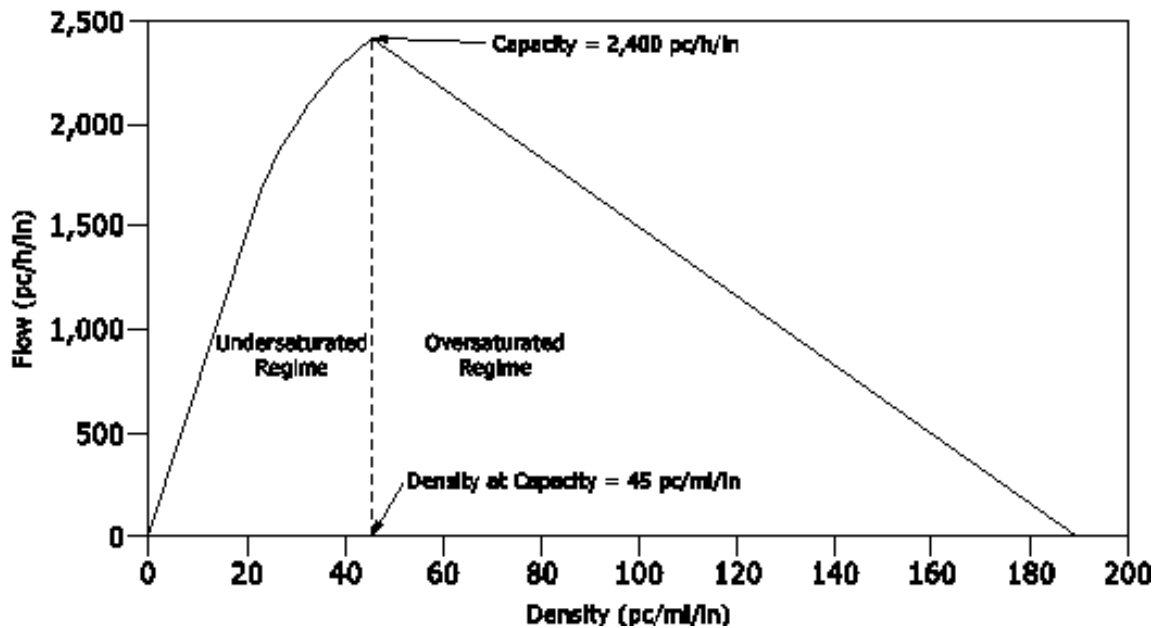


Figure 2.3 Example fundamental flow-density diagram (*Highway Capacity Manual* 2016).

2.3.2 Optimization Extensions to the Core CTM Formulation

The first work to utilize the CTM in a pure linear programming framework was presented in the context of the System Optimal Dynamic Traffic Assignment (SO-DTA) problem (Ziliaskopoulos, 2000). Due to the relative computational tractability of linear programming, a large amount of research has since been devoted to building on the work. Waller & Ziliaskopoulos (2001) extended the model into a two stage LP in order to account for stochastic time-dependent demands. Li et al. (2003) developed a Dantzig-Wolfe decomposition scheme for the formulation. Work has also been done to incorporate UE (or UO) route assignment as opposed to the simpler SO assignment.

Travis Waller & Ziliaskopoulos (2006) first proposed a combinatorial optimization approach (as opposed to mathematical programming), while later Ukkusuri & Waller (2008) directly extended the CTM linear programming formulation to account for UE conditions. In their work, they also compared the effect of the two route assignment conditions (SO and UE) on solutions for sample TNDP problems. Han et al. (2011) further built on the formulation making use of complementarity theory to improve the calculation of travel time estimates and capture multiple user classes with elastic demands. Active Traffic Management (ATM) techniques such as dynamic congestion pricing for tolls have also been proposed based on sensitivity and dual variable analysis allowed by the linear programming formulations (Lin et al., 2011).

In order to model the CTM as a pure linear program, Ziliaskopoulos (2000) used the simplifying assumption that the minimum relationship between flow and capacity could be represented as a set of "less-than" inequality constraints on link flow. However, this assumption does not guarantee that the assigned quantity is truly the minimum of the values, i.e., $x \leq y, x \leq z$ does not imply $x = \min\{y, z\}$. This can lead to an issue known as the *vehicle-holding* or *holding-back* problem. An overview of the *holding-back* problem as well as a survey of the work done to account for it was compiled by Doan & Ukkusuri (2012). Lo (2001) proposed accounting for this nonlinearity by incorporating integer linear constraints that effectively turn sets of constraints on or off, though the work was mostly investigated for its use in signal control applications. In a similar vein, Tampère & Immers (2007) proposed the use of the Heaviside function for the approach. Zheng & Chiu (2011) addressed the issue by proving that the single destination system optimal dynamic traffic assignment (SD SO-DTA) problem is equivalent to the earliest arrival flow (EAF) network problem, and developed a network flow algorithm that eliminated vehicle holding in basic and merge segments (but not necessarily for diverge segments).

Doan & Ukkusuri (2012) presented multiple formulations to completely remove vehicle-holding under SO routing for problems with multiple destinations. The removal of vehicle-holding was achieved by using complementarity constraints and converting them to a nonlinear programming model, as well as by incorporating mixed integer programming (MIP) techniques. Next, Ukkusuri et al. (2012) extended the techniques to handle the UE conditions. Unfortunately, each of these formulations requires assumptions that may not accurately reflect realistic conditions, and in addition still contain some nonlinearities or non-convexities (Zhu & Ukkusuri, 2013). Further, in all of the previously mentioned papers the authors acknowledge that the formulations are still impractical for larger and more realistic sizes of networks.

Another linear complementarity approach was developed by Zhong et al. (2013) in order to better deal with the nonlinearity of the state jump conditions and hard nonlinear *min* functions. The proposed formulation readily extends to both DTA and TNDP analysis, and while not as com-

putationally efficient as pure LP formulations, it benefits from the existing work done on linear complementarity problems (LCP) (Hu et al., 2012). More recently, Zhu & Ukkusuri (2013) were able to develop a pure linear programming solution to the vehicle-holding problem for the SD SO-DTA problem by introducing an easily computable penalty function into the formulation. Sun et al. (2014) extended this to incorporate demand uncertainty and developed an adjustable robust optimization (ARO) linear programming formulation.

2.3.3 CTM Formulations and ATM Analysis

The CTM formulation and its extensions have also been used to tune ATM strategies. Gomes & Horowitz (2006) developed the Asymmetric Cell Transmission Model (ACTM) to compute optimal ramp metering rates. While the ACTM formulation remains nonlinear, the authors proved sufficient conditions under which the globally optimal solution could be found through the use of a single linear program. Unfortunately, these sufficiency conditions were rather constrictive, and often highly unrealistic of true freeway conditions. Finding a lack of in-depth theoretical results concerning the CTM/MCTM, Gomes et al. (2008) further investigated the properties of congested and uncongested equilibria for the model and interpreted the findings with respect to the effectiveness of ramp metering strategies. The authors found that optimally computed ramp metering strategies can reduce congestion and travel time on freeways, and that the strategies do more than just move the congestion to on-ramps. A dynamic on-line method to determine Variable Speed Limits (VSL) has been proposed by Li et al. (2014), with benefits stemming from the low computational cost of the CTM vs the more complex existing VSL algorithms.

2.3.4 Overview the Transportation Network Design Problems (TNDP)

Beyond supporting sensitivity analysis, optimization frameworks can also provide a direct path to finding ways of improving existing transportation networks. Most problems of this nature fall under the umbrella of Transportation Network Design Problems (TNDP). These problems seek to provide an optimization model for the selection of network improvements. For example, if funds for road widening are available, a TNDP formulation could be used to select the optimal portion to be widened. There are a number of appropriate ways to formulate a problem such as this, but often the model would seek to maximize the post-improvement traffic flow, while simultaneously minimizing construction costs, all while satisfying the system's flow conservation and traffic assignment constraints. DTA and TNDP have many important applications beyond just their initial scope. They can be used to improve reliability analysis of transportation networks, especially under abnormal conditions such as evacuations (Lim et al., 2015; Malveo, 2013; Yao et al.,

2009).

Due to the interplay between the equilibrium constraints and the multiple objective functions, single level TNDP formulations are generally nonconvex and nonlinear. These types of problems are known to be at least NP hard (Zeng & Mouskos, 1997). To regain some linearity, TNDP is often formulated as a *bilevel program*, i.e., one with an upper level and a lower level objective function. Despite the inherent difficulty in finding a solution for formulations such as these, it is still important that at least a near globally optimum solution be found. As such, heuristic algorithms have often been found to be the best choice to efficiently seek an optimal or near-optimal solution. In recent years, much research has been conducted into the use of metaheuristic techniques such as genetic algorithms (GA), simulated annealing (SA), and tabu search for extending and solving TNDP formulations (Delbem et al., 2012; Juan et al., 2012; Karoonsoontawong & Waller, 2006; Luathep et al., 2011; Mudchanatongsuk et al., 2008; Xu et al., 2009; Zeng & Mouskos, 1997).

2.4 Formulation of Methodology as Dynamic Traffic Assignment

With the exception of the minimum and maximum functions, the majority of the key flow and associated relationships of the HCM's oversaturated model are linear. Thus, so long as those functions can be approximated linearly, this allows the methodology to be incorporated into classical linear optimization approaches. A minimum or maximum of a set of values can easily be approximated through a group of linear inequalities, which can then serve as constraints in a linear optimization model. In general, this falls under the umbrella of linear programming, which consists of a linear objective function that is minimized (or maximized) subject to a set of linear inequality and equality constraints. For the purposes of this work, any number of objective functions can be chosen, and it will largely serve as a way to tailor the model for the various applications. This is because the crux of the model is truly found in the constraints. The relationships laid out therein define and govern the flows along the facility such that they are consistent with the oversaturated methodology. In this way, the formulation can be thought of as a dynamic traffic assignment (DTA) model.

Further, through the addition of two "dummy" nodes to the model, a freeway facility can be thought of as being a single origin - single destination network. These two added nodes serve as source and sink nodes for the demand desiring to use traverse the network. In this way, a freeway facility can be thought of as having a single origin and a single destination. However these dummy nodes are never explicitly used in any computations. Figure 2.4 shows a facility with one on-ramp, one weave, and one off-ramp and the corresponding single origin - single destination DTA network representation.

As stated earlier, DTA models generally fall into two types: system optimal (SO) or user optimal

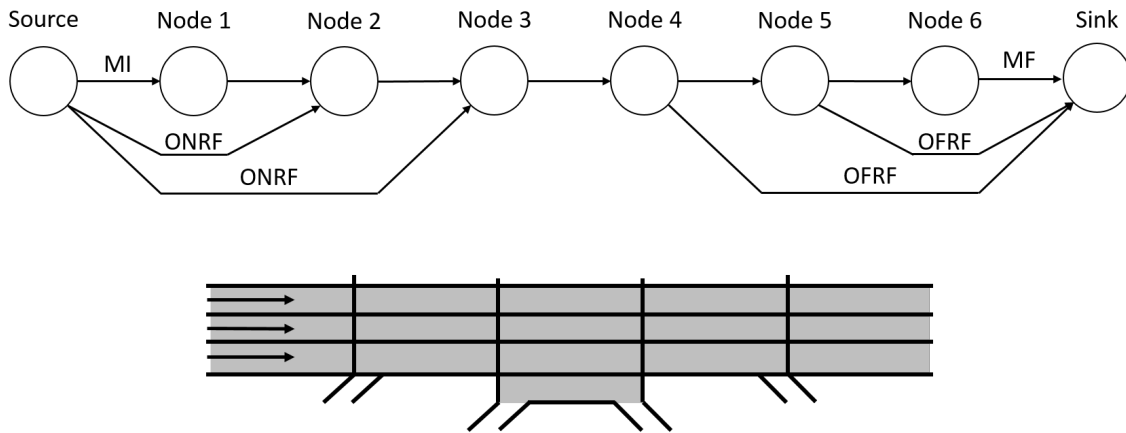


Figure 2.4 Dynamic Traffic Assignment representation of a network showing source and sink nodes added. Flow conservation can be preserved at these nodes.

(UO) (Ukkusuri & Waller, 2008). Treating the minimum functions as a group of “less-than” constraints inherently leads to the SO condition as the exact flows may not be forced through each node when it is detrimental to the system performance as it defined in the objective function. The primary challenge of modeling the freeway facilities method of the HCM in an optimization framework comes from exactly this. Depending on the choice of objective function, simply using inequalities can lead to the phenomenon of *holding-back* flows. Holding-back flows are a commonly arising issue in macroscopic dynamic traffic assignment, and the resulting model will generally be considered to be a system optimal model. While this type of model is not necessarily as realistic as a user optimal model, it still provides many useful insights for both planning and operational purposes.

It is important to note that the model proposed in the following sections differs from the HCM’s computational procedure in a number of ways. While both operate on a known set of demand and capacity inputs to compute the resulting flows and performance of the system, the proposed formulation approaches the methodology in a fundamentally different way than the current framework in order to provide new avenues for analysis of the flow relationships. Specifically, the computational procedure seeks to build a solution by analyzing facility conditions in an iterative manner. On the other hand, the use of linear programming allows us to define the desired properties of a solution through a set of constraints. Solving the LP then allows us to seek solutions with those properties that are optimal in regards to a specified set of metrics.

While this model removes some of the biggest limitations of the HCM’s computational procedure, it does come with its own set of limitations. Macroscopic models generally struggle to maintain a “first-in, first-out” (FIFO) property, and both the original and this reformulation have issues

preserving this condition under high levels of congestion — an issue discussed in more detail in Section 2.4.4. This model also is limited in its ability to enforce a UO condition. As discussed in the literature review, work has been done to implement the UO condition for similar models, and much of that work can be directly applied to the model. However, for the purposes of this paper, the model is never assumed to enforce the UO condition.

Table 2.1 Summary of key variables and parameters of the oversaturated methodology.

Symbol	Definition
Indices	
N_S	Set of indices indicating the node, $i = 1, \dots, N_S $. The number of nodes is one greater than the number of segments. Segment i refers to the segment immediately downstream from node i .
S	Set of indices indicating the time step of the current period, $t = 1, \dots, S $.
P	Set of indices indicating the current analysis period, $p = 1, \dots, P $.
\tilde{N}	Index set of the on-ramp segments of a facility, $\tilde{N} \subset N_S$
\tilde{F}	Index set of the off-ramp segments of a facility, $\tilde{F} \subset N_S$
Segment Variables	
$NV_{i,t,p}$	Number of vehicles on a segment i , in step t , for period p .
$UV_{i,t,p}$	Unserved vehicles: the additional number of vehicles stored in Segment i at the end of time step t in interval p due to a downstream bottleneck.
$SF_{i,t,p}$	Segment flow out of segment i during step t in interval p .
$SC_{i,p}$	Capacity of segment i in analysis period p .
$KB_{i,p}$	Background density of segment i during analysis period p .
KJ	Jam density - the maximum per lane density allowed on a segment.
KC	Ideal Density - the per lane density of a segment operating at capacity
L_i	Length of segment i .
N_i^L	Number of lanes in segment i .
Node Variables	
$MI_{i,t,p}$	Desired mainline input flow into node i during time step t in period p .
$MF_{i,t,p}$	Actual mainline flow rate leaving node i during time step t in period p .
$ONRF_{i,t,p}$	Actual ramp flow rate that can cross on-ramp node i during step t in period p .
$ONRQ_{i,t,p}$	Queue due to unmet on-ramp demand at node i during step t in period p .
$OFRF_{i,t,p}$	Actual flow that can exit at off-ramp node i during time step t in period p .

There are three principal segment variables that are used in order to track and assess the flow of vehicles on a facility. The first is the number of vehicles $NV_{i,t,p}$ on particular segment i in time step t , in time period p . The second is the number of unserved vehicles $UV_{i,t,p}$ on particular segment

i in time step t , in time period p . Lastly, the variable $SF_{i,t,p}$ is defined to be the segment flow for a segment i , in time step t for time period p . Almost all traditional performance measures for the analysis can be calculated using some combination of these three values in combination with the underlying facility characteristics. Additionally, there exist three node flow variables that track the mainline flow, on-ramp flow, and off-ramp flow. These are denoted $MF_{i,t,p}$, $ONRF_{i,t,p}$, and $OFRF_{i,t,p}$, respectively. Table 2.1 provides a summary of the initial model variables.

2.4.1 Flow Conservation

Since the underlying principles of the methodology are based on hydrodynamic theory, one of the first relationships that should be considered is one that ensures a “conservation of flow” (Daganzo, 1994, 1995). The property is implicitly present in the iterative procedure, but requires an explicit constraint in the linear programming framework. Equation 2.1 achieves this and states that all demand that enters the facility at the first upstream segment or at an on-ramp, must also exit the facility at the final downstream mainline segment or an off-ramp. Consequently, there is an implicit assumption that the length of the study period chosen for the analysis allows the flows to fully discharge, but equation 2.1 can be relaxed to an inequality to allow queues to persist through the end of the study period.

$$\sum_{t=1}^{|S|} \sum_{p=1}^{|P|} MF_{0,t,p} + \sum_{i \in \bar{N}} ONRF_{i,t,p} = \sum_{t=1}^{|S|} \sum_{p=1}^{|P|} MF_{NS,t,p} + \sum_{i \in \bar{F}} OFRF_{i,t,p} \quad (2.1)$$

There are four equations in the HCM related to flow conservation that can be directly incorporated into the linear programming model. Equation 2.2 initializes each segment at the beginning of each analysis period by defining the minimum number of vehicles that will be on a segment at any point in time based on the underlying demand. Equation 2.3 states that the number of vehicles on a segment is equal to the number of vehicles on the segment at the previous time step, plus the input from the upstream node (i), and minus the output at the downstream node ($i + 1$). A new constraint is added here to ensure that the number of vehicles on a segment at any step never exceeds the maximum allowed by the pre-specified jam density KJ . Equation 2.5 serves to track the growth and dissipation of any vehicle queues that may form in the analysis. The constraint defines the number of unserved vehicles to be any vehicles on segment exceeding the normal operating background density, a value computed from demand inputs. Lastly, equation 2.6 defines the segment flow, which can be used to compute the speed at a segment.

Each of the flow variables in equations 2.3 – 2.6 represents the *actual flow* of that type at the node. These values must correctly model the relationship between desired input and available output. The

following sections will present each flow sub-problem in a declarative manner that approximates the relationships as they are defined in the HCM.

$$NV_{i,0,p} = KB_{i,p}L_i + UV_{i,S,p-1} \quad \forall i \in N_S, p \in P \quad (2.2)$$

$$NV_{i,t,p} = NV_{i,t-1,p} + MF_{i,t,p} + ONRF_{i,t,p} - MF_{i+1,t,p} - OFRF_{i+1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.3)$$

$$NV_{i,t,p} \leq N_i^L L_i KJ \quad \forall i \in N_S, t \in S, p \in P \quad (2.4)$$

$$UV_{i,t,p} = MI_{i+1,t,p} - MF_{i+1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.5)$$

$$SF_{i,t,p} = MF_{i+1,t,p} + OFRF_{i+1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.6)$$

2.4.2 Mainline Flow

The formulation of the mainline flow relationship presents the biggest differences between the HCM oversaturated methodology and the CTM. Consequently, it is also the main area in which this model deviates from the SO-DTA formulation proposed by Zilliaskopolous. In addition to a segment's capacity, the HCM introduces three new limits on mainline flow output in order to capture additional congestion dynamics. These three values are referred to as Mainline Outputs 1, 2, and 3, respectively. The following set of linear equalities and inequalities approximates the mainline flow relationship as it is defined in the HCM.

$$MI_{i,t,p} = MF_{i-1,t,p} + ONRF_{i-1,t,p} - OFRF_{i,t,p} + UV_{i,1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.7)$$

$$MF_{i,t,p} \leq MI_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.8)$$

$$MF_{i,t,p} \leq SC_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.9)$$

$$MF_{i,t,p} \leq SC_{i-1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.10)$$

$$MF_{i,t,p} \leq MO1_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.11)$$

$$MF_{i,t,p} \leq MO2_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.12)$$

$$MF_{i,t,p} \leq MO3_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (2.13)$$

Equation 2.7 defines the desired mainline input into the node, and equations 2.8 – 2.13 limit the flow to be less than the desired input at the node and available throughput of the node. Each of the three mainline outputs serves to limit throughput in a unique way. The first, denoted *MO1*, accounts for limits on node output based on turbulence due to merging on-ramp flows. This relationship is defined as the minimum of three values in the HCM, and can be approximated by the following

three inequalities.

$$MO1_{i,t,p} \leq SC_{i,t,p} - ONRF_{i,t,p} \quad (2.14)$$

$$MO1_{i,t,p} \leq MO2_{i,t-1,p} \quad (2.15)$$

$$MO1_{i,t,p} \leq MO3_{i,t-1,p} \quad (2.16)$$

Mainline Output 2 (*MO2*) serves to limit node output due to queue accumulation from a downstream bottleneck. In the HCM methodology, this value is computed by first estimating the maximum density KQ allowed on a segment based on the size of its queue, and limiting the flow entering the segment from the node to be no greater than that which would push the segment to the maximum allowed density. However, because the HCM procedure is iterative, queue size for the segment at the current step is unknown at that point in the computations, so KQ must be estimated from conditions the previous time step. This estimation is made by computing the maximum density allowed by the flow rate, utilizing the jam capacity KJ , the ideal capacity KC , and the decreasing linear relationship of the flow density diagram that exists when density is greater than capacity (see Figure 2.1). Equations 2.17 and 2.18 show these relationships.

$$KQ_{i,t,p} = KJ - \frac{KJ - KC}{SC_{i,p}} * SF_{i,t-1,p} \quad (2.17)$$

$$MO2_{i,t,p} = SF_{i,t-1,p} - ONRF_{i,t,p} + KQ_{i,t,p} * L_i - NV_{i,t-1,p} \quad (2.18)$$

Since our LP model lacks this restriction of known information, it is possible to reformulate this relationship through an alternative, and arguably more accurate, set of equations. First, it must be noted that a key assumption for the HCM procedure for calculating KQ is that a queue exists on the downstream segment, which indicates that flow is on the right side of the flow density relationship. Otherwise, the same value for flow can correspond to two different densities. Hence a check on the size of UV at the downstream segment is required, which necessitates the use of integer variables to implement correctly. To avoid this, we can invert the relationship, and the maximum allowable segment flow for a given segment density can be used to the same effect. Further, since *MO2* should only limit flow in the presence of a queue, we can simply extend the linear relationship of the right side of the flow-density diagram to compute the maximum allowable segment flow at any density. Once the maximum allowable segment flow has been determined, *MO2* is simply that value minus any competing on-ramp flow that enters at the node. These new relationships are given in equations

2.19 and 2.20.

$$ASF_{i,t,p} = \left([KJ * N_i^L] - \frac{NV_{i,t,p} + NV_{i,t-1,p}}{2 * L_i} \right) * \frac{SC_{i,p}}{N_i^L * (KJ - KC)} \quad (2.19)$$

$$MO2 = ASF_{i,t,p} - ONRF_{i,t,p} \quad (2.20)$$

The final limit on mainline output ($MO3$) models the effects of queue dissipation of front-clearing queues. A queue is considered to be "front-clearing" when the queue density decreases while the rear position of the queue remains unchanged. Modeling this behavior requires a linear estimation of the recovery wave speed¹ $ws_{i,p}$ based on capacity as well as on known ideal and jam densities. The recovery speed is then translated to the wave travel time² $wtt_{i,p}$ based on the length of the segment being analyzed. A more detailed explanation of front-clearing queues can be found in the *Highway Capacity Manual* (2016). The $MO3$ relationship is presented as a single minimum statement in the HCM methodology and can thus be directly approximated using a set of linear inequalities as shown in equations 2.21 - 2.25.

$$MO3_{i,t,p} \leq MO1_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad (2.21)$$

$$MO3_{i,t,p} \leq MO2_{i+1,t-wtt,p} + OFRF_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad (2.22)$$

$$MO3_{i,t,p} \leq MO3_{i+1,t-wtt,p} + OFRF_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad (2.23)$$

$$MO3_{i,t,p} \leq SC_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad (2.24)$$

$$MO3_{i,t,p} \leq SC_{i+1,t-wtt,p} + OFRF_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad (2.25)$$

2.4.3 On-Ramp Flow

For each 15 minute analysis period, there exists demand at each on-ramp node defined as the known value $ONRD_{i,p}$. The on-ramp flow at a node i for time step t in period p is defined by the following

¹ $ws_{i,p} = \frac{SC_{i,p}}{N_i^L * (KJ - KC)}$, where KJ is the jam density, and KC is the ideal density.

² $wtt_{i,p} = \frac{T_h * L_i}{ws_{i,p}}$, where T_h is the number of time steps (t) in one hour.

relationships:

$$\sum_{t=1}^S \sum_{p=1}^P ONRF_{i,t,p} = \sum_{p=1}^P ONRD_{i,p} \quad \forall i \in \tilde{N} \quad (2.26)$$

$$ONRQ_{i,t,p} = \sum_{x=1}^{p-1} \left[ONRD_{i,x} - \sum_{\tau=1}^S ONRF_{i,\tau,x} \right] + \sum_{\tau=1}^t \left[\frac{ONRD_{i,p}}{|S|} - ONRF_{i,\tau,p} \right] \quad \forall i \in \tilde{N}, t \in N_S, p \in P \quad (2.27)$$

$$ONRF_{i,t,p} \leq ONRD_{i,t,p} + ONRQ_{i,t-1,p} \quad (2.28)$$

$$ONRF_{i,t,p} \leq RM_{i,t,p}, \quad (2.29)$$

$$ONRF_{i,t,p} \leq ONRC_i \quad (2.30)$$

The first equation provides a conservation of demand by requiring that for each ramp the total on-ramp flow is equal to the sum of the on-ramp demand for the study period. For complete model accuracy, this assumption requires that the study period is long enough such that all on-ramp demand can be served. However, this condition can be relaxed to a less-than inequality if necessary, which will allow the study period to conclude with vehicles still in the on-ramp queue. Equation 2.27 updates the on-ramp queue by tracking the deficit in on-ramp in comparison to the underlying demand.

Equation 2.28 requires that the flow is less than the desired input of the on-ramp. Equations 2.29 – 2.30 constrain the flow based on the available output of the ramp. The relation accounts for a ramp metering rate (RM) and the capacity of the roadway itself (*ONRC*). The on-ramp flow is also constrained by some of the relationships presented previously. Equations 2.3 and 2.4 of Section 2.4.1 place limits on on-ramp flow from the maximum allowable density of a segment. Additionally, each of the limits on mainline output discussed in the previous section ensures that the combination of on-ramp flow and mainline flow does not exceed the effective flow capacity of a node.

The HCM methodology limits the allowable on-ramp output in one additional way. The computational procedure uses a maximum relationship embedded within a minimum relationship to both limit flow when queues exist in an on-ramp segment, while also ensuring that the available output is always at least as large as one-half of the capacity of a single lane. This allowance serves to approximate a one-to-one zipper merge that may occur when a segment is fully queued but still has high on-ramp demand. Unfortunately, enforcing this condition exactly is a non-convex relationship and is thus out of the scope of this linear programming formulation. Further, this aspect of the relationship constitutes a user optimal condition as opposed to a system optimal condition, and can conflict with applications of the model where system optimality is the goal (e.g., ramp metering).

2.4.4 Off-Ramp Flow

Modeling off-ramp flow provides a significant challenge for most macroscopic simulation models. While the relationship is simple to maintain when there is no congestion, the flow relationship can become very complicated in the presence of large queues. The issues arise because macroscopic flow models do not track individual vehicles, and makes enforcing a “first-in, first-out” (FIFO) condition is extremely difficult without the use of integer variables (Carey & Watling, 2012; Carey et al., 2014a,b).

To illustrate this issue, consider the case where a bottleneck occurs upstream of an off-ramp segment and a queue of delayed vehicles is formed. The mainline flow coming into the off-ramp segment will consequently be less than the demand. In turn, the resulting number of vehicles that exit at a ramp should also decrease as it is likely that some of the delayed traffic was intended for off-ramp. This reduction in itself does not cause an issue, as the off-ramp demand can be modeled as a diverge percentage of mainline flow. However, as the queued mainline traffic is released upstream of the off-ramp, the issue becomes messy because there is no way to tell what percentage of the traffic is headed to the off-ramp. This uncertainty is especially problematic if the queue has existed for more than a single period, and occurs because a macroscopic model does not differentiate flows that become queued. The methodology essentially mixes all flow that is delayed, and extracting the destination of individuals when the flow resumes moving is impossible.

In the original formulation for the Cell Transmission Model, Daganzo assumes that all diverge percentages are known a priori. In the SO-DTA formulation of the CTM, Ziliaskopoulos addresses this issue by saying the known values can be used or proposes that the model can simply be left to determine the optimal diverge percentages. Alternatively, the HCM methodology attempts to maintain some approximation of FIFO by tracking the decreased mainline flow. This relationship is valid when queues remain small, but fails when they grow large. The methodology proposes that if there is a deficit in mainline flow accumulated during the preceding period, the off-ramp flow for the current period uses the previous period’s diverge percentage until mainline flow exceeding the deficit has passed. However, in the case in which the mainline deficit exceeds the mainline flow for a period (e.g., a large queue exists), this condition fails to fully capture the FIFO property.

Implementing this relationship faithfully into the model would require the use of integer variables, and since it fails under some conditions, this model operates with Daganzo’s initial assumption that diverge percentages $PD_{i,p}$ for each period are known. For practical purposes, these values can be computed either from the known demand inputs of the HCM methodology, or from observed traffic counts. Thus the off-ramp flow for a given node i at time step t in period p is given by the following equality:

$$OFRF_{i,t,p} = PD_{i,p} * MF_{i,t,p} \quad \forall i \in \tilde{F}, t \in S, p \in P \quad (2.31)$$

However, for some applications this may not be desirable, and Zilliaskopolous' second proposed solution can also easily be implemented for the model with a flow conservation constraint and a single upper bound constraint as shown in equations 2.32 – 2.34. Further these equations define an off-ramp flow deficit variable ($DEF_{i,t,p}^{OFR}$) to track unmet off-ramp demand. This is treated as a “deficit” in flow as opposed to a queue because the vehicles destined for the off-ramp are not queued up at the ramp, but rather are delayed and remain on a segment somewhere upstream of the off-ramp.

$$\sum_{t=1}^S \sum_{p=1}^P OFRF_{i,t,p} = \sum_{p=1}^P \frac{OFRD_{i,p}}{4} \quad \forall i \in \tilde{F} \quad (2.32)$$

$$DEF_{i,t,p}^{OFR} = \sum_{x=1}^{p-1} \left[\frac{OFRD_{i,x}}{4} - \sum_{t=1}^S OFRF_{i,t,x} \right] - \sum_{\tau=1}^{t-1} \left[OFRF_{i,\tau,p} - \frac{OFRD_{i,p}}{T_h} \right] \quad \forall i \in \tilde{F}, t \in S, p \in P \quad (2.33)$$

$$OFRF_{i,t,p} \leq \frac{OFRD_{i,p}}{T_h} + DEF_{i,t,p}^{OFR} \quad \forall i \in \tilde{F}, t \in S, p \in P \quad (2.34)$$

2.4.5 Objective Function

The previous sections define the desired flow properties of a solution as constraints in a linear programming formulation. The last piece of the model then becomes specifying an objective function that defines the optimal property of the flow. There are near limitless choices for objective function depending on the application of the model. In fact this is evidenced by much of the work presented in sections 2.3.2 – 2.3.4 where a wide variety of objectives are used. These range from those tailored to specific traffic management and network design applications, to those that strive to better approximate user optimal conditions.

One simple choice of objective could be to minimize the number of vehicles on each segment during each time step. This serves to encourage vehicles to proceed through the network as fast as possible, which in some ways approximates minimizing the travel time of the system. This formulation of the objective function is given in Equation 2.35.

$$\text{Minimize: } \sum_{i=1}^{N_s} \sum_{t=1}^S \sum_{p=1}^P NV_{i,t,p} \quad (2.35)$$

For a better approximation of the user optimal condition, the objective function can be modified to include weights that try to force flows to move forward instead of holding back. A simple way of enforcing this behavior is to penalize vehicles based on their location on the facility at the end of each time step. Such an objective function is shown in equation 2.36, which penalizes vehicles for remaining at upstream segments when it is possible to move forward (a similar approach was discussed more in depth by Zhu & Ukkusuri (2013)).

$$\text{Minimize: } \sum_{i=1}^{N_s} \sum_{t=1}^S \sum_{p=1}^P \left(1 - \frac{N_s - i}{N_s}\right) * NV_{i,t,p} \quad (2.36)$$

As a note, perhaps the most basic choice for objective would be to minimize the total travel time of the system. For the HCM freeway facilities methodology, the travel time for each segment i and each period p is computed as the segment length divided by the computed segment speed. However, the computed speed of a segment is the ratio of the average segment flow in a period $SF_{i,p}$ to the average number of number of vehicles on the segment in the period $NV_{i,p}$. Unfortunately, this relationship poses a major problem for the model as optimizing the sum of a set of ratios cannot be modeled in linear programming. Thus an alternative choice of objective function such as the one presented above must be used.

2.5 Computational Results and Model Applications

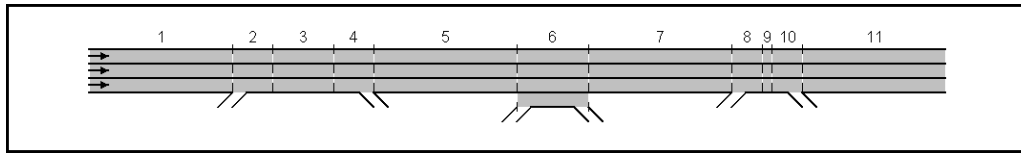
In order to test computational results for the framework, the LP model is built and solved for a set of example facilities. For each example, it is assumed that the underlying inputs for the model are known and have been calibrated for the specific location the facility is modeling (albeit the examples below all represent fictional locations). Performance measures such as speeds and intermediate values such as flows are obtained from the LP model solution and compared to those generated by the current HCM computational procedure. The comparisons are used to gain insight for each of the applications presented.

While there are a vast number of ways to build and solve a linear programming model, the number of variables for medium and large size problems will generally require the use of a commercial

solver. For the examples presented below, the Gurobi Optimizer API and Python were used to build and solve the linear program. For the initial analysis of the examples, the objective function given in equation 2.36 was used to better approximate the behavior of the underlying HCM methodology.

2.5.1 System Optimal Results and Peak Spreading

The first example facility is shown in Figure 2.5a and consists of 11 segments, including two on-ramps, two off-ramps, and a weaving segment. A three-hour study period from 4pm to 7pm was selected in order to analyze the facility under high levels of recurring demand. The speed contours for the facility over the course of the study period are shown in Figure 2.5b. The geometry and layout of this facility was chosen in order to represent the scale and complexity of a moderate sized facility. A summary of the size of the resulting linear program is given in Table 2.2.



(a) Geometry of the medium-sized example facility.

Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11
#1 16:00 - 16:15	59.0	52.6	57.1	55.7	58.3	47.4	57.7	52.3	52.3	55.7	57.6
#2 16:15 - 16:30	55.5	50.8	51.9	51.9	54.0	46.4	48.9	50.6	50.6	51.1	53.2
#3 16:30 - 16:45	55.5	50.8	51.9	51.9	54.0	45.3	43.7	50.6	50.6	51.1	53.2
#4 16:45 - 17:00	55.5	50.8	51.9	51.9	54.0	37.9	42.9	50.6	50.6	51.1	53.2
#5 17:00 - 17:15	54.8	50.5	51.1	51.1	53.1	27.3	42.9	50.6	50.6	51.1	53.2
#6 17:15 - 17:30	55.5	50.8	51.9	51.9	54.2	26.0	43.0	50.6	50.6	51.1	53.3
#7 17:30 - 17:45	55.5	50.8	51.9	51.9	53.3	24.2	42.9	50.6	50.6	51.1	54.5
#8 17:45 - 18:00	55.5	50.8	51.9	51.9	46.1	23.8	42.9	50.6	50.6	51.1	53.2
#9 18:00 - 18:15	57.1	51.5	54.2	54.2	54.0	22.5	42.9	50.6	50.6	51.1	52.6
#10 18:15 - 18:30	57.1	51.5	54.2	54.2	55.3	29.9	43.0	50.6	50.6	51.1	53.3
#11 18:30 - 18:45	56.0	51.0	52.7	52.7	54.6	36.6	43.0	50.6	50.6	51.1	53.3
#12 18:45 - 19:00	60.0	54.0	59.7	56.1	60.0	49.0	60.0	53.5	53.5	56.0	59.8

(b) Speed contour for the medium sized example facility.

Figure 2.5

Table 2.2 also gives some statistics that compare the solution of the LP to the solution generated from the current HCM computational procedure. For the example facility, there was an average absolute difference in the computed speeds of a little less than 1 mile per hour, and an average difference of around 5% in the predicted flows. These differences are due to the LP model only approximating some of the user optimal and FIFO conditions found in the min operations of the HCM model. Figure 2.6 shows the distribution of the average absolute speed differences across the segments of the facility. The speeds mostly differ at segments 6 and 7, which correspond to the

spillback from the facility bottleneck at segment 8 as seen in Figure 2.5b. Here the linear programming model is shifting vehicles in order to improve the overall performance of the system, which is reflected with by a lower average travel time.

Table 2.2 Summary of LP model size and solution for the first example facility.

Number of Decision Variables	158,928
Number of Constraints	202,132
Average Absolute Difference of Facility Speed	0.90
Average Absolute % Difference of Observed Flows	0.05%

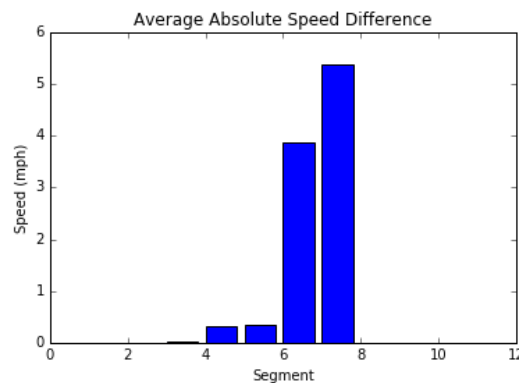


Figure 2.6 Average absolute speed difference between solutions for each methodology for each segment along the facility over the course of the three hour study period

While the solution of the LP model does not exactly match that of the computational procedure, the results of this system optimal approach have their own set of merits. Primarily, the computed flow rates can provide insight into the kinds of flow patterns that provide optimal system performance. For example, the constraints that load the facility at the first upstream segment can be relaxed to find flows that represent an optimal peak spreading of demand. This optimal spreading can then help to improve planning decisions for congestion management by better informing the decision making processes for things like toll pricing and managed lane availability. Additionally, the theory of sensitivity analysis for linear programs can be applied to determine the cost benefits of improving or altering flow within the model. Dual prices for the decision variables can be examined in order to

determine the points along a facility where an increase in throughput provides the highest system benefit.

2.5.2 Ramp Metering and Example

The use of dynamic traffic assignment constraints within an optimization framework lends itself directly to a wide array of traffic management applications. One of the most straightforward applications of the model presented in this paper is the determination of system-wide ramp metering rates. Ramp metering as a strategy is used to improve the congestion conditions and subsequent total travel time by managing the flow at on-ramps along a facility. While reducing the incoming flow at on-ramps may temporarily inconvenience some vehicles, the mainline flow conditions can improve greatly as a result, improving the system performance as a whole. The specifics of how ramp metering is implemented in the model can differ in a vast number of ways, but the core adjustments will likely always be the same. The objective function can be used to optimize a desired performance measure, while the on-ramp flow constraints allow the demand to vary in order to achieve the optimal behavior. Once the model has been solved, the on-ramp flows allowed at each step represent the effective metering rate.

A simple implementation of ramp metering is to change the objective function such that it minimizes both travel time and the number of delayed vehicles. Delayed vehicles are represented by two variables in the model, unserved vehicles (UV) on the mainline, and the queue for each ramp ($ONRQ$). Combining these with the previously mentioned travel time minimization yields the simple objective function shown in equation 2.37. This model computes metering rates for each 15 second time step of the oversaturated methodology. These full resolution rates can be used in a number of ways, but for the purposes of this research, the total on-ramp flow allowed for each analysis period is used as a fixed ramp metering rate for the corresponding 15 minute analysis period.

$$\text{Minimize: } \sum_{i=1}^{N_s} \sum_{t=1}^S \sum_{p=1}^P NV_{i,t,p} + UV_{i,t,p} + ONRQ_{i,t,p} \quad (2.37)$$

While the ramp metering strategy computed will be optimal for the LP model, it may not be viable in a real world scenario. For example, the metering rate may be so low that extreme queues form on the on-ramps, or no on-ramp flows at all are allowed onto the mainline. One way of countering this would be to enforce minimum flow rates by specifying lower bounds for $ONRF_{i,t,p}$ values. However, this may not be desirable as it requires the analyst to determine additional parameters for each ramp. Alternatively, upper bounds can be placed on the size of queues waiting at on-ramps, which

ensures ramp-flow will not be unrealistically limited. This limitation also reflects the assumption in the HCM methodology, and its computational engine, which both recommend that ramp metering cease when the on-ramp queue has reached or exceeded the on-ramp queue capacity (*Highway Capacity Manual 2016*).

In order to test the performance of ramp metering rates generated by the model, a second example facility based on a stretch of I-40 near Raleigh, North Carolina was tested. The facility is based on a 12.5 mile stretch of freeway, and its geometry is shown in Figure 2.7. The facility consists of 34 segments, and includes multiple lane changes at various points. A recurring bottleneck occurs at segment 32 on the facility, where the number of mainline lanes drops to two. Again, the three hour peak afternoon demand period from 4pm to 7pm was selected as the study period, and the facility demand data was based on NC default distributions. The set of demands used for the scenario result in a moderate amount of congestion during the study period. Figure 2.8a shows the speed contour of the facility, and it is clear there is a breakdown at the bottleneck occurring near the downstream end of the facility, with the congestion spilling back 11 segments during the worst analysis period.

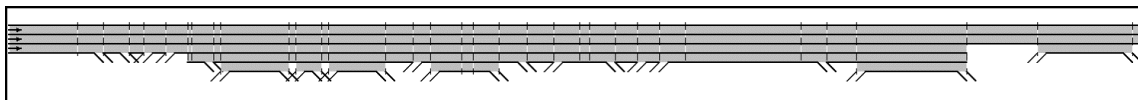


Figure 2.7 Geometry of the I-40 example facility.

Three metrics are of particular importance when evaluating the effectiveness of ramp metering. First, any reduction in average mainline travel time is desirable. Second, the vehicle-hours of delay (VHD) is important as it accounts for all delay in the system, including travel delay that has been pushed from the mainline to the on-ramps. If a ramp metering rate drops the average mainline travel time by 10%, but triples the VHD of the study period, the ramp metering is likely impractical for real-world applications. Lastly, it is important to consider the on-ramp queue occupancy over the course of the study period. On-ramp queue occupancy provides valuable insight into how the rates are shifting traffic, and can also highlight potentially impractical metering patterns.

Three sets of ramp metering rates were calculated for the example scenario, with each corresponding to different maximum allowed ramp queue size. The three maximum queue lengths considered were 150, 100, and 50 vehicles per lane. The results for each rate are shown in Table 2.3. In all three cases, the mean mainline travel time for the study period was reduced by over two minutes. As expected, there was an increase in VHD for all cases when ramp metering was used, but is offset by the improvement of the mainline travel time because mainline demand is significantly

larger than incoming ramp demands. Figure 2.8b shows the speed contour for the study period using the metering rates determined in Case 3. It can clearly be seen that the congestion on the mainline present in Figure 2.8a has largely been eliminated!

Table 2.3 Ramp metering results for the I-40 facility with high congestion.

Scenario	No RM	Case 1	Case 2	Case 3
Maximum Allowable ONRQ (veh)	–	150	100	50
Mean Mainline Travel Time	15.70	13.04	13.14	13.62
% Improvement	–	16.94	16.30	13.25
Veh Hours Delay	1735	2213	2219	2083
% Increase	–	27.55	27.90	20.06
Mean ONR Queue %	–	23.74	35.86	66.53
Avg ONR Queue (veh)	3.48	35.61	35.86	33.27
Max ONR Queue (veh)	149	146	98	50

Analysis Period	Segment																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
#1 16:00 - 16:15	65.3	65.2	64.6	67.4	61.5	42.8	42.8	61.9	66.6	56.7	65.2	56.6	62.7	57.8	68.4	63.4	69.4	69.4	69.4	67.0	68.0	63.0	66.6	66.6	67.6	62.6	62.2	64.1	64.1	66.4	48.1	53.3	63.3	53.3
#2 16:15 - 16:30	65.1	65.1	64.6	67.3	61.5	43.1	43.1	62.0	66.6	56.6	65.2	55.5	62.7	58.2	68.3	63.4	69.4	69.4	69.4	67.0	68.0	63.0	66.6	66.6	67.5	62.6	62.1	63.7	63.7	66.2	30.7	53.3	57.3	53.3
#3 16:30 - 16:45	64.9	64.9	64.6	67.1	61.4	43.5	43.5	62.1	66.6	56.5	65.1	55.4	62.7	58.1	68.3	63.4	69.4	69.4	69.4	67.0	68.0	63.0	66.5	66.5	67.5	62.5	62.1	63.6	63.6	66.1	22.0	53.3	51.9	53.3
#4 16:45 - 17:00	64.7	64.7	64.5	67.0	61.3	43.8	43.8	62.2	66.6	56.4	65.1	55.3	62.6	58.1	68.3	63.4	69.4	69.4	69.4	67.0	67.9	62.9	66.4	66.4	67.4	62.5	62.1	63.5	63.5	64.0	16.8	53.3	47.2	53.3
#5 17:00 - 17:15	64.4	64.4	64.6	66.7	61.3	44.2	44.2	62.3	66.5	56.9	65.2	55.8	62.9	58.3	68.2	63.4	69.4	69.4	69.4	67.0	67.9	63.5	60.8	60.4	37.4	37.2	38.3	38.3	32.5	24.4	15.4	53.3	43.3	53.3
#6 17:15 - 17:30	66.2	65.3	64.7	67.9	61.8	41.2	41.2	61.5	66.5	57.7	65.3	56.6	63.2	59.0	68.4	63.5	69.4	69.5	68.1	59.4	40.6	27.2	24.6	28.2	22.6	23.3	31.7	37.6	33.9	26.3	15.6	53.3	40.7	53.3
#7 17:30 - 17:45	67.6	65.4	64.9	68.8	62.3	48.2	48.2	63.5	67.7	58.5	66.8	57.5	65.2	59.6	69.1	63.9	69.4	69.8	48.2	34.5	22.6	23.1	24.9	30.3	25.1	26.5	32.0	37.8	37.3	26.8	15.6	53.3	40.3	53.3
#8 17:45 - 18:00	68.6	65.5	65.0	69.3	62.8	57.2	57.2	66.2	69.0	59.3	68.8	58.4	67.5	60.3	69.8	64.4	69.5	69.9	70.0	60.0	39.6	37.7	30.8	34.7	29.3	31.0	33.9	38.8	37.3	27.2	15.5	53.3	43.1	53.3
#9 18:00 - 18:15	69.4	65.6	65.0	69.3	63.2	59.7	59.7	66.9	69.2	60.0	69.5	59.1	67.7	60.8	69.8	64.8	69.5	69.9	70.0	67.9	69.8	64.5	69.0	67.9	60.8	56.3	60.7	53.6	44.0	30.7	15.5	53.3	48.1	53.3
#10 18:15 - 18:30	69.8	65.6	65.1	69.3	63.5	61.4	61.4	67.4	69.3	60.8	69.7	59.9	67.9	61.4	69.8	65.1	69.5	69.9	70.0	68.1	69.9	64.8	69.0	68.1	69.8	64.6	64.3	69.8	67.4	69.8	20.0	53.3	57.1	53.3
#11 18:30 - 18:45	70.0	65.6	65.2	69.3	63.8	62.6	62.6	67.8	69.4	61.6	69.8	62.0	68.3	61.8	69.8	65.3	69.5	69.9	70.0	68.3	69.9	65.1	69.1	68.3	69.8	65.0	64.7	70.0	67.6	69.8	54.1	61.6	70.0	60.8
#12 18:45 - 19:00	70.0	65.7	65.2	69.3	64.0	63.5	63.5	68.1	69.5	62.4	69.8	62.8	68.5	62.4	69.9	65.6	69.6	69.9	70.0	68.5	69.9	65.3	69.1	68.5	69.8	65.3	65.0	70.0	67.9	69.8	70.0	68.5	70.0	68.5

(a) Speed Contour of the I-40 facility under moderate congestion

Analysis Period	Segment																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
#1 16:00 - 16:15	65.3	65.2	64.6	67.4	61.5	42.8	42.8	61.9	66.6	56.7	65.6	56.6	63.5	57.8	68.7	63.7	69.4	69.6	69.6	67.1	68.5	63.2	67.3	67.1	68.1	62.9	62.5	65.0	65.0	67.1	53.6	53.3	66.3	53.3
#2 16:15 - 16:30	65.1	65.1	64.6	67.3	61.6	42.8	42.8	61.9	66.6	56.6	65.1	55.5	62.9	58.2	68.5	63.5	69.4	69.5	69.5	67.0	68.1	63.1	66.8	66.8	67.7	62.7	62.2	64.1	64.1	66.5	40.7	53.3	65.8	53.3
#3 16:30 - 16:45	64.9	64.9	64.6	67.1	61.6	42.7	42.7	61.9	66.6	56.5	65.3	55.4	63.3	58.1	68.6	63.7	69.4	69.8	69.8	67.2	68.8	63.4	67.8	67.2	68.5	63.0	62.8	65.9	65.9	67.7	32.4	53.3	65.3	53.3
#4 16:45 - 17:00	64.7	64.7	64.5	67.0	61.5	43.3	43.3	62.1	66.6	56.4	65.1	55.3	62.6	58.1	68.5	63.5	69.4	69.5	69.5	67.0	68.1	63.2	67.0	67.0	67.8	63.0	62.4	64.7	64.7	66.9	26.5	53.3	64.7	53.3
#5 17:00 - 17:15	64.4	64.4	64.6	66.7	61.3	44.2	44.2	62.3	66.5	56.9	65.4	55.8	63.1	58.3	68.4	63.5	69.4	69.5	69.5	67.1	68.2	63.1	66.9	66.9	67.7	62.8	62.4	64.6	64.6	66.7	21.4	53.3	64.6	53.3
#6 17:15 - 17:30	66.2	65.3	64.7	67.9	61.8	48.4	48.4	63.6	67.4	57.7	66.4	56.6	64.5	59.0	68.8	63.7	69.4	69.7	69.7	67.2	68.6	63.3	67.5	67.2	68.2	63.0	62.6	65.4	65.4	67.3	17.6	53.3	63.0	53.3
#7 17:30 - 17:45	67.6	65.4	64.9	68.8	62.3	53.6	53.6	65.1	68.5	58.5	67.7	57.5	66.3	59.6	69.4	64.1	69.4	69.9	69.9	67.4	69.3	63.7	68.6	67.4	69.1	63.5	63.1	67.1	66.6	68.5	16.6	53.3	62.6	53.3
#8 17:45 - 18:00	68.6	65.5	65.0	69.3	62.8	57.2	57.2	66.2	69.0	59.3	68.8	58.4	67.5	60.3	69.8	64.4	69.5	69.9	70.0	67.7	69.8	64.1	68.9	67.7	69.6	63.9	63.6	68.4	66.9	69.3	18.7	53.3	69.8	53.3
#9 18:00 - 18:15	69.4	65.6	65.0	69.3	63.2	58.6	58.6	66.6	69.1	60.0	69.2	59.1	67.7	60.8	69.8	64.5	69.5	69.9	70.0	67.7	69.8	64.2	68.9	67.7	69.7	64.1	63.7	68.6	67.0	69.4	24.8	53.3	69.7	53.3
#10 18:15 - 18:30	69.8	65.6	65.1	69.3	63.3	61.3	61.3	67.4	69.3	60.8	69.7	59.9	67.9	61.4	69.8	64.8	69.5	69.9	70.0	67.9	69.8	64.3	69.0	67.8	69.8	64.0	63.6	68.5	66.9	69.3	37.8	53.3	67.9	53.3
#11 18:30 - 18:45	70.0	65.6	65.2	69.3	63.8	62.0	62.0	67.6	69.4	61.6	69.8	62.0	68.3	61.8	69.8	65.1	69.5	69.9	70.0	68.1	69.9	64.7	69.0	68.1	69.8	64.6	64.3	69.7	67.4	69.8	70.0	62.3	70.0	61.5
#12 18:45 - 19:00	70.0	65.7	65.2	69.3	64.0	63.5	63.5	68.1	69.5	62.4	69.8	62.8	68.5	62.4	69.9	65.6	69.6	69.9	70.0	68.5	69.9	65.3	69.1	68.5	69.8	65.3	65.0	70.0	67.9	69.8	70.0	68.5	70.0	68.5

(b) Speed Contour of the I-40 facility using ramp metering rates determined by the LP model for the 50 vehicle maximum queue length.

Figure 2.8 Speed contours for the I-40 facility before and after ramp metering.

2.6 Conclusions and Future Work

This paper presents a novel optimization framework of the oversaturated model of the freeway facilities methodology of the Highway Capacity Manual. The framework is built as a linear programming based dynamic traffic assignment model. The formulation brings advantages over the original iterative solution scheme by providing new ways in which analysis can be conducted. Its declarative as opposed to descriptive nature provides new options for analyzing the interplay between the inputs, outputs and parameters of the underlying methodology. The linear nature of the constraints presented and developed for the model provide it with a number of desirable qualities. The model is extremely flexible, and can utilize known efficient solution techniques for linear programming. Additionally, the model benefits from the vast body of theory that exists and documents the usefulness of linear programming models for in-depth sensitivity analysis.

The model can either be used directly or easily extended for most applications where optimization of a network is desirable. It can be used to investigate optimal spreading of demands in order to better understand the behavior of a freeway facility and inform decision making processes. The model can be utilized for various active traffic management analysis, and in particular can be used for the computation of system optimal ramp metering rates. The numerical examples of this paper show that the model can compute ramp metering rates within realistic bounds and constraints that are effective in improving the overall performance of a facility.

A number of directions for future research exist based on the findings of this paper. Additional work on the ramp flow constraints can be done to better approximate the desired "first-in, first-out" flow condition. More complex system ramp metering strategies can be explored, and the model could be investigated as a real-time reactive strategy consisting of an iterative solution process revolving around short-term predicted demands. Further, the model's use for network design problems was not explored, but the addition of binary variables would open the model up for these types of decision making applications. Additionally, while the presented model is considered to be system optimal, integer variables could be introduced to better approximate user optimal conditions by strictly enforcing the minimum and maximum conditions of the flow relationships.

Finally, any analyses conducted using the techniques proposed in this work are only useful if the underlying core model has previously been calibrated to match real-world conditions. Calibration represents a major challenge when creating a model, and no comprehensive procedure currently exists to guide a user in the process. Future work is needed to simplify or even automate this process in order to maximize the usefulness of the proposed optimization framework. While it raises issues of computationally tractability, introducing integer variables to model the exact flow relationships of the HCM's computational procedure could allow the model to be solved in a "backwards" manner,

i.e., an inversion of the decision variables to compute inputs of the methodology from known outputs. This unfortunately cannot be done with the pure LP model due to the existence of the *holding-back* phenomenon. Since performance measures such as speeds are more easily obtained from real world data than many of the methodology's required inputs, this could be very beneficial for both demand volume estimation and facility calibration.

CHAPTER

3

CALIBRATING FREEWAY FACILITIES WITH GENETIC ALGORITHMS AND THE HCM

3.1 Introduction and Motivation

The Highway Capacity Manual (HCM) is a widely used tool for conducting freeway facilities analysis for planning level and operational applications. The HCM defines a *freeway facility* as an extended length of freeway composed of contiguously connected basic freeway, weaving, merge, and diverge segments (*Highway Capacity Manual* 2016). Figure 3.1 represents a simple freeway facility that could be analyzed using the methodology. A facility is generally a 9-15 mile portion of a single freeway, and each analysis is conducted for a study period typically falling within a single 24-hour time span. Each study period is further divided into individual 15 minute analysis periods, each of which requires a distinct set of inputs.

When conducting a freeway facilities analysis, an analyst must first calibrate the model to their specific study site before it can be utilized for planning or operational purposes. The calibration process seeks to ensure that the methodology reports performance measures consistent with available

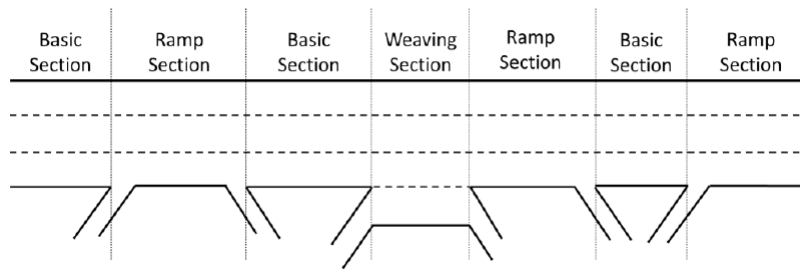


Figure 3.1 Example of a basic facility modeled in the HCM freeway facilities model (*Highway Capacity Manual 2016*).

real-world data. Calibration is most often done for recurring daily congestion conditions, which provide a baseline for other analyses such as reliability and active traffic demand management studies. Each model can be calibrated by tuning a number of adjustable parameters until an acceptable level of matching is obtained. While the methodology requires a large number of inputs, many of them are known explicitly *a priori*. Thus calibration of a facility is largely done through the manipulation of parameters relating to input demand volumes and segment capacities.

The crux of the methodology is the relationship between traffic demand volumes and segment flow capacities. When demand does not exceed capacity, all demand is met, and speeds, densities, and other performance measures can be easily estimated. However, once demand exceeds capacity, a flow breakdown occurs at the bottleneck, and complex sets of equations are required to model the ensuing repercussions. Thus the specification of the inputs that determine the capacity and demand values is central to the ability of the methodology to accurately model congestion conditions. Segment capacities are largely determined by various physical and geometric aspects of the facility being modeled, while input demand volumes, which define the vehicle flows seeking to enter or exit the facility during each analysis period, are largely unknown in the real world and must be estimated to the best of the ability of the analyst.

Demand estimation represents a major challenge in using the methodology for a number reasons. First, observable performance measures largely do not directly correspond to demand volumes, especially under congested flow conditions. Consequently, demands must be manually estimated using an analyst's knowledge of the study area combined with available real-world speed or flow data. Additionally, the methodology requires a large number of demand inputs as volumes are needed for each 15-minute analysis period at each location where vehicles enter or exit the system. For even moderate sized facilities, this can exceed over a thousand individual demand volume inputs. Finally, the relationships between demand inputs and performance measure outputs under

congested conditions are highly interrelated and complex, and even slight variations in demand can have far-reaching and unpredictable effects. This sensitivity makes diagnosing estimation errors and prescribing fixes in any algorithmic way extremely difficult.

Currently these complicating factors require that an analyst undertake a lengthy and often difficult calibration process in order to match predicted performance measures to observed real-world data. Further, there is only limited existing guidance concerning calibration provided in the HCM. For example, the upcoming 6th edition of the manual recommends a process largely involving repeated parameter estimation and adjustments. In many cases, calibration comes down to trial and error manipulation of a large number of inputs until the analyst can achieve a satisfactory level of matching. Even then, there is no guarantee that an analyst will be able to calibrate the facility to the level where it can provide meaningful results.

The goal of this research is to provide a new framework for facility calibration through the manipulation of input demand volumes. A genetic algorithm metaheuristic is implemented to estimate and adjust demand profiles that result in predicted performance measures consistent with provided real-world data. The framework is automated and greatly reduces the burden placed on the analyst. The framework accounts for two approaches that can be used depending on the quality of the data available to the analyst when modeling the facility. First, when high quality information is available, the problem is treated as a system identification problem and an existing profile is adjusted. Second, a novel demand estimation technique is proposed for use in situations where high quality data is not available. The approach is shown to be flexible enough to account for most profiles that have been observed under real-world conditions, and is generalized for use under any conditions. Finally, calibrations for a real-world case study in Raleigh, North Carolina, are developed and compared using both approaches.

3.2 Calibration through Demand Estimation

The freeway facilities methodology is tightly built on demand volumes as the main inputs for all freeway segments. Demand volumes are converted into densities, which are used to estimate flow and congestion conditions along the facility. Unfortunately, despite demand volumes being the critical input for a freeway facilities analysis, they are mostly unknown in real-world situations. In practice, most available data comes from sensors such as loop detectors or manual counts, which can only provide actual flows and travel times along sections of a freeway. In the context of the freeway facilities methodology, these types of measures are considered outputs as opposed to inputs since they represent the performance of the facility.

Consider, as an example, a set of observations recording the observed flows along a portion

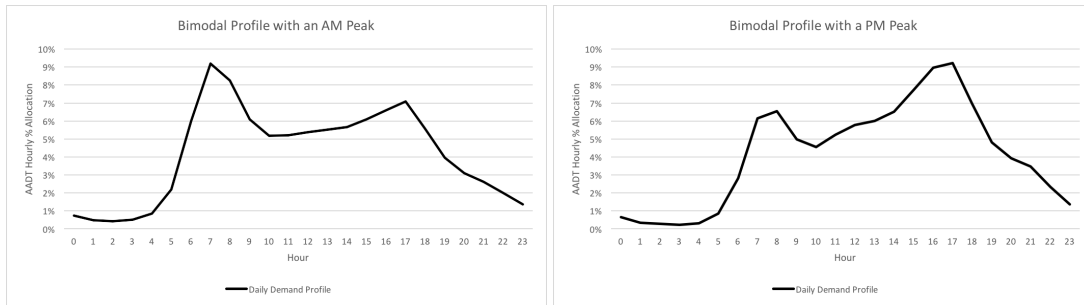
of a facility. Under conditions where there is little to no congestion, these flows would accurately represent the underlying demands.¹ However, when there is heavy congestion, this may not necessarily be true. In this case, observed flows represent the number of vehicles that were able to successfully traverse the segment in the observation period, but not necessarily the number of vehicles that desired to cross the segment. The former represents a measure of performance (a methodology output), and the latter represents the underlying demand (a methodology input). In other words, while demand is theoretically unlimited, observed flow is always capped at the capacity² of a segment or freeway facility. This hard limit requires that analysts estimate underlying demand volumes from available sensor data in order to conduct an analysis on any situation where demand exceeds capacity, such as a bottleneck analysis.

The goal of the demand estimation problem is to determine values for the set of entering and exiting demand volume inputs of a freeway facility such that performance measures predicted by the HCM methodology match a set of real-world values. Demand volume inputs are required at the initial upstream mainline segment, as well as for each on-ramp and off-ramp segment located along the facility. At each entry or exit point, demand volumes vary over time, and the rise and fall of the values over the course of the study period can be thought of as forming a volume profile for each particular location. In contrast, the capacity of a segment is fixed over time and is estimated using a combination of physical aspects (e.g., number of lanes) and additional values such as estimated free flow speed. As such, this work chooses to focus solely on the problem of calibration through demand estimation as it requires the manipulation of a larger number of variables and consequently tends to be the more difficult of the two.

Beyond yielding predicted outputs consistent with observed values, it is also important that the demand calibration process result in profiles that are viable in the real world. This type of feasibility requires that any profile be sufficiently smooth and reasonably reflect known daily volume trends. A sufficiently smooth profile will ensure that the volume inputs maintain real world feasibility as demands can only vary so much between consecutive 15-minute analysis periods. Profiles should also be expected to reasonably reflect known trends for daily profile patterns, especially when facility specific information is available. Figure 3.2 shows examples of behavior represented in demand profiles over a 24-hour period. In general, it is likely that the rough behavior of daily traffic patterns will be known to the analyst (e.g., the approximate time and magnitude of peak-hour volumes), and this knowledge can then be used to inform the selection of the shape of the estimated demand profile. An analyst will likely also have access to average annual daily traffic (AADT) values for the facility, which can be combined with knowledge of the shape of the profile to form a reasonably

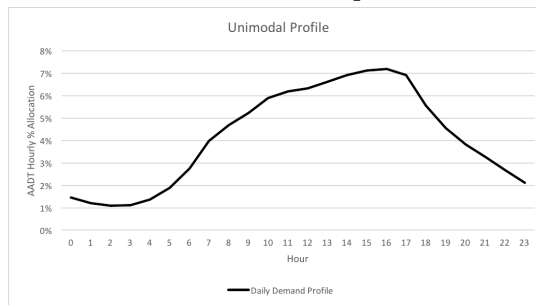
¹Unless there is a bottleneck upstream of the uncongested section metering the incoming traffic.

²A segment's capacity may be further constricted by downstream congestion conditions.



(a) Example of a Bimodal with AM peak demand profile.

(b) Example of a Bimodal with PM peak demand profile.



(c) Example of a unimodal demand profile.

Figure 3.2 Examples of well-known behavior exhibited in daily demand profiles (NCDOT, 2016).

restricted search space for demand volumes.

3.3 Literature Review

This section presents a review of literature related to calibration of the HCM freeway facilities methodology and related methods. We find that little research has been conducted beyond what is given in the highway capacity manual concerning calibration. Much more research has been conducted for related methods such as the cell transmission model (CTM), the hydrodynamic model of flow from which the HCM methodology draws. Following that, a brief literature review is presented on system identification problems and the use of metaheuristics such as genetic algorithms as solution techniques.

3.3.1 HCM Calibration and Hourly Demand Profiles

The 6th edition of the *Highway Capacity Manual* (2016) provides guidance on how to address model calibration in the form of a five-step process as shown in Figure 3.3. The first step requires the user to

gather the required input data for the model including geometric aspects, segment free flow speeds (FFS), and initial demand volume estimates for the mainline and ramps at each analysis period of the study period. No guidance is provided to address the process of initial demand estimation. The manual breaks the calibration adjustment process into three pieces: free flow speed calibration, bottleneck capacity calibration, and demand adjustment. Free flow speeds can be calibrated using observed measures for uncongested or off-peak periods. For bottleneck capacity calibration, the HCM advises a trial and error strategy. In the event the location and or severity of congestion predicted by the model does not match that of an observed set of target performance measures, the manual recommends that an analyst repeatedly adjust the capacity of bottleneck segments by 50 pc/h/ln until an acceptable level of matching is achieved. Identification of bottleneck segments is left up to the analyst's knowledge of the facility. For demand estimation, the manual advises that an analyst manually adjust volumes in increments of 50 pc/h/ln until predicted model travel times and bottleneck queue lengths are within a specified margin of error.

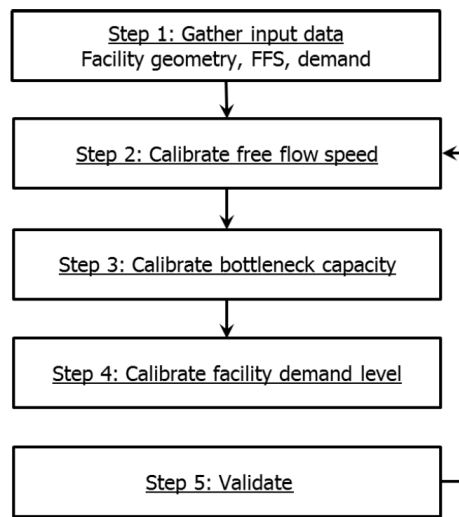
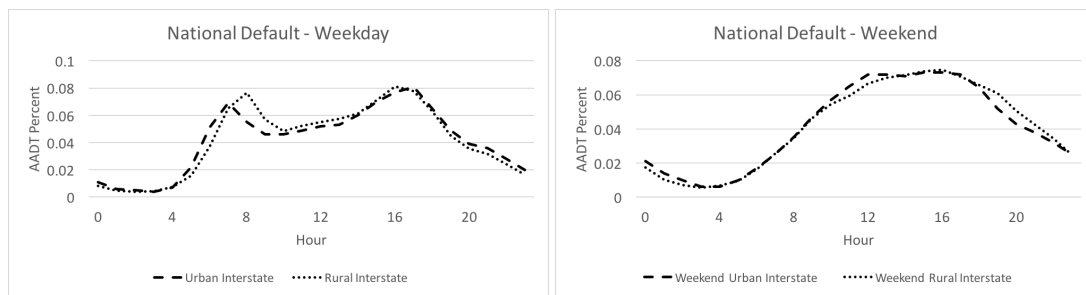


Figure 3.3 Core freeway analysis calibration framework in the 6th edition of the HCM (*Highway Capacity Manual* 2016).

In order to estimate the initial set of demand inputs, an analyst generally makes use of an average annual daily traffic (AADT) value projected into individual demand volumes using an hourly demand profile. An hourly demand profile provides a breakdown of the percentage of an AADT value occurring during each hour of the day. Individual profiles are often highly specific to the each

location, but general trends are common across most facilities. Demand profiles for weekdays are often bimodal in that they contain two clear peaks. In this case, one peak typically corresponds to travelers on their morning commute, and one corresponds to travelers commuting home from work in the evenings. In contrast, weekend profiles are often unimodal, or have a single peak, that builds more slowly throughout the day. Hallenbeck et al. (1997) developed a set of national default profiles for weekdays and weekends, broken down into both urban and rural categories. Both bimodal and unimodal behavior are reflected in the national default profiles, shown in Figure 3.4.



(a) National default weekday profile shape.

(b) National default weekend profile shape.

Figure 3.4 National default hourly demand profiles for weekdays and weekends on urban and rural interstates.

Additionally, a study conducted for the state of North Carolina provides a further level of detail and divides the profiles into 3 default categories: unimodal, bimodal with an AM peak, and bimodal with a PM peak (NCDOT, 2016). Figure 3.2, above, shows examples of each of these three types. The unimodal profiles are very similar to the weekend category of profiles, while the bimodal profiles provide two variations of the weekday profile. Further, six additional profile shapes are provided for each of three main profile types. These represent the minimum, the 25th percentile, the average, the 75th percentile, the maximum, and median of all profiles generated from the collected data.

3.3.2 Calibration of the CTM and Related Models

The cell transmission model is commonly used as a basis for automated density estimation and calibration approaches. Some of the earliest work on this is presented by Munoz et al. (2003) who develop the modified cell transmission model (MCTM) and the switching mode model (SMM). This formulation differs from Daganzo's basic CTM in three key ways. First, cell densities are used as state variables as opposed to cell occupancy, directly leading to the second difference in that the

MCTM can handle non-uniform cell lengths. Lastly, the formulation allows congested conditions at the downstream end of a facility. These changes lead to some nonlinearities in the difference equations regarding moving congestion wavefronts, so the authors develop the SMM to return to a piecewise linear formulation. The SMM accounts for five different congestion status modes that allow the model to switch between several sets of linear difference equations. The authors are also able to show that the CTM and the MCTM/SMM can estimate demand within freeway facilities with reasonable accuracy.

A fair amount of density estimation research has come out of the development of the MCTM and the SMM. Muñoz et al. (2004) propose a methodological calibration procedure for the CTM/MCTM and further show the methods accurately estimate density on longer stretches of freeway. This procedure is further explored by Zhong et al. (2014), but their proposed nonlinear optimization framework again has the deficiency of an inability to reliably find globally optimal solutions. The MCTM/SMM has been extended to include stochastic variations in traffic, especially with respect to their effects on the fundamental flow-density diagram. The Stochastic Cell Transmission Model (SCTM) directly incorporates stochastic noise into the parameters of the fundamental diagram (Sumalee et al., 2011). Pascale et al. (2013, 2014) develop an alternative approach in which they use a Bayesian framework to account for stochasticity, and propose a particle filtering approach to estimate density. Both methods are shown to be reliably accurate in matching sample detector data. Most recently, a robust switching approach for the SMM is introduced by Morbidi et al. (2014) in situations where the congestion wave speed is unknown.

Similar alternatives to the mechanisms of the MCTM/SMM for state detection have also been developed. Stankova & De Schutter (2010) use a jump Markov linear model (JMLM) to account for switching between free-flow and congested states, while Aligawesa and Hwang propose a State Dependent Transition Hybrid Estimation (SDTHE) algorithm motivated by the SMM but with improved state detection. Further research has been conducted into density estimation and state prediction loosely based on the LWR equations and the CTM, though most of the methods developed rely on relaxations and heuristics to even attempt to find solutions (e.g., Lu et al. (2013) and Ma et al. (2015)).

3.3.3 System Identification and Metaheuristics

System identification was initially developed as a technique to construct mathematical models of dynamic systems based on observed data (Ljung, 2010). The approach provides an effective way of modeling systems where relationships between inputs and outputs are either unknown or poorly understood to the point where an exact model cannot be built. For the approach, systems are

generally classified in terms of a "transparency" on the scale between white and black. A "white-box" model of a system is one where all relationships are known and have been defined. For a "black-box" model little to nothing of the true relationships are known or can be defined (De Nicolao, 1997). There also exist a number of variations of "grey-box" models that fall between the two extremes where varying amounts of information are known (Ljung, 2010). In approaching a grey-box model, a basic understanding of the relationships is assumed, but the complexities are not fully or explicitly considered.

Metaheuristics are a class of search approaches that "permit an abstract level of description" and are used to "efficiently explore the search space in order to find (near-) optimal solutions" (Blum & Roli, 2003). At their core, metaheuristics are not problem-specific, but implementations can make use of "domain-specific knowledge...controlled by the upper level strategy" (Blum & Roli, 2003). This lack of specificity paired with an abundance of flexibility makes metaheuristic approaches particularly useful for solving grey or even black-box system identification problems. There are a wide variety of types of metaheuristics that have been developed including but not limited to simple local search techniques, evolutionary algorithms, simulated annealing, and artificial neural networks (Gendreau & Potvin, 2005).

A genetic algorithm (GA) is an evolutionary search metaheuristic commonly used for complex optimization problems where the underlying mathematics preclude the use of classical approaches. The approach draws from the processes of evolution and natural selection in order to generate an optimal or near-optimal solution by "evolving" a pool of candidate solutions over time (Goldberg, 1989). Each candidate solution is represented as an individual *organism*, with the decision variables of the problem encoded as the organism's "genes". An evaluation function is defined in order to calculate the *fitness* of a candidate solution based on these genes. By choosing an encoding and fitness function such that the problem is framed as a minimization or maximization of fitness, fitness-based "competition" can be used to make statements on the quality of one candidate solution versus another. In this way, the fitness function is essentially analogous to the objective function of classical optimization techniques like linear programming.

The most critical aspect of using a GA is the process of defining the encoding of a candidate solution as the genes of an organism. In fact, the key underlying assumption to the GA approach is that high-quality solutions will contain common "building blocks" in their encoded genetic material (Mitchell, 1998). Maintaining these blocks within a population of organisms, and combining them with other "good" blocks should then theoretically lead to the discovery of better solutions. Blocks can be combined through the use of a *crossover* operator that represents the "mating" of organisms. While real-valued encodings are occasionally used, most encodings consist of some type of binary representation of the solution. This allows for a straightforward realization of genetic building blocks

(e.g., small binary patterns such as “010” or “110”), as well as providing a simple notion of crossover through exchanging binary digits.

3.4 Genetic Algorithm System Identification Approach

In many ways, demand estimation for the freeway facilities methodology can be thought of as a system identification problem. The output performance measures are known, but the exact inputs leading to these conditions stem from complicated relationships and compounding interplay between model inputs. While a “white-box” model in the form of the methodology exists for computing outputs from inputs, its complexities preclude its feasibility as a model for the inverse problem of computing inputs from outputs. These complexities stem from the heavy reliance on a strict “minimum” relationship between desired demand inputs and available flow capacities. Each minimum introduces a discontinuity, and the large number of minimum instances that occur makes it nearly impossible to “reverse compute” inputs from outputs. As a workaround for this limitation, system identification allows for the methodology to be treated as a “grey-box” model. A metaheuristic such as a genetic algorithm can then be wrapped around the methodology, and the parameters can be optimized without explicitly considering the mathematics of the underlying model. Since the underlying mathematics of the methodology are both nonlinear and discontinuous, a metaheuristic approach is used to avoid intractabilities that would likely arise in the use of more classical techniques.

A genetic algorithm begins with an initial *population* of randomly generated organisms, and the search process operates by “evolving” this population over time. The evolution process primarily consists of competition-based *selection* and *crossover* of organisms in order to create new generations of the population. The *selection* operator identifies high-quality candidate solutions fit to undergo a “mating” process and produce new “offspring” organisms. By using a *crossover* operator to replicate the hereditary mixing and inheritance of genetic material, each offspring will contain some of each “parent” organism’s genes. The group of offspring created from a given population forms a new generation, with each successive generation providing an opportunity to identify a better candidate solution. Additional operators such as *mutation* and *preservation* can be utilized to help prevent premature convergence to undesirable local optima, while *elitism* can be enforced to improve the overall rate of convergence. The evolutionary process of a genetic algorithm repeats until either an acceptably optimal solution is found, or a maximum number of iterations is reached. Figure 3.5 shows the process flow for a simple genetic algorithm.

The following sections present a detailed discussion of the genetic encodings and implementation of the operators discussed above for the demand estimation and calibration problem.

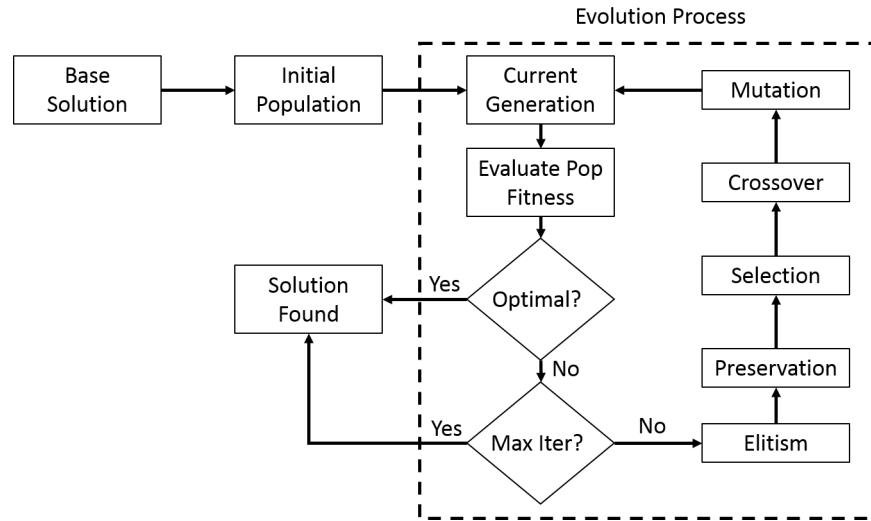


Figure 3.5 Flow chart detailing the process of a genetic algorithm.

3.5 Genetic Encoding of the Problem Space

While the formulation of the fitness function for the demand estimation problem is largely straightforward (covered in Section 3.6.1), defining an appropriate genetic encoding of the decision space is a much more difficult endeavor. This difficulty is primarily due to the concerns raised at the end of Section 3.2. These concerns stem from the notion that the allocation of demand over the study period must both be physically feasible and reflect realistic behavior. As such, any encoding used for the problem must address these additional constraints, either in an implicit or explicit manner.

This work explores two different encoding approaches for the demand estimation problem. The first approach assumes that the analyst has high quality data regarding both the average annual daily traffic (AADT) value for the facility as well as for the hourly demand profile of the study period. This information is used to create a search space around the known profile that explicitly restricts the candidate solution's shape and smoothness. The second approach assumes that AADT values are known, but either little or poor quality data is available regarding the demand profile, and uses a weighted combination distribution of probability functions to generate each profile from the ground up. The resulting profiles are inherently smooth, and the location and shape parameters for each probability distribution can be varied or restricted to incorporate information about the shape of the demand profile when available. Further, since each approach considers a single profile, both encodings can be used concurrently and can vary between each entry and exit point for a facility based on the availability of data.

The following two sections describe each approach in depth. However, before doing that it is important to first note a shared attribute of the encodings. While the two represent vastly different approaches to the problem, both encodings convert a demand profile into a single binary string. Since each candidate solution consists of a group of demand profiles, both encodings will then simply produce an array of binary strings as the genetic material of a representative organism. This shared representation is extremely important as it allows for a single genetic algorithm framework to be developed that can approach the problem using both encodings. Consequently, the evolution strategies defined for the GA framework in Section 3.6 are designed for use on arrays of binary strings and can be used to operate on organisms encoded using either approach.

3.5.1 Profile-Based Encoding

The profile-based encoding approach requires that an analyst start with high quality information about both the AADT value for the study period and the shape of the underlying demand profile. This approach is best used when the analyst is confident in his or her projected demand profile, but the performance measures predicted by the methodology do not match observed values. This indicates that the demand volumes are likely close to what they should be, and can be varied within a specified confidence interval in order to find the optimal distribution of demand. An example of a known demand profile shape with a specified 15% confidence search interval is given in Figure 3.6.

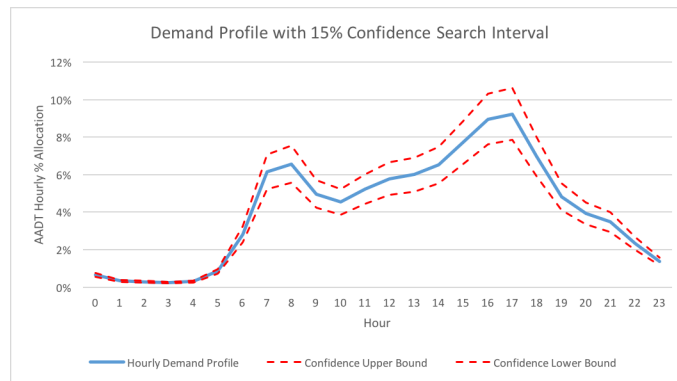


Figure 3.6 Example of an input demand profile shape with a 15% search interval specified around it.

The HCM freeway facilities methodology includes a set of parameters that can be used to adjust input demand volumes. These values are called Demand Adjustment Factors (DAFs), and exist for every segment where vehicles enter or exit the facility. Just as with demands, DAFs vary over time

and each 15-minute analysis period can be assigned a unique corresponding adjustment. During the computational procedure of the methodology, the DAFs for a segment are applied as multipliers to the entering or exiting demand volume input for each analysis period. DAFs provide a perfect mechanism for this modeling approach as a confidence interval of 15% can easily be represented by restricting the value of each DAF to being between 0.85 and 1.15. More generally, a confidence percent of α is represented by a DAF interval of $[1 - \alpha, 1 + \alpha]$.

For this encoding, each DAF is represented as a binary string of fixed length n . By discretizing the given confidence interval with a step size of $1/2^n$, the integer value corresponding to this string can then be projected into the interval to compute the corresponding real-valued DAF. To demonstrate, if the search space is the interval $[0.85, 1.15]$, and an 8-digit binary encoding is used, the discretized step size is $1/2^8 = 0.00117$. If a gene is expressed by the binary string "00001111", it is converted to base 10 as an integer value of 31, and the resulting DAF is $0.85 + (31 * 0.00117) = 0.97$. Further, the resolution of the search space can easily be adjusted by simply increasing or decreasing the number of binary digits considered.

The set of DAFs for a single segment can be thought of as representing the estimated demand profile for that entry or exit point when combined with the fixed demand volume inputs. Since the initial demand inputs provided by the analyst are considered to be fixed in this case, the set of DAFs is all that is needed to maintain a representation of a candidate solution demand profile. Thus, with each individual DAF encoded as a binary string of a specified length, the set of DAFs for a single segment can be concatenated into one long string that represents the entire profile for the study period. Once this is done, the set of all DAFs for a facility, which normally constitutes a two-dimensional array of real-valued numbers, has been converted directly to a one-dimensional array of binary strings.

3.5.2 N-Distribution Encoding

While an analyst often has access to reliable information regarding the magnitude and daily behavior of demand for their facility and study period, there are still many cases where such information is unavailable. This renders use of the approach of Section 3.5.1 largely untenable, as the algorithm is nearly guaranteed to converge to a sub-par local optimum due to the search space being entirely dependent on the initial guess for the shape of the profile. In order to circumvent this shortcoming, this section proposes a novel demand estimation encoding that can be used in the GA framework to build viable profiles from scratch. The approach is hinged on one key assumption: the daily distribution of traffic demand can be represented by the weighted combination of n continuous random variable distributions. The density function of the resulting mixture distribution can then

be paired with an AADT value to generate a demand profile for the study period.

First, it must be established that the role of each daily demand profile is analogous to that of a probability density function for a continuous random variable. At its core, a daily demand profile represents an allocation of AADT over a 24-hour period. The percent of AADT for each analysis period represents the density of daily demand assigned to that time interval. Additionally, each profile must allot exactly 100% of the AADT value over the course of the day. In this way, the demand profile effectively represents the density function for the daily allocation of AADT. This exactly mirrors the role a probability density function (PDF) provides for a continuous random variable. Thus, so long as the PDF function for a distribution is mapped to an interval corresponding to a 24-hour period, it fulfills all requirements to be used as a daily demand profile.

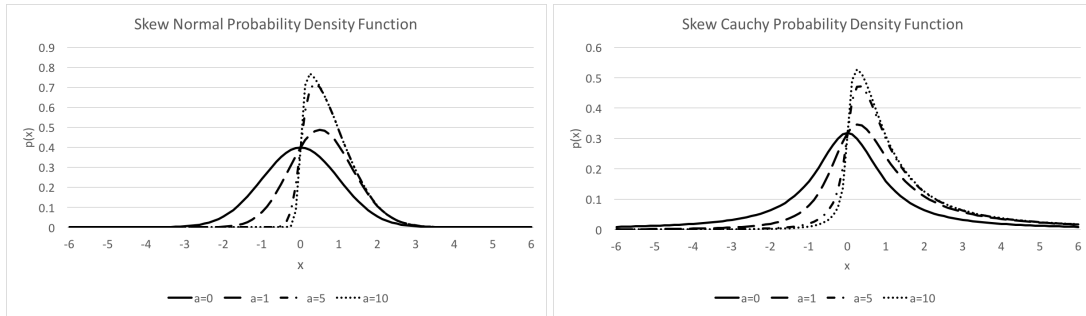
A brief survey of the PDFs for most common continuous random variable distributions reveals that none are described by a PDF that provides an accurate approximation of well-known daily traffic behavior such as that shown in Figure 3.2. This is unsurprising as it is unlikely that daily traveler behavior can be categorized by a single instance of a distribution. As an intuitive example, consider daily traffic for an average non-holiday day of the work week. Most vehicles on the road at any point of the day likely fall into one of three categories: vehicles heading to work in the morning, vehicles heading home in the evening, and random travelers throughout the day. This breakdown of travelers can then be represented as a mixture of three distinct distributions. One distribution represents the morning home-to-work travelers, the second represents the afternoon work-to-home travelers, and the third represents all other random daily travelers.

In general, a mixture distribution is defined as being the weighted combination of n independent probability distributions. Each distribution is assigned a weight w_i , with all weights summing to a value of 1. The probability density function of such a distribution is defined as the weighted sum of the PDFs of each individual distribution, as shown in Equation 3.1. For the purposes of this work, it is assumed that all random variables making up the mixture distribution have continuous PDFs, which implies that the PDF of Equation 3.1 is also continuous.

$$p(x) = \sum_{i=1}^n w_i p_i(x), \quad \text{where} \quad \sum_{i=1}^n w_i = 1 \quad (3.1)$$

A mixture distribution provides the representation with near unlimited flexibility in the shape of the resulting PDF and corresponding demand profile. In order to reduce the search space to a reasonable size as well as improve convergence, assumptions can be made about the number of distributions n as well as the type of each. In keeping with the intuitive example mentioned above, n is always assumed to take on a value of 3 for this research. Further, the type of the individual distributions is limited to either being a skew-normal or a skew-cauchy distribution. Figure 3.7

shows examples of the variety of shapes afforded by these two distributions. However, in the event that for a particular facility these assumptions do not provide acceptable results, both can be altered or relaxed back to their generalized state.



(a) The skew normal distribution with various values for the skewness parameter a .

(b) The skew Cauchy distribution with various values for the skewness parameter a .

Figure 3.7 Visual example of the skew normal and skew Cauchy distributions. Both distributions are generated for a mean μ of 0 and a standard deviation σ of 1.

With these assumptions, each of the three distributions has five parameters that become the decision variables in the GA search process. For random variable i of the mixture distribution, these five values are the type T_i , the mixture weight w_i , the mean μ_i , the standard deviation σ_i , and the skewness a_i . The mean μ_i serves as a location parameter that will anchor distributions around key time intervals such as an AM or PM peak in demand. The shape of each distribution will largely be defined by σ_i and a_i , which will control the steepness of drop-off and amount of probability density located on either side of the mean. The GA can then manipulate these values until they generate demand profile shapes resulting in predicted performance measures matching observed values. Table 3.1 summarizes these parameters and their functions.

Each profile can then be represented by a total of 15 real-valued parameters, five values for each of the three distributions. If each parameter is represented as a fixed length binary string, then the profile as a whole can be represented as one long string formed from a concatenation of the individual values. Since the type T_i of each distribution is limited to being one of two types, it can simply be represented by a single binary digit. The other four parameters of each distribution can make use of the simple projection method detailed in Section 3.5.1. All that is required is that a range of values be specified for each parameter, which can be discretized accordingly with the maximum integer value allowed by the binary encoding.

Table 3.1 Summary of the five parameters of each distribution representing the decision variables of the problem.

Parameter	Description
T_i	Type of the random variable i .
w_i	Weight of random variable i in the mixture distribution.
μ_i	Mean of random variable i .
σ_i	Standard deviation of random variable i .
a_i	Skewness of random variable i

Reasonable ranges for each of the remaining four parameters can easily be obtained. The weights w_i can be specified to sum to a value of 1, or can be allowed to vary more generally so long as the resulting distribution is normalized before it is used. Each location parameter μ_i must fall somewhere in the 24-hour period, and in many cases, such as for a workday, more specific ranges for AM or PM peak locations can be provided. Using the area covered by normal and Cauchy distributions, a 24-hour study period can be converted to a corresponding range of -6 to 6 for the domain of the mixture distribution PDF. Consequently, a simple linear function $y = 0.5x - 6$ can be used to project an hour value x into the range as y . For example, if an analyst knows that a morning peak occurs around 8am one of the location parameters μ_i could be limited to a range around a projected value of $0.5 \cdot 8 - 6 = -2$.

Since the overall behavior of both σ_i and a_i is predictable, reasonable ranges for both can be specified to place them within conservative limits. The standard deviation σ_i determines the magnitude of the peak for each distribution, thus can be limited such that the peak does not exceed a viable allocation of demand for the time interval. The skewness a_i can be centered around a value of 0 (corresponding to no skew), and again conservative upper and lower bounds are all that need be provided, which can easily be obtained by testing a few values. Figure 3.7 provides an example visual reference that limiting the range for a_i to be between -10 and 10 should be sufficient, although again this can be tailored to any specific application.

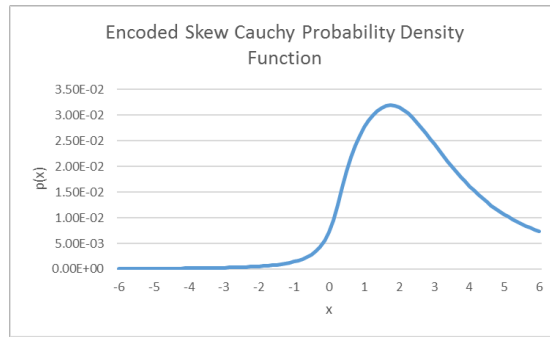
Figure 3.8 demonstrates the process of encoding the five decision parameters for a single random variable into a binary string. The table at the top of the figure outlines the ranges for the five parameters, and computes the step size based on the number of binary digits of the encoding. The bottom of the figure explicitly demonstrates the relationship between the binary representation and the values that shape the probability distribution function. Building on this process, Figure 3.9 shows how three random variables are combined to create a mixture distribution representing the demand profile for a study period. The values for the five parameters of each random probability

variable are given in the table at the top of the figure, and the individual component and combined probability distributions are shown in the bottom chart.

Parameter	Min Value	Max Value	# Binary Digits	Resulting Step Size
T_i	0	1	1	1
w_i	0	0.5	4	0.03
μ_i	-5	5	4	0.66
σ_i	0.5	5	4	0.3
a_i	-10	10	4	1.33

(a) Distribution parameter ranges.

0	→ $T_i = 0$ → Cauchy Distribution
1	
0	→ $w_i = 9$ → $0.05 + 9*0.03 = 0.32$
0	
1	
1	
0	→ $\mu_i = 11$ → $-5 + 11*0.66 = 1.6$
1	
1	
0	
1	→ $\sigma_i = 6$ → $0.5 + 6*0.3 = 2.3$
1	
0	
1	
1	→ $a_i = 12$ → $-10 + 12*1.33 = 5.96$
0	
0	

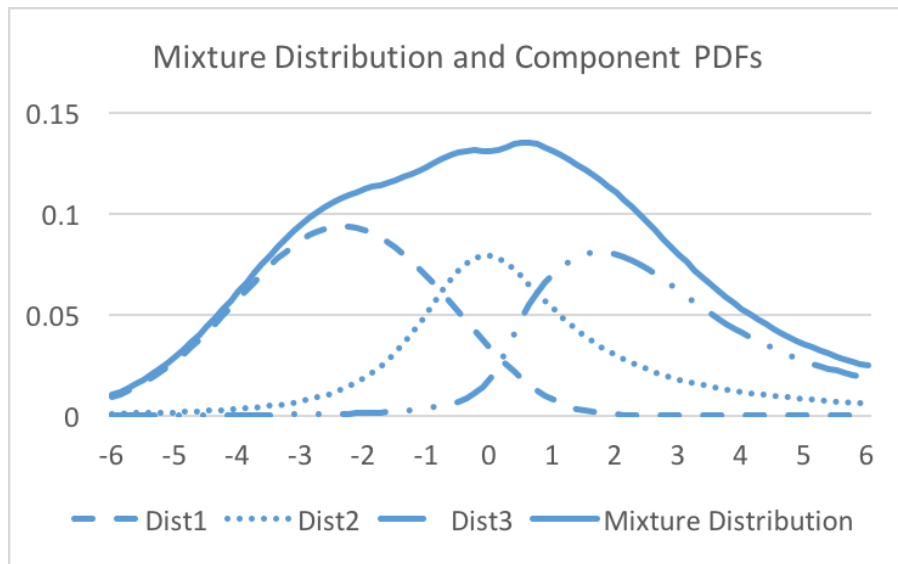


(b) String to random probability density function conversion.

Figure 3.8 Example demonstrating the encoding of an n -distribution candidate profile into a binary string.

Parameter	T_1	w_1	μ_1	σ_1	a_1
Value	Normal	0.2	-2.33	1.7	-2.02
Binary	1	0101	0100	0100	0110
Parameter	T_2	w_2	μ_2	σ_2	a_2
Value	Cauchy	0.32	-0.33	1.0	1.0
Binary	0	1001	0111	0011	1000
Parameter	T_3	w_3	μ_3	σ_3	a_3
Value	Cauchy	0.32	1.6	2.3	5.96
Binary	0	1001	1011	0110	1100

(a) Parameters for each of the three random variables of the example mixture distribution.



(b) Probability density functions of the full mixture distribution and the component random variables.

Figure 3.9 Full mixture distribution encoding example.

3.6 Fitness Evaluation and Evolutionary Operators

The following sections present a more detailed description of the strategies that make up the search process of the GA framework. First, a performance measure based fitness evaluation function is defined that is used to measure the quality of one candidate solution compared to another. Next, five evolutionary operators are presented along with details on how each strategy is implemented for the demand estimation problem.

3.6.1 Fitness Evaluation

The primary objectives of the demand estimation are presented above in Section 3.2. The overarching goal is to match the performance measures predicted by the methodology with known real-world observations. Therefore, a straightforward formulation of the objective function for the GA is as a minimization of the absolute difference in observed and predicted values for a given performance measure. Real-world performance measure data is often available in the form of observed speeds, denoted $v_{i,p}$ for a segment i ($i = 1 \dots N_S$) in analysis period p ($p = 1 \dots P$). Analogously, using the demand profiles of a candidate solution, the methodology predicts the speed for each segment i in analysis period p , denoted as $v'_{i,p}$. Thus the objective function can simply be formulated as a minimization of the absolute difference of observed speeds and predicted speeds as shown in Equation 3.2. Additional choices for the objective function could seek to minimize differences in travel times, vehicle flow volume served, or segment occupancy percentages.

$$\min \sum_{i=1}^{N_S} \sum_{p=1}^P |v_{i,p} - v'_{i,p}| \quad (3.2)$$

The choice of Equation 3.2 as the objective function provides the GA with a theoretically achievable global optimum of 0. However, in practice an exact value of 0 will likely not be achievable for a number of reasons. First, the encodings presented in Section 3.5 create a discrete search space, which precludes the GA from being able to consider certain demand volume values. Second, all real-world observations the GA is striving to match will have at least some degree of uncertainty. There will also be a disparity between the number of significant digits of the observed and computed values. Lastly, any computer implementation of the HCM methodology entails a large number of floating point calculations, and round-off errors prohibit any GA from progressing past some tolerance ϵ around 0. However, for the intended purpose and applications, an inability to achieve the exact global optimum is not detrimental so long as the optimality gap falls within one of the above mentioned bounds for error.

It is also important to note that this choice of objective function does not guarantee uniqueness for globally optimal solutions. This non-uniqueness is particularly likely for facilities with large numbers of ramps, where slight shifts in demand volumes among adjacent or nearby ramps will not affect the predicted performance measures within the afforded tolerance for accuracy. In addition, some equations of the HCM methodology are not one-to-one mappings from demand volumes to certain performance measures. For example, for all demand volumes that create a demand-to-capacity ratio below a certain value, the predicted speed will always be the free flow speed. However, this is largely restricted to the undersaturated methodology, and especially to analysis periods falling in the early morning or late at night. In order to truly calibrate these types of analysis periods, information about real world flow volumes would be needed, but as demand is far below capacity, there is no need for demand calibration as the observed flows represent the true demand.

Lastly, it is possible that a given set of demands generated by the GA will be infeasible for the methodology. This can occur when implicit constraints of the methodology are violated. For example, in some situations with poor input data or large amounts of uncertainty in AADT values, the specified exiting demand could exceed the entering demand at some point in the study period. In the event of this occurring, a simple “death penalty” (see Coello (2002)) is enforced for the organism representing the infeasible candidate solution. This entails assigning a very large fitness value to the organism to prevent it from being chosen by the selection operator.

3.6.1.1 Computational Burden of the HCM Methodology

Evaluating the fitness of an organism requires first analyzing the facility using the HCM methodology to compute the necessary performance measure outputs. Profiling results indicate that approximately 85% of the computation for a single GA run consists of executing the HCM analysis through the FREEVAL computational engine. Comparatively, the evolutionary operators described in subsequent sections consist mostly of simple operations on binary strings and random number generation. While generating random numbers can be considered costly in comparison to simple operations, the complexity added for this application is far outweighed by the burden of the fitness computation. Further, the complexity of the methodology is almost solely dependent on the number of segments and time periods being modeled. Geometric specifics of a facility have little impact on the computations of the methodology, as essentially the same set of equations must be run for each segment in each time period regardless of segment length or type. Hence the run time of the calibration process is almost entirely dependent on the number of segments and time periods.

3.6.2 Selection and Crossover

The primary drivers of the evolutionary process of a GA are the selection and crossover operators. The selection operator is used to choose the two organisms that will undergo crossover in order to produce new offspring organisms. There are a number of ways to implement the selection process including random selection, roulette wheel selection, and tournament selection. This research makes use of a simple k -tournament selection strategy in which k solutions are randomly picked from the current generation, and the one with the best (lowest) fitness is selected to be a parent. This process is repeated twice to determine the two parent organisms to undergo crossover. The value of k can be varied to improve the convergence rate of the GA.

Most crossover strategies were initially developed in order to operate on two individual binary strings with the goal of exchanging binary digits between each. However, the encoding approaches presented in Section 3.5 both model candidate solutions as an array of binary strings, with each individual string representing a single profile. Consequently some care must be taken to ensure the intended crossover behavior is not skewed when dealing with arrays, which occasionally necessitates some slight modifications to the operators. An additional concern is that mixing information corresponding to differing entry or exit points along a facility is counterintuitive, and as such any strategy used should refrain from exchanging information between non-corresponding profiles.

In the interest of simplicity, a simple uniform crossover strategy is implemented for the GA framework. The nature of the uniform operator circumvents both of the aforementioned issues because the strategy only considers individual binary digits. More specifically, when performing crossover between two encoded solutions, the strategy iterates sequentially through each binary digit of each profile and performs a random test to determine if the corresponding digits should be exchanged between the two encodings. The random test consists of generating a random number uniformly between 0 and 1 for each digit. If the random value is less than a pre-specified mixing percentage, the digits are then swapped. Generally a 50% mixing rate is used for the test. Figure 3.10 demonstrates the process of the uniform crossover operator.

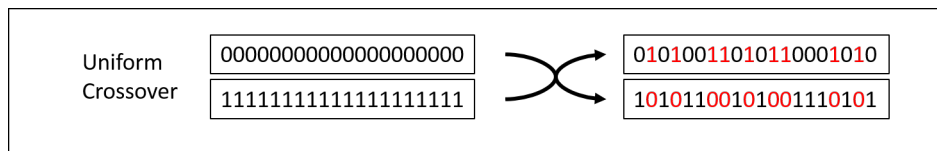


Figure 3.10 Visualization of the three crossover strategies.

3.6.3 Mutation

A simple uniform mutation strategy is implemented for use in the demand estimation GA framework. As its name would suggest, a uniform mutation strategy is very similar to the previously described uniform crossover strategy. However, unlike crossover, mutation strategies operate just on the genetic encoding of a single organism. This randomly introduces changes into the population in order to help maintain diverse pool of candidate solutions and stave off premature convergence.

Just as with uniform crossover, uniform mutation considers each binary digit of an encoding individually. The strategy again sequentially iterates through each digit of each profile and generates a random number uniformly between 0 and 1. However, instead of a mixing percentage hovering around 50%, the value is tested against a mutation rate that is generally much lower. If the random is less than the specified mutation rate, the corresponding binary digit is flipped to its binary alternative. Specifically, a 0 becomes a 1, and a 1 becomes a 0. The choice of mutation rate varies for each specific problem, but research generally shows that for binary GA encodings, a rate of 2-5% has the most beneficial effect on convergence and final solution quality (Mitchell, 1998).

3.6.4 Elitism and Preservation

Two additional evolutionary operators are implemented in the GA framework. Unlike the crossover and mutation strategies described above, these two operators do not interact with the encoding of a candidate solution, but rather operate at the level of the entire population.

First, an elitism strategy is implemented that selects the best individual or individuals of a given generation and automatically replicates them into the subsequent generation, skipping both the crossover and mutation phases for those organisms. Elitism often proves helpful because crossover and mutation strategies are not always guaranteed to be beneficial to an organism's fitness. An elitism strategy thus serves to ensure that at least some of the best building blocks of the current generation will survive unaltered into the next generation. Most testing for this framework found that specifying the top 5% of the current generation be preserved through elitism provided the most benefit for convergence.

A preservation operator serves to benefit the population almost through a completely opposite approach as elitism. For this framework, a simple preservation strategy is implemented that selects a number of random organisms from a generation and automatically replicates them into the subsequent generation. As its name suggests, the operator helps to preserve diversity in the population that could be hiding yet untapped high quality building blocks. More importantly though, if only the best organisms move on, it is likely a population will lose its genetic diversity, which can cause the GA to converge prematurely to a local optimum. Inclusion of a preservation strategy provides

an additional way for maintaining population diversity alongside a mutation operator. A random preservation rate of 5% of a generation's organisms is used for the demand estimation GA framework.

3.7 Examples and Computational Results

Sections 3.5 and 3.6 presented each of the required pieces of the demand estimation GA. With this complete, the encodings and evolutionary operators are implemented in the Java programming language as a new GA framework. Java is chosen largely because the upcoming 6th edition of the HCM includes a Java implementation of the freeway facilities method in the form of the FREEVAL computational engine. The GA framework is designed to interact with this engine, and uses it to perform the the analysis of the methodology in order to evaluate each organism's fitness.

The following sections present three computational experiments that show the value of the proposed demand estimation and calibration framework. Section 3.7.1 demonstrates that the n -distribution encoding can be used to accurately recreate well-known demand profile shapes. Subsequently, two case studies are presented in which the set of demand volumes for a facility are generated through adjustment or estimation in order to better match a set of target facility performance measures. First, a simple example is presented in which the ground truth demand is known, and the performance of each encoding under varying conditions is analyzed. The second case study is conducted on a real world stretch of freeway outside of Raleigh, North Carolina. Only imperfect information is known regarding both the AADT and the shape of the underlying demand profile for this facility, and the effectiveness of the two encodings in matching the real-world performance measure data is compared.

3.7.1 Matching Hourly Demand Profiles

There have been a number of studies conducted in order to identify common shapes of daily demand profiles for both urban and rural freeways. Some studies conducted were aimed at creating national default profiles, while others were aimed at state and regional levels. Before moving on to the computational examples involving the HCM methodology, this section presents a quick case study that shows the ability of the n -distribution approach to match the shapes of these published demand profile defaults. While the default profiles are provided at the hourly level, the freeway facilities requires volumes for each 15-minute analysis period. As such, a simple linear interpolation was applied to project the hourly profile into the corresponding 15-minute bins. This increased the number of data points by a factor of four from 24 to 96.

The GA framework of Section 3.6 was used to test how well the n -distribution approach could

match each of these known demand profile shapes. This was accomplished by modifying the fitness function to measure the absolute error between the profile predicted by each mixture distribution and the true shape. In order to demonstrate the flexibility of the encoding, as few assumptions as possible were made about the five parameters that describe each distribution, and large ranges for w_i , μ_i , σ_i , and a_i were used. The ranges for each value for the experiment are given in Table 3.2.

Table 3.2 Parameter ranges for the mixture distribution encoding.

Parameter	Min Value	Max Value	# Digits
T_i	0	1	1
w_i	0	0.5	12
μ_i	-5	5	12
σ_i	0.5	5	12
a_i	-10	10	12

Table 3.3 provides the results of the computational experiment. The first two columns identify the profile shape the GA is trying to match. The third column gives the fitness of the best organism found by the GA run. Fitness is defined as the sum of the absolute error at each point along the profile, and represents the accumulated error over the course of the 24-hour period. For 16 of the 22 test cases, the GA is able to identify a profile shape with less than 6% total error, while the maximum accumulated error is just 7.15%. The next column reports the root mean square error between the computed and the true profiles. For all but one of the profiles, the RMSE of the computed profile is less than 0.1%.

These results reflect that the profiles generated using the n -distribution encoding provide an accurate representation of the underlying default demand shapes. For example, Figure 3.11 visually shows that the approach is especially accurate in matching the unimodal and weekend profile shapes. While the shapes are not perfectly matched, discrepancies are largely due to the fact that the default profile shapes have jagged edges, while the generated profiles are inherently much smoother. Real-world profiles are likely to be somewhat jagged, and to overcome this difference in shape, an additional local search within a given confidence bandwidth could be employed to increase the accuracy. Figure 3.12 shows a relaxation to a 10% confidence bandwidth around the computed profiles for the Bimodal PM profile shapes. It is clear from this that the relaxation provides a more complete covering of the true shape. Further, the profile-based encoding introduced in Section 3.5.1 provides a direct way of searching this bandwidth and could be included as a second stage local refinement method.

Table 3.3 Results for each profile.

Main Type	Sub Type	Fitness	RMSE
Unimodal	Minimum	3.21	0.051
	25th Percentile	3.69	0.057
	Average	4.25	0.061
	75th Percentile	3.18	0.051
	Maximum	4.49	0.066
	Median	4.57	0.063
Bimodal AM	Minimum	6.24	0.089
	25th Percentile	5.36	0.078
	Average	4.23	0.058
	75th Percentile	6.59	0.085
	Maximum	5.23	0.074
	Median	4.13	0.057
Bimodal PM	Minimum	6.79	0.088
	25th Percentile	6.97	0.097
	Average	5.90	0.076
	75th Percentile	7.15	0.101
	Maximum	5.83	0.079
	Median	4.45	0.063
National Weekday	Urban	5.96	0.085
	Rural	5.67	0.078
National Weekend	Urban	5.34	0.088
	Rural	6.72	0.096

Another observation standing out in Figures 3.11 and 3.12 is that much of the error occurs at the tails of the profiles. These tails correspond to demands at times far removed from the peak hours, and are therefore largely unimportant in a freeway facilities analysis. As mentioned in Section 3.6.1, many performance measures are the same for all periods with sufficiently low demands. This is especially noteworthy since large portions of the tails allocate demands to 15-minute analysis periods that are 0.5% or less of the total AADT. For what is considered a large unidirectional AADT value of 250,000, this is merely 125 vehicles. By comparison, the HCM recommended capacity for the minimum allowed two-lane cross section of a freeway during a 15-minute time frame is 1200 vehicles.³ This in turn corresponds to a demand-to-capacity (D/C) ratio of just 0.1, indicating the methodology would be in the highly undersaturated regime. As such, for situations requiring the *n*-distribution encoding for demand estimation, it is highly likely that a large amount of the error in

³The HCM recommends a default capacity of 2400 vehicles per hour per lane, which corresponds to a 600 vehicles per lane for a 15-minute analysis period.

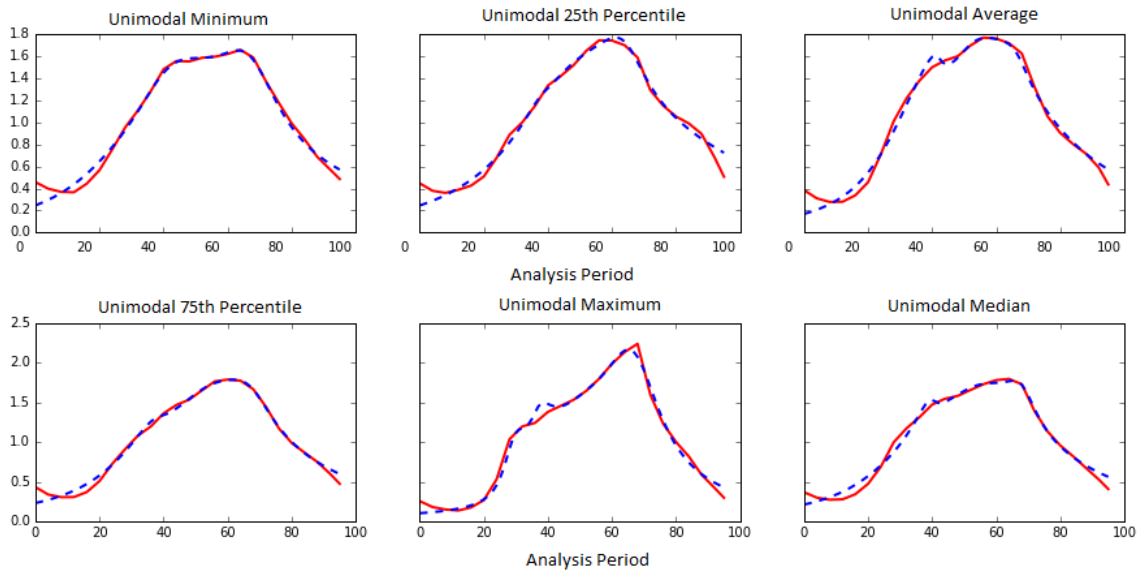


Figure 3.11 Plotted results for matching the six unimodal subtypes. The true profile shape is represented by the red line, and the computed profile is shown as the blue dashed line.

the profile shape will either be undetectable when considering most performance measures or be inconsequential to the analysis as a whole.

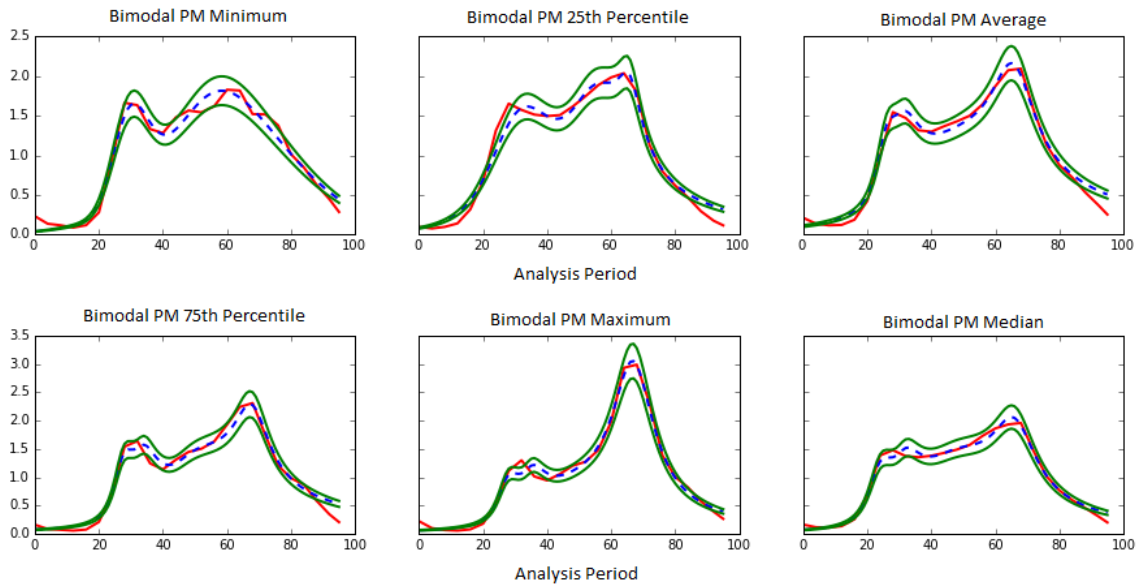


Figure 3.12 Bimodal with PM peak known profiles (NCDOT, 2016). The red line represents the true profile, the blue dashed line shows the computed profile, and the green lines provide a 10% relaxation bandwidth of the profile.

3.7.2 Simple Case Study

The previous section decouples the proposed GA framework from the HCM methodology in order to demonstrate the effectiveness of the n -distribution approach for generating realistic profile shapes. This section reintroduces the methodology into the GA framework and tests its ability to calibrate the methodology for a simple example where the underlying demand is known. This assumption of known “ground truth” provides an opportunity to analyze how the framework performs in a best case scenario, instead of jumping directly to a real-world case study where a lack of ground truth can make it difficult to draw certain conclusions about its effectiveness. It is assumed that values for free flow speed and capacity are known and utilized for the example. Both the profile-based and the n -distribution encodings are tested and compared based on their ability to match predicted facility performance measures to a set of target data.

For this case study, a basic facility was designed to be analyzed over a 24-hour study period. We assume that values for all inputs other than demands are known with 100% confidence. An AADT value is chosen, and demand volume inputs for the facility are allocated according to a known profile shape. An HCM freeway facilities analysis is then conducted on the facility, and the predicted

speeds are recorded to serve as a ground truth data set. This allows for analysis to be conducted where the target performance measures are guaranteed to be achievable using the methodology. Further, the demand profile estimated by the GA framework can be compared to its underlying ground truth - a feat which is generally impossible when analyzing real-world facilities.

The geometry of the test facility is shown in Figure 3.13. The facility consists of 27 half-mile segments, with a lane-drop in the 26th segment designed to create a flow bottleneck. The facility has a free flow speed of 65 mph, and the HCM default values for capacity are used for each segment. The facility is assigned an unidirectional AADT value of 69,000, which is allocated to the study period according to the 25th percentile bimodal AM peak demand profile shown in Figure 3.14. Figure 3.15 shows the resulting speed contour obtained from the HCM freeway facilities analysis. The contour clearly shows two congestion areas, with the larger of the two occurring during the morning peak hours. This set of speeds serves as the target data set the the GA framework will be tasked with matching.

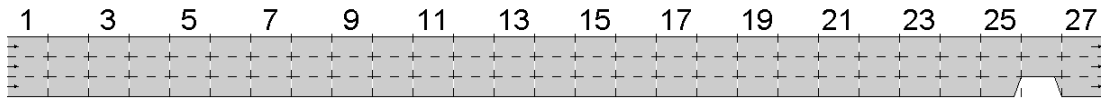


Figure 3.13 Geometry of the simple example facility. The lane drop in segment 26 provides a bottleneck for the facility that creates two congested regimes.

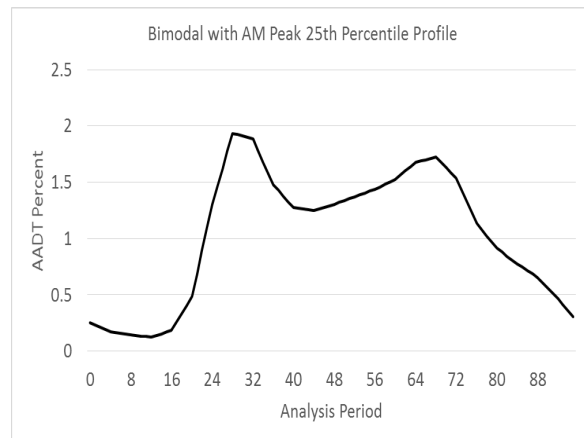


Figure 3.14 Bimodal with AM peak 25th percentile demand profile (NCDOT, 2016).

An interesting note about this example is that while the geometry of the facility can be considered to be very simple, the size of the analysis actually pushes the upper-bounds allowed by the HCM methodology. The length of the facility being studied is 13.5 miles, and the number of segments is at least around the average number encountered for most analyses. Additionally, the example is modeled over the maximum allowed study period duration. In other words, though technically a basic example, this case study actually represents one of the larger size analyses that would be calibrated using the GA framework. Two scenarios are developed in order to test the performance

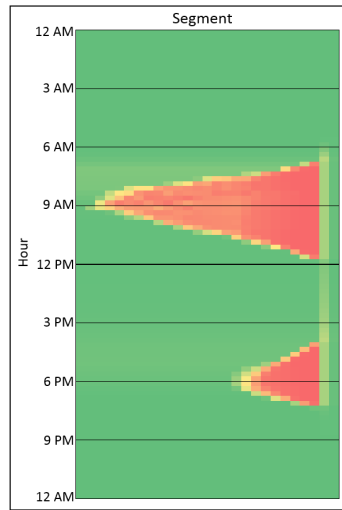


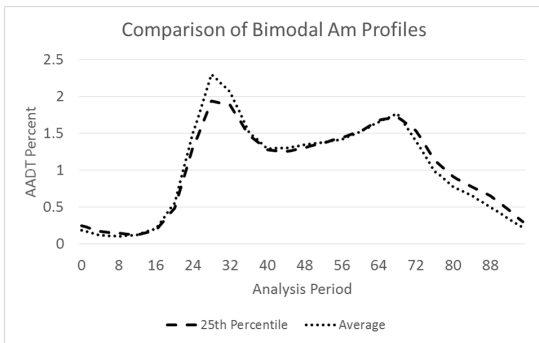
Figure 3.15 Speed contour predicted by the HCM analysis for the contrived ground truth demand allocation.

of both encoding approaches. First, the profile-based encoding is applied to calibrate the analysis when close but incorrect values for both AADT and the demand profile shape are used as an initial guess. The second approach uses a 3-distribution encoding to estimate the profile when only an AADT value and knowledge for two location parameters is provided based on the temporal vicinity in which the congestion occurs in the target data set.

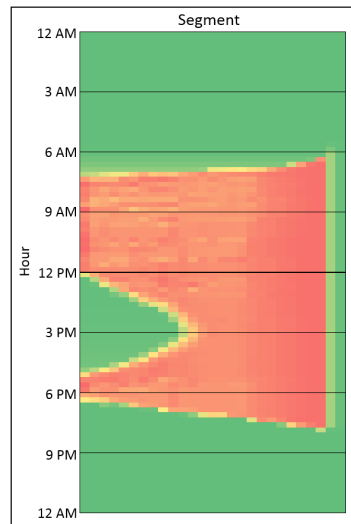
3.7.2.1 Profile-Based Encoding Results

The first scenario assumes that the analyst starts with the average bimodal AM peak demand profile (NCDOT, 2016). A comparison of the 25th percentile and average profile is shown in Figure 3.16a. Additionally, a unidirectional AADT value of 80,000 is initially used to allocate demands volumes

to the 15-minute analysis period intervals. The resulting speed contour predicted by the analysis is shown in Figure 3.16b. This contour reveals a much larger amount of congestion compared to that of the target speeds of Figure 3.15, including problematic spillover beyond the boundary of the analysis. A set of metrics is given in Table 3.5 concerning the “fitness” of this initial starting point. These show that on average there is nearly a 15 mph difference between the uncalibrated and target speeds, with some uncalibrated speeds being more than 55 mph off. These major errors between the predicted and target speeds indicate that the basic initial demand estimation used is insufficient for calibrating the facility.



(a) Comparison of the 25th percentile and average profiles for the Bimodal AM peak profile (NCDOT, 2016).



(b) Speed contour predicted by the HCM analysis for the initial guess demand allocation.

Figure 3.16 Demand profile comparison and speed contour of the initial guess demand volume inputs.

The profile-based encoding approach is used to calibrate the demand profile of the facility under this scenario. A confidence interval of 15% is specified to create the search space for the algorithm. A summary of the additional parameters for the experiment and the GA setup is given in Table 3.4, and the results of five GA runs are shown in Table 3.5. These results show that the GA was very successful in calibrating the facility. On average, the overall error in speeds is reduced by 97%, which translates to an average error of just 0.4 mph per segment per time period. While some large individual errors still exist, an average 90th percentile error of 0.78 mph shows the GA is able to closely match the speeds for the vast majority of the segments and time periods. Figure 3.17

Table 3.4 Summary of the setup parameters for the profile-based encoding GA runs.

Experiment Parameter	Value
Number of Runs	5
Profile Confidence Interval	15%
Population Size	200
Max Iterations	400
Binary Encoding Digits	16
Crossover Mixing %	50%
Mutation Rate (%)	2%

provides a contour showing the distribution of speed error across the facility and study period. Each cell of the contour represents the absolute speed difference for a segment in the corresponding analysis period, with white cells indicating very small errors in speed, and red cells indicating larger errors in speed.

Table 3.5 Summary of the calibration results for the profile-based encoding GA runs.

Metric	Uncalibrated	GA Average	GA Best	GA Worst
Total Speed Difference (mph)	38,647.31	1040.41	963.98	1138.97
Average Speed Difference (mph)	14.91	0.40	0.37	0.44
90th Percentile Speed Difference (mph)	49.96	0.78	0.64	0.93
Max Speed Difference (mph)	57.04	26.05	17.23	43.33
GA Run Time (Minutes)	-	22	22	22

Since this example is designed with the explicit goal of being able to compare estimated demands with known ground truth demands, a deeper look can be taken at the underlying profiles generated by the GA framework. However, unlike most other result metrics, multiple demand profiles cannot be averaged into a single profile to gauge the effectiveness. As such, both profiles corresponding to the best and the worst of the GA runs are shown in Figure 3.18. Each profile plot shows the initial profile guess, the actual underlying profile, and the profile estimated by the GA. While there is some noise within the generated profiles, it is clear that the GA consistently estimated a shape much closer to that of the true profile. The improvement is especially apparent during the AM peak where there is a near perfect reduction from the base profile to the true shape.

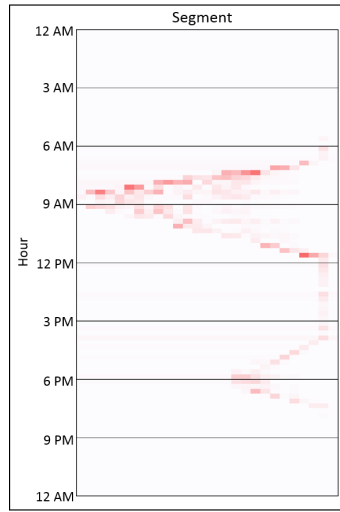


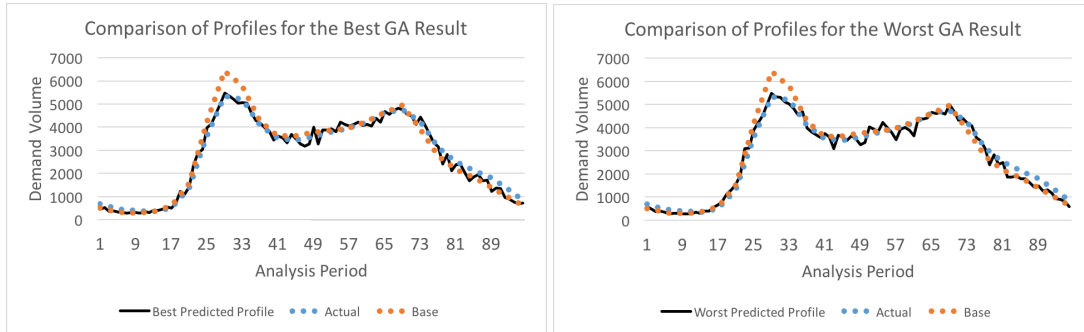
Figure 3.17 Contour highlighting the speed differences between the GA run with the lowest total speed error and the target data set. White cells represent minimal speed errors, while red cells indicate larger speed errors.

3.7.2.2 3-Distribution Encoding Approach

The second test scenario replicates a situation in which an analyst has little information regarding a facility's demand. Specifically, this scenario assumes that the only information the analyst possesses is an accurate AADT value and the target speed contour. The n -distribution encoding will be utilized in order to estimate a demand profile from the ground up.

Additional useful information can be extracted from the target data set and used to help guide the search process. Two separate congestion regions clearly arise during the 24-hour study period as seen in Figure 3.15. These congested time periods can be used to place restrictions on the search area for two of the random variables of the mixture distribution of the encoding. The congestion of the AM region is contained entirely between hours of 6:30am and 12:00pm, and the congestion in the PM region is contained entirely between 4:00pm and 7:30. Taking advantage of this knowledge, for each of the two regions, the location parameter of one of the random variables of the mixture distribution is confined to be within that area.

Table 3.6a presents the experiment parameters used to set up the GA and execute the demand estimation runs using a 3-distribution encoding. A total of 5 runs are executed with a population size of 400 and an iteration limit of 100 generations. A 16 digit binary encoding is used for each parameter of the mixture distribution (excluding random variable type, which only requires a single digit). The parameters for the mixture distributions are shown in Table 3.6b. For all parameters other



(a) Profile comparison for the best calibration found by the GA. (b) Profile comparison for the worst calibration found by the GA.

Figure 3.18 Comparison of profiles generated by the best and worst calibration of the GA framework. The true profile is represented by the blue dotted line, and the initial incorrect profile is shown as an orange dotted line. The black line represents the profile as calibrated by the GA framework.

than the location μ_i , a single range is used as the search space for each of the three random variables. The knowledge gained through examining the target contour is incorporated in the experiment by specifying individual ranges for each of the three location parameters. One is confined to the AM region, a second is confined to the PM region, and the third is specified to fall somewhere in the range of the two.

Table 3.6 Summary of the setup parameters for the 3-distribution encoding GA runs.

(a) Genetic Algorithm Parameters

Experiment Parameter	Value
Number of Runs	5
Population Size	400
Max Iterations	200
Binary Encoding Digits	16
Crossover Mixing %	50%
Mutation Rate (%)	2%

(b) Mixture Distribution Parameters

Distribution Parameter	Symbol	Range
Weight	w_i	0.05, 0.5
Mean 1	μ_1	6:30am, 12:00pm
Mean 2	μ_2	4:00pm, 7:30pm
Mean 3	μ_3	12:00am, 11:59pm
Std. Dev.	σ_i	0.5, 5.0
Skewness	a_i	-10, 10

A summary of the performance of the GA runs is shown in Table 3.7. Though it is not quite as accurate as the profile-based approach, the 3-distribution encoding approach is still able to match the target speed contours with a high degree of accuracy. The average speed difference for each segment and time period across all of the GA runs is just 0.94 mph, and the average 90th

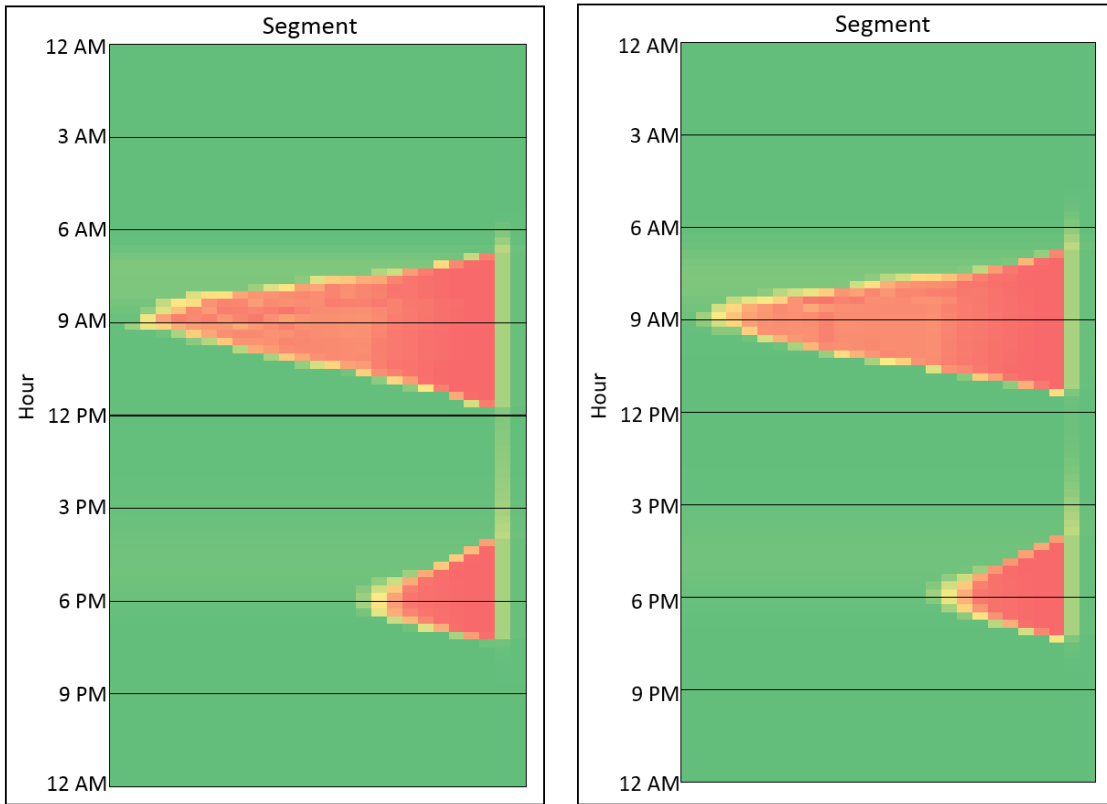
percentile error is only 1.12 mph. Further, the two areas of congestion visible in the target data also appear in the speeds predicted from each of the demand estimation attempts. Figure 3.19 provides a side-by-side comparison of the speed contours for the target data set and the best calibration attempt. While the shapes do not match perfectly, the overall congestion trend is matched well.

Table 3.7 Summary of the demand estimation results for the 3-distribution encoding GA runs.

Metric	GA Average	GA Best	GA Worst
Total Speed Difference (mph)	1857.16	1530.39	2252.03
Average Speed Difference (mph)	0.71	0.59	0.87
90th Percentile Speed Difference (mph)	0.79	0.39	1.10
Max Speed Difference (mph)	50.96	45.78	55.00
GA Run Time (Minutes)	11	10.56	11.78

Figure 3.20 shows a comparison between a selection of the demand profiles generated by the 3-distribution encoding and the true shape. Both the best and the worst calibration runs generated a profile that is largely the same as the true shape. However, as expected, the profiles do differ noticeably in the tails corresponding to the early morning and late night. An examination of the speed contours reflects that this is largely inconsequential and likely unavoidable for applications calibrating just on speeds. Though the demand volumes are significantly higher in the tails, the average total error accumulated for the speeds predicted before 5am and after 8pm for the 5 runs was just 3.7 mph, which corresponds to an average error of less than 0.005 mph for each segment and analysis period.

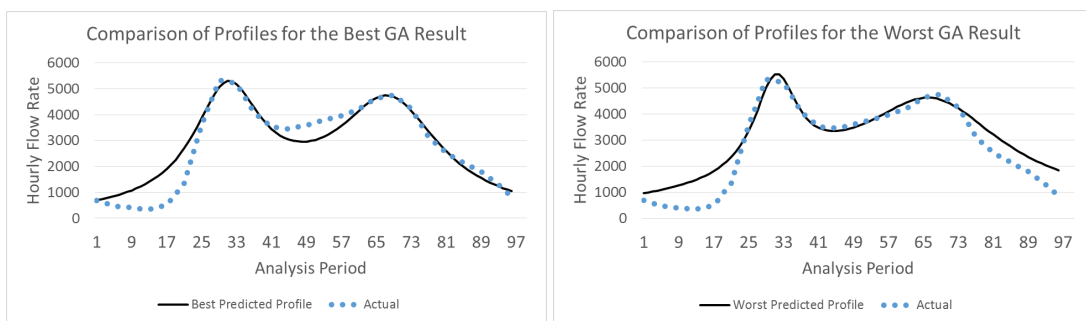
The accuracy of the predicted speed contour and the generated demand profile show that the *n*-distribution encoding is an effective calibration approach even when an analyst has little available information to work with. The demand estimation of this section was all accomplished just by using a single AADT value and basic information regarding the temporal location of congestion in the study period. While the approach is not guaranteed to perfectly estimate demands, this example shows it has the ability to identify and replicate the underlying behavior modeled using the HCM freeway facilities methodology.



(a) Contour representing the “ground truth” target speeds.

(b) Contour resulting from the best demand profile generated by the best GA run.

Figure 3.19 Side by side comparison of the target speeds (left) and the speeds predicted by the best GA run (right).



(a) Profile comparison for the best calibration found by the GA.

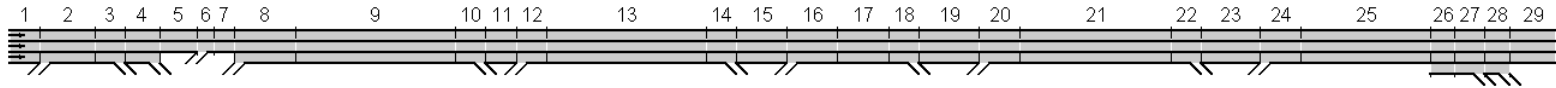
(b) Profile comparison for the worst calibration found by the GA.

Figure 3.20 Comparison of profiles generated by the best and worst calibration of the GA framework. The true profile is represented by the blue dotted line and the black line represents the profile as estimated by the GA framework.

3.7.3 I-540 Case Study

While constructed examples are useful for showing the theoretical merits of the framework, it is equally important to show the approach works for practical applications. In order to accomplish this, a case study for testing both approaches with real world data is developed for analysis based on a stretch of I-540 in Raleigh, NC. The facility consists of 14.5 miles of urban freeway, and contains seven on-ramps and eight off-ramps, yielding a total of 16 (including the mainline) demand profiles needing to be estimated or calibrated. Figure 3.21a shows the geometry of the facility as modeled using the HCM methodology. The facility is broken up into 29 segments, most of which have three lanes, and the facility free flow speed is thought to be 75 mph. A set of representative speeds for a five-and-a-half-hour study period between 6:00am and 11:30am on Tuesday, August 19th 2014, is collected from RITIS.org (Maryland CATT Lab, 2008) and are shown in Figure 3.21b. A set of AADT values for the mainline and ramps are collected from information available from the North Carolina Department of Transportation (2016).

The target speed data clearly shows that speeds are low around the AM peak hours. As such, an initial set of demands for the facility is assigned to the study period according to the average bimodal with AM peak demand profile for the mainline entry and all ramps. Figure 3.22 shows the speeds predicted by the methodology for these inputs, which serve as the initial best guess. Unfortunately, the contour shows a congestion regime that is very different from the real-world observations. There is a total speed error of 3,901 mph between the uncalibrated initial contour and the target data, which translates to an average difference of 6.11 mph per segment and time period. Consequently, it is necessary to calibrate the model before it can be deployed for planning or operational applications.



(a) Geometry of the I-540 WB case study facility.

Analysis Period	Segment																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
#1 6:00 - 6:15	65.4	72.6	72.6	72.6	73.0	73.0	73.0	73.0	73.0	73.0	73.4	74.1	74.1	74.1	73.6	73.7	73.7	73.7	72.4	70.5	70.5	70.5	71.8	71.1	71.1	71.1	71.1	72.8	72.8
#2 6:15 - 6:30	68.9	70.4	70.4	70.4	72.5	72.5	72.5	72.1	72.1	72.1	73.3	73.9	73.9	73.9	74.0	74.3	74.3	74.3	74.1	73.4	73.4	73.4	74.5	74.5	74.5	74.5	74.5	72.7	72.7
#3 6:30 - 6:45	73.2	70.8	70.8	70.8	71.1	71.1	71.1	71.5	71.5	71.5	73.1	72.5	72.5	72.5	73.5	73.1	73.1	73.1	72.5	72.0	72.0	72.0	71.7	73.7	73.7	73.7	73.7	72.9	72.9
#4 6:45 - 7:00	72.5	72.6	72.6	72.6	71.3	71.3	71.3	71.6	71.6	71.6	70.8	71.0	71.0	71.0	72.4	71.7	71.7	71.7	71.7	70.8	70.8	70.8	72.0	72.9	72.9	72.9	72.9	73.5	73.5
#5 7:00 - 7:15	73.7	70.3	70.3	70.3	70.7	70.7	70.7	71.9	71.9	71.9	73.3	71.9	71.9	71.9	72.5	72.5	72.5	72.5	72.5	70.8	70.8	70.8	71.7	72.1	72.1	72.1	72.1	73.7	73.7
#6 7:15 - 7:30	68.6	67.6	67.6	67.6	68.4	68.4	68.4	68.4	68.4	68.4	67.4	67.3	67.3	67.3	69.4	68.3	68.3	68.3	69.8	69.1	69.1	69.1	71.7	72.1	72.1	72.1	72.1	71.7	71.7
#7 7:30 - 7:45	67.9	64.7	64.7	64.7	68.6	68.6	68.6	60.9	60.9	60.9	46.5	48.8	48.8	48.8	63.9	62.8	62.8	62.8	59.0	59.1	59.1	66.4	70.1	70.1	70.1	70.1	70.9	70.9	
#8 7:45 - 8:00	71.3	68.7	68.7	68.7	67.8	67.8	67.8	53.9	53.9	53.9	31.5	38.7	38.7	38.7	45.3	46.4	46.4	46.4	36.0	53.9	53.9	53.9	64.8	68.5	68.5	68.5	68.5	68.5	68.5
#9 8:00 - 8:15	71.3	68.1	68.1	68.1	67.1	67.1	67.1	56.1	56.1	56.1	36.8	41.0	41.0	41.0	36.9	43.9	43.9	43.9	44.2	55.1	55.1	55.1	62.3	67.6	67.6	67.6	67.6	70.2	70.2
#10 8:15 - 8:30	72.7	68.7	68.7	68.7	67.5	67.5	67.5	67.3	67.3	67.3	55.9	43.0	43.0	43.0	35.8	43.8	43.8	43.8	42.2	56.9	56.9	56.9	64.6	69.1	69.1	69.1	69.1	71.1	71.1
#11 8:30 - 8:45	70.3	70.8	70.8	70.8	73.0	73.0	73.0	70.5	70.5	70.5	64.3	59.0	59.0	59.0	57.7	54.9	54.9	54.9	46.7	57.0	57.0	57.0	63.8	69.5	69.5	69.5	69.5	70.2	70.2
#12 8:45 - 9:00	71.8	70.9	70.9	70.9	71.1	71.1	71.1	70.1	70.1	70.1	70.3	70.0	70.0	70.0	63.4	54.9	54.9	54.9	57.8	62.4	62.4	62.4	65.7	68.1	68.1	68.1	68.1	70.0	70.0
#13 9:00 - 9:15	73.3	70.9	70.9	70.9	69.7	69.7	69.7	70.3	70.3	70.3	72.9	72.9	72.9	72.9	72.6	73.0	73.0	73.0	73.1	71.2	71.2	71.2	72.5	72.2	72.2	72.2	72.2	73.2	73.2
#14 9:15 - 9:30	73.8	69.8	69.8	69.8	71.3	71.3	71.3	73.6	73.6	73.6	73.3	72.6	72.6	72.6	72.5	73.0	73.0	73.0	72.6	72.3	72.3	72.3	74.4	73.3	73.3	73.3	73.3	73.9	73.9
#15 9:30 - 9:45	70.1	68.5	68.5	68.5	68.3	68.3	68.3	69.8	69.8	69.8	71.4	72.7	72.7	72.7	73.1	73.4	73.4	73.4	73.9	71.7	71.7	71.7	72.1	72.2	72.2	72.2	72.2	73.0	73.0
#16 9:45 - 10:00	67.2	68.7	68.7	68.7	68.2	68.2	68.2	68.1	68.1	68.1	68.9	70.1	70.1	70.1	71.0	71.1	71.1	71.1	71.1	70.5	70.5	70.5	71.9	71.3	71.3	71.3	71.3	72.9	72.9
#17 10:00 - 10:15	69.8	69.1	69.1	69.1	67.4	67.4	67.4	67.7	67.7	67.7	68.1	69.7	69.7	69.7	69.0	70.7	70.7	70.7	71.4	70.7	70.7	70.7	72.7	72.3	72.3	72.3	72.3	72.6	72.6
#18 10:15 - 10:30	72.5	66.9	66.9	66.9	69.1	69.1	69.1	72.7	72.7	72.7	74.4	74.0	74.0	74.0	72.9	72.1	72.1	72.1	72.3	71.1	71.1	71.1	73.5	74.2	74.2	74.2	74.2	73.9	73.9
#19 10:30 - 10:45	69.9	69.8	69.8	69.8	70.9	70.9	70.9	70.5	70.5	70.5	70.7	71.4	71.4	71.4	71.1	71.6	71.6	71.6	73.7	73.3	73.3	73.3	73.3	73.3	73.6	73.6	73.6	73.5	73.5
#20 10:45 - 11:00	69.0	67.0	67.0	67.0	68.0	68.0	68.0	68.3	68.3	68.3	69.4	69.7	69.7	69.7	69.6	70.5	70.5	70.5	71.4	70.8	70.8	70.8	71.5	71.8	71.8	71.8	71.8	73.1	73.1
#21 11:00 - 11:15	68.7	70.4	70.4	70.4	67.8	67.8	67.8	68.0	68.0	68.0	69.5	70.3	70.3	70.3	70.5	73.5	73.5	73.5	74.1	72.0	72.0	72.0	71.1	71.5	71.5	71.5	71.5	72.5	72.5
#22 11:15 - 11:30	68.0	70.3	70.3	70.3	67.7	67.7	67.7	70.4	70.4	70.4	68.1	70.2	70.2	70.2	73.2	73.5	73.5	73.5	72.9	72.2	72.2	72.2	71.4	72.4	72.4	72.4	72.4	72.7	72.7

(b) Observed speed contour for the study period from 6:00am to 11:30am used for the set of target performance measures.

Figure 3.21 Geometry and target speed data for the I-540 WB case study.

Analysis Period	Segment																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
#1 6:00 - 6:15	75.0	71.0	69.4	74.6	71.9	63.3	70.9	71.0	73.5	68.8	74.3	66.9	72.0	69.4	73.5	69.1	71.6	68.5	73.2	67.7	71.7	68.6	73.1	67.9	71.5	74.7	70.4	74.5	74.4
#2 6:15 - 6:30	74.6	70.4	69.3	73.7	69.1	61.6	67.6	70.2	71.6	68.6	73.1	65.6	69.2	68.9	71.6	67.7	68.5	68.3	71.1	66.6	68.7	68.5	71.0	66.7	68.4	73.8	70.0	74.5	73.3
#3 6:30 - 6:45	73.9	69.8	69.2	72.3	65.4	59.0	63.3	69.0	69.0	68.5	71.3	64.1	65.6	65.6	69.1	64.6	64.6	64.6	68.4	64.8	64.8	64.8	68.2	64.4	64.4	72.4	69.6	74.4	71.6
#4 6:45 - 7:00	72.7	69.0	69.1	70.3	60.8	55.1	57.9	65.7	65.7	65.7	68.9	61.0	61.0	61.0	65.7	59.7	59.7	59.7	64.8	60.0	60.0	60.0	64.5	59.4	59.4	70.5	69.2	73.6	69.3
#5 7:00 - 7:15	71.1	66.4	66.9	29.7	35.9	54.1	56.8	64.3	64.3	64.3	67.8	58.5	58.5	58.5	63.9	56.6	56.6	56.6	62.5	56.7	56.7	56.7	62.0	55.8	55.8	69.0	69.0	72.9	67.5
#6 7:15 - 7:30	71.6	62.5	24.6	18.4	35.1	54.2	56.9	64.6	64.6	64.6	68.0	59.1	59.1	59.1	64.4	57.5	57.5	57.5	63.2	57.6	57.6	57.6	62.7	56.9	56.9	69.4	69.0	73.1	68.1
#7 7:30 - 7:45	66.3	36.3	20.8	18.4	35.3	54.2	56.9	64.9	64.9	64.9	68.2	59.7	59.7	59.7	64.8	58.2	58.2	58.2	63.7	58.4	58.4	58.4	63.3	57.7	57.7	69.8	69.1	73.3	68.5
#8 7:45 - 8:00	61.2	34.4	20.8	18.4	35.5	54.2	56.9	65.1	65.1	65.1	68.4	60.2	60.2	60.2	65.2	58.9	58.9	58.9	64.2	59.2	59.2	59.2	64.0	58.7	58.7	70.2	69.1	73.5	68.9
#9 8:00 - 8:15	67.3	51.7	21.3	18.5	35.6	54.2	56.9	65.3	65.3	65.3	68.6	60.8	60.8	60.8	65.6	59.5	59.5	59.5	64.7	60.0	60.0	60.0	64.5	59.5	59.5	70.5	69.2	73.6	69.3
#10 8:15 - 8:30	73.8	69.3	54.2	26.6	36.4	54.2	56.9	65.9	65.9	65.9	69.0	62.0	62.0	62.0	66.5	61.2	61.2	61.2	65.9	61.8	61.8	61.8	65.9	61.5	61.5	71.3	69.4	74.0	70.3
#11 8:30 - 8:45	74.4	70.2	69.3	73.1	67.3	60.2	65.2	69.6	70.3	68.6	72.2	64.9	67.4	67.4	70.3	66.6	66.6	66.6	69.8	65.8	66.8	66.8	69.6	65.8	66.4	73.1	69.8	74.5	72.5
#12 8:45 - 9:00	74.8	70.6	69.4	74.0	69.8	62.0	68.4	70.4	72.1	68.7	73.4	66.0	69.9	69.1	72.1	68.0	69.3	68.4	71.7	66.9	69.4	68.5	71.6	67.0	69.1	74.0	70.1	74.5	73.6
#13 9:00 - 9:15	75.0	70.9	69.4	74.6	71.7	63.1	70.6	70.9	73.3	68.8	74.2	66.8	71.7	69.4	73.3	68.9	71.3	68.5	73.0	67.6	71.4	68.6	72.9	67.8	71.2	74.6	70.4	74.5	74.3
#14 9:15 - 9:30	75.0	71.1	69.4	74.6	72.4	63.5	71.5	71.1	73.7	68.8	74.5	67.1	72.4	69.5	73.8	69.3	72.0	68.5	73.5	67.9	72.1	68.7	73.4	68.1	71.9	74.8	70.5	74.5	74.6
#15 9:30 - 9:45	75.0	71.2	69.4	74.6	73.0	63.9	72.2	71.3	74.1	68.8	74.5	67.3	73.0	69.6	74.1	69.6	72.7	68.6	73.9	68.2	72.7	68.7	73.8	68.4	72.6	74.9	70.6	74.5	74.7
#16 9:45 - 10:00	75.0	71.3	69.4	74.6	73.5	64.2	72.6	71.5	74.4	68.8	74.5	67.6	73.5	69.8	74.4	69.8	73.2	68.6	74.2	68.4	73.3	68.8	74.2	68.6	73.2	75.0	70.7	74.5	74.9
#17 10:00 - 10:15	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	74.0	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.8	68.8	74.5	68.9	73.7	75.0	70.8	74.6	75.0
#18 10:15 - 10:30	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	73.9	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.7	68.8	74.5	68.8	73.7	75.0	70.8	74.6	75.0
#19 10:30 - 10:45	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	73.9	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.7	68.8	74.5	68.8	73.7	75.0	70.8	74.6	75.0
#20 10:45 - 11:00	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	73.9	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.7	68.8	74.5	68.8	73.6	75.0	70.8	74.6	75.0
#21 11:00 - 11:15	75.0	71.4	69.4	74.6	73.8	64.4	72.7	71.6	74.6	68.9	74.5	67.8	73.9	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.7	68.8	74.5	68.8	73.6	75.0	70.8	74.6	75.0
#22 11:15 - 11:30	75.0	71.4	69.4	74.6	73.8	64.4	72.7	71.6	74.6	68.9	74.5	67.7	73.8	69.8	74.6	70.0	73.6	68.6	74.4	68.6	73.6	68.8	74.4	68.8	73.5	75.0	70.8	74.6	74.9

Figure 3.22 Speed contour predicted by the HCM analysis for the initial demand assignment.

3.7.3.1 Profile-Based Calibration

For the first attempt at calibrating the I-540 facility using the GA framework, the profile-based encoding is utilized. In order to make use of the profile-based encoding, it is assumed that the AADT values and the average bimodal with AM peak profile are acceptable guesses for the true underlying demands (which are entirely unknown). Since the speeds of the initial demand allocation differ greatly from the target data, a large confidence bandwidth of 20% is specified to define the search space for the encoding. Table 3.8 provides the additional parameters used for the GA and profile-based encoding. The setup parameters are the same as those used for with the exception of the confidence interval being expanded to 20%. A population size of 200 organisms is used, and the algorithm is limited to 400 iterations. A 16 digit binary encoding is used for the demand adjustment factor encoding.

Table 3.8 Summary of the setup parameters for the profile-based encoding GA runs.

Experiment Parameter	Value
Number of Runs	5
Profile Confidence Interval	20%
Population Size	200
Max Iterations	400
Binary Encoding Digits	16
Crossover Mixing %	50%
Mutation Rate (%)	2%

Table 3.9 summarizes the results of the five calibration attempts. On average, the GA is able to calibrate the model such that it results in a 48% reduction in total speed error. This corresponds to an average speed difference of around 3 mph for each segment in each time period. Figure 3.23 shows the speed contour predicted by the best calibration attempt. While there is still a moderate amount of error in the speeds, the contour does show congestion far more consistent with the target data than is seen in the initial guess (see Figure 3.22).

It is important to note that unlike the previous example, this case study is attempting to match target speeds that may not be exactly achievable by the methodology. The reported real-world speeds contain uncertainty in their measurement, and only provide an averaged snapshot of a particular point in time. There is always a guarantee of some degree of error, which holds especially true for segments and time periods where the HCM methodology reports traffic is moving at free flow speed. Free flow speed is an input of the methodology and is assumed to be one constant value

for all vehicles. Additionally, one of its primary uses is to serve as an upper bound on predicted speed. However, in real-world conditions, free flow speed will vary from driver to driver independent of the demand volume, even when demands are far below capacity. With this in mind, the calibration shown in Figure 3.23 can be considered a success as it successfully removes the incorrect congestion regime of the initially estimated contour, and now predicts a speed pattern largely consistent with the target data.

Table 3.9 Summary of the calibration results for the profile-based encoding GA runs.

Metric	Uncalibrated	GA Average	GA Best	GA Worst
Total Speed Difference (mph)	3,901.49	2,018.01	2,010.70	2,027.79
Average Speed Difference (mph)	6.12	3.16	3.15	3.18
90th Percentile Speed Difference (mph)	13.48	6.83	6.60	6.98
Max Speed Difference (mph)	50.31	32.75	28.92	34.63
GA Run Time (Minutes)	-	4.84	5.10	5.46

At this point, it is useful to take a deeper look at a breakdown of the error based on speed regime. Often the most crucial aspects of an HCM freeway facilities analysis revolve around the periods when speeds are the lowest, and consequently it is important that any calibrated facility match those conditions well. Table 3.10 shows a breakdown of the average absolute speed error for three regimes. The first regime considers any segment and time period combination in which the target speed is greater than 55 mph. The second speed regime considers a target speed less than 55mph, and the final regime considers a target speed of less than 45 mph. It is clear that for the uncalibrated facility, the errors escalate severely as the target speed drops, with mean errors of 17 mph for the second speed regime and 22 mph for the third speed regime. On the other hand, the first attempt at calibration, which uses an unweighted objective function and whose overall results are discussed above, shows a marked improvement. The error breakdown by speed regime is shown in the middle row of Table 3.10, and the errors for the lower speed regimes drop by 50% and 60%, respectively. However, this still may not provide the accuracy necessary to best make use of the model.

Analysis Period	Segment																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
#1 6:00 - 6:15	75.0	71.0	69.4	74.6	71.9	63.2	70.8	70.8	73.2	68.9	74.0	66.8	71.7	69.3	73.3	69.0	71.4	68.7	72.9	67.6	71.3	68.8	72.6	67.8	71.0	74.6	70.7	74.5	74.0
#2 6:15 - 6:30	75.0	71.2	69.3	74.6	73.7	64.2	72.6	71.3	74.1	68.6	74.5	67.3	73.0	69.5	74.3	69.7	72.9	68.1	74.4	68.5	73.5	68.6	74.4	68.7	73.4	75.0	70.6	74.5	75.0
#3 6:30 - 6:45	74.9	70.5	69.2	74.1	71.0	62.7	69.8	70.7	72.9	68.3	74.3	66.7	71.7	69.2	73.5	68.9	71.4	68.4	73.2	67.4	71.0	68.6	72.7	67.4	70.3	74.4	69.8	74.4	74.4
#4 6:45 - 7:00	74.7	70.5	69.3	73.8	71.3	62.7	69.7	70.6	72.6	68.6	73.9	65.6	69.8	68.6	72.8	68.2	70.0	68.2	72.4	67.1	70.1	68.0	72.8	67.5	70.5	74.4	69.8	74.4	74.5
#5 7:00 - 7:15	74.1	69.8	69.1	72.7	68.8	61.1	66.6	69.7	70.6	68.2	72.8	65.2	68.4	68.4	71.8	67.6	68.5	67.5	72.4	66.6	69.4	68.4	71.7	66.6	68.5	73.8	69.5	74.4	73.9
#6 7:15 - 7:30	74.1	69.9	69.1	72.8	68.9	61.3	67.1	69.6	70.5	68.2	72.9	65.2	68.4	68.4	71.9	67.7	68.6	68.1	71.5	66.3	68.4	68.1	71.3	66.5	68.2	73.7	69.0	74.4	74.0
#7 7:30 - 7:45	72.6	68.4	68.4	69.5	60.6	54.4	57.2	64.2	64.2	64.2	68.1	58.0	58.0	58.0	64.2	58.3	58.3	58.3	64.0	59.3	59.3	59.3	64.8	60.5	60.5	70.9	69.6	73.6	69.2
#8 7:45 - 8:00	71.3	67.9	67.9	69.0	58.1	52.5	55.0	62.6	62.6	62.6	66.2	54.8	54.8	54.8	58.3	52.4	46.1	47.7	32.2	56.8	56.8	56.8	62.4	57.5	57.5	69.7	69.4	73.0	67.5
#9 8:00 - 8:15	71.6	68.4	68.4	69.6	59.6	54.5	57.3	63.8	61.8	58.9	32.1	50.5	47.5	43.7	30.1	44.5	44.3	41.7	30.1	56.9	56.9	56.9	62.3	56.8	56.8	69.4	69.0	73.1	68.0
#10 8:15 - 8:30	73.5	69.7	69.3	71.9	64.8	58.5	62.5	67.5	67.5	67.5	58.2	59.2	44.9	65.4	34.4	49.1	43.7	46.2	30.8	56.9	56.9	56.9	62.2	57.8	57.8	69.8	69.3	73.1	67.7
#11 8:30 - 8:45	74.2	70.0	69.3	72.7	66.1	59.6	64.2	68.9	68.9	68.7	70.8	63.6	62.2	59.8	50.4	61.5	57.3	51.9	35.7	56.9	56.9	56.9	62.5	57.6	57.6	69.7	69.6	72.9	67.4
#12 8:45 - 9:00	73.6	69.7	69.3	71.9	64.0	58.4	62.4	69.0	69.0	68.8	70.7	63.7	64.7	64.7	67.3	62.6	62.6	62.6	66.4	62.7	62.7	62.7	66.4	63.3	63.3	72.0	70.0	74.0	70.0
#13 9:00 - 9:15	74.9	70.9	69.5	74.5	71.2	63.0	70.3	70.8	73.1	68.8	74.1	66.3	71.0	69.2	72.8	68.5	70.4	68.5	72.4	67.4	70.7	68.6	72.5	67.6	70.6	74.5	70.5	74.5	74.1
#14 9:15 - 9:30	75.0	71.3	69.4	74.6	73.6	64.2	72.6	71.3	74.2	68.6	74.4	67.4	73.2	69.5	74.4	69.8	73.2	68.6	74.3	68.3	73.2	68.4	74.3	68.5	73.1	75.0	70.9	74.6	74.8
#15 9:30 - 9:45	74.9	70.9	69.4	74.5	71.2	62.9	70.1	70.8	73.1	68.7	74.1	67.1	72.3	69.4	73.8	69.2	71.9	68.4	73.6	67.9	72.1	68.8	73.3	68.1	71.9	74.8	70.9	74.6	74.4
#16 9:45 - 10:00	74.9	70.8	69.5	74.3	69.9	62.2	68.8	70.5	72.4	69.0	73.3	66.2	70.2	69.4	71.7	68.4	69.7	68.7	71.4	67.1	69.9	68.7	71.6	67.4	69.9	74.3	70.7	74.5	73.5
#17 10:00 - 10:15	75.0	71.1	69.4	74.6	71.9	63.4	71.2	71.1	73.8	68.9	74.4	67.0	72.2	69.6	73.5	69.3	72.0	68.9	73.1	67.8	71.8	69.0	72.8	68.1	71.7	74.7	70.6	74.5	74.4
#18 10:15 - 10:30	75.0	71.1	69.5	74.6	71.8	63.2	70.7	70.9	73.4	68.9	74.2	67.2	72.5	69.5	73.8	69.3	72.1	68.8	73.3	68.0	72.1	68.8	73.3	68.2	72.1	74.8	71.3	74.6	74.3
#19 10:30 - 10:45	74.9	70.9	69.4	74.5	70.7	62.8	69.9	71.1	73.7	69.1	74.3	67.2	72.5	69.7	73.5	69.0	71.5	68.6	73.1	67.9	71.9	68.9	73.0	67.9	71.5	74.7	71.3	74.6	73.9
#20 10:45 - 11:00	74.9	70.9	69.4	74.5	70.2	62.5	69.3	70.7	72.9	69.0	73.7	66.7	71.4	69.3	73.0	68.8	70.9	68.8	72.3	67.4	70.7	68.7	72.3	67.6	70.6	74.5	71.3	74.6	73.5
#21 11:00 - 11:15	75.0	71.1	69.5	74.6	71.8	63.3	70.9	71.3	74.1	68.9	74.5	67.3	72.9	69.6	74.0	69.6	72.7	68.7	73.8	68.1	72.6	68.7	73.7	68.4	72.6	74.9	71.2	74.6	74.6
#22 11:15 - 11:30	75.0	71.0	69.4	74.6	71.5	63.0	70.4	70.9	73.3	68.7	74.2	67.0	72.2	69.5	73.5	69.3	72.0	68.7	73.3	67.9	72.0	68.7	73.3	68.0	71.7	74.7	70.9	74.6	74.2

Figure 3.23 Predicted speed contour of the profile-based demand calibration run with the lowest total speed error.

Table 3.10 Breakdown of the mean absolute speed error for each segment in each time period for the three speed regimes (average of 5 GA runs).

Speed Regime	Overall	> 55	< 55	< 45
Uncalibrated Avg Speed Error (mph)	6.11	5.31	17.22	21.83
Calibrated (Unweighted) Mean Speed Error (mph)	3.16	2.81	8.07	8.79
Calibrated (Weighted) Mean Speed Error (mph)	4.48	4.52	3.95	4.01

$$\min \sum_{i=1}^N \sum_{p=1}^P \lambda_{i,p} |v_{i,p} - v'_{i,p}|, \text{ where } \lambda_{i,p} = \begin{cases} 20 & \text{if } v_{i,p} < 55 \\ 1 & \text{if } v_{i,p} \geq 55 \end{cases} \quad (3.3)$$

In order to rectify this, the fitness function is altered such that errors occurring when the target speed is lower are weighted more heavily. Any number of weightings can be used, but a simple weighting scheme for the objective is given in Equation 3.3. This serves to weight errors where the target speed is less than 55 mph twenty times more important than when the target speed is greater than 55 mph. A value of 20 is chosen because the speed regime cutoff of 55 mph is 20 less than the facility free flow speed of 75 mph. Another 5 runs of the calibration GA are made using this new objective function, and the breakdown of the average speed error is given in the final row of Table 3.10. While the overall average speed error is higher, there is sizable improvement in the predicted speeds for the lower speed regimes, where the average error fell by more than 50% from the unweighted calibration. From this it can be seen that making use of a weighted objective function such as that in Equation 3.3 can be very helpful to tailor the focus of a calibration. More focused calibration such as this can be particularly useful when the number of periods with low speeds is outnumbered by periods in which speeds are essentially at free flow conditions.

3.7.3.2 Demand Estimation using a 3-Distribution Encoding

For a second attempt at calibrating the I-540 facility using the GA approach, the n -distribution encoding approach is utilized and the results are compared to those obtained using the profile-based encoding. Table 3.11 provides the parameters used for the GA for a demand estimation approach using a 3-distribution encoding and the unweighted objective function of Equation 3.2. The setup parameters are largely the same as those used for the profile-based encoding approach with the only changes being an increase in population size to 400 organisms, and a decrease in the number of iterations. The ranges specified for the mixture distribution are also mostly the same as with previous examples. However, for this case study, two of the location parameters are left unconstrained (within

the 24-hour daily limit), while the target data is used to restrict the location of one of the random variables. Figure 3.21b shows that the congestion falls entirely between 7:00am and 10:00am, and that range is used to limit the value of μ_1 .

Table 3.11 Summary of the setup parameters for the profile-based encoding GA runs.

(a) Genetic Algorithm Parameters		(b) Mixture Distribution Parameters		
Experiment Parameter	Value	Distribution Parameter	Symbol	Range
Number of Runs	5	Weight	w_i	0.05, 0.5
Population Size	400	Mean 1	μ_1	7:00am, 10:00am
Max Iterations	200	Mean 2	μ_2	12:00am, 11:59pm
Binary Encoding Digits	16	Mean 3	μ_3	12:00am, 11:59pm
Crossover Mixing %	50%	Std. Dev.	σ_i	0.5, 5.0
Mutation Rate (%)	5%	Skewness	a_i	-10, 10

A summary of the results of the experiment is given in Table 3.12, and the contour predicted by the estimated demands for the best run is shown in Figure 3.24. The contour shows a patch of lowered speeds from 7am to 9am, with a shape roughly matching that of the target speed data, indicating a successful calibration. While a margin of error is still present in the prediction, when compared to the initial uncalibrated model of the previous section (see Figure 3.22), it shows predicted performance that is far more true to the real-world data. Further, the average error of 3.12 mph is slightly less than that obtained with the profile-based calibration approach. This implies that for a situation with this little of quality input data available, estimating the demand profile using the n -distribution approach is the better alternative.

Table 3.12 Summary of the setup parameters for the profile-based encoding GA runs.

Metric	GA Average	GA Best	GA Worst
Total Speed Difference (mph)	1991.84	1885.49	2105.31
Average Speed Difference (mph)	3.12	2.96	3.30
90th Percentile Speed Difference (mph)	6.27	5.72	6.95
Max Speed Difference (mph)	30.14	28.06	32.72
GA Run Time (Minutes)	8.5	8.3	9.2

Analysis Period	Segment																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
#1 6:00 - 6:15	75.0	71.6	68.6	74.5	74.7	65.1	72.8	71.7	74.8	69.5	74.5	67.9	74.0	70.1	74.6	70.2	73.8	69.5	74.0	68.5	73.4	68.5	74.5	69.3	74.3	75.0	72.4	74.7	74.9
#2 6:15 - 6:30	75.0	71.6	68.8	74.5	74.6	65.0	72.8	71.5	74.6	69.4	74.5	67.9	73.8	70.0	74.4	70.1	73.6	69.5	73.9	68.5	73.3	68.2	74.5	69.4	74.4	75.0	72.5	74.7	74.9
#3 6:30 - 6:45	75.0	71.5	69.0	74.6	74.1	64.7	72.7	70.9	73.7	69.5	74.0	67.4	72.7	69.9	73.5	69.5	72.3	69.5	72.6	67.9	71.8	67.9	73.9	69.0	73.7	75.0	72.5	74.7	74.6
#4 6:45 - 7:00	75.0	71.4	69.2	74.6	73.9	64.6	72.7	70.0	72.5	69.4	72.9	67.0	71.4	69.7	72.4	68.9	70.9	69.4	71.4	67.4	70.3	67.5	73.5	68.8	73.3	75.0	72.4	74.7	74.3
#5 7:00 - 7:15	75.0	71.3	69.3	74.6	73.5	64.4	72.7	68.1	70.1	69.3	70.8	66.0	68.6	68.6	70.0	67.7	67.9	67.9	68.5	66.2	67.1	67.0	72.3	68.4	72.1	74.8	72.1	74.7	73.7
#6 7:15 - 7:30	75.0	71.1	69.3	74.6	73.1	64.2	72.6	64.4	66.7	66.7	67.6	64.9	65.0	65.0	66.8	64.1	64.1	64.1	64.8	63.0	63.0	63.0	70.8	67.9	70.6	74.5	72.1	74.7	72.5
#7 7:30 - 7:45	75.0	71.0	69.4	74.6	72.6	63.9	72.2	58.3	62.6	62.6	63.9	61.4	61.4	61.4	63.7	60.4	60.4	60.4	61.2	59.1	59.1	59.1	69.4	67.4	69.2	74.0	72.0	74.5	71.4
#8 7:45 - 8:00	75.0	70.8	69.4	74.4	71.7	63.5	71.3	49.7	58.4	58.4	59.6	56.4	56.4	56.4	58.8	57.7	48.1	48.2	38.9	56.8	56.8	56.8	68.3	67.0	68.1	73.7	72.0	74.2	70.4
#9 8:00 - 8:15	74.9	70.7	69.5	74.1	71.2	63.2	70.8	49.4	57.9	57.9	59.1	51.3	43.0	61.0	47.6	42.2	40.8	38.9	36.7	56.9	56.9	56.9	68.2	66.9	67.8	73.6	71.9	74.2	70.2
#10 8:15 - 8:30	74.9	70.5	69.5	73.7	70.8	63.0	70.4	56.3	54.8	62.7	50.2	43.6	41.0	59.7	41.4	45.6	46.6	42.7	36.5	56.9	56.9	56.9	67.9	66.8	67.7	73.6	71.8	74.2	70.2
#11 8:30 - 8:45	74.8	70.3	69.5	73.3	70.5	62.9	70.1	62.8	64.2	64.2	64.3	58.5	46.7	60.8	55.8	51.8	42.6	40.6	36.7	56.9	56.9	56.9	67.2	66.6	66.9	73.3	71.7	74.0	69.8
#12 8:45 - 9:00	74.6	70.1	69.5	72.9	70.5	62.9	70.1	66.7	67.4	67.4	68.3	65.5	66.5	66.5	68.1	59.5	60.1	57.3	45.8	59.3	59.3	59.3	67.6	66.8	67.4	73.5	71.8	74.1	70.0
#13 9:00 - 9:15	74.4	70.0	69.5	72.5	70.7	63.0	70.3	68.7	69.8	69.3	70.5	66.3	69.1	69.1	70.5	67.5	67.8	67.8	68.5	66.1	66.8	66.8	71.2	68.0	71.1	74.6	72.1	74.7	72.8
#14 9:15 - 9:30	74.2	69.8	69.5	72.1	70.9	63.0	70.4	69.8	71.3	69.3	71.9	66.9	70.8	69.6	71.9	68.3	69.7	69.3	70.3	66.8	68.8	67.9	72.0	68.3	71.8	74.7	72.3	74.7	73.2
#15 9:30 - 9:45	74.0	69.7	69.5	71.7	71.1	63.2	70.7	70.5	72.5	69.3	73.0	67.3	72.2	69.8	73.1	69.0	71.3	69.4	71.7	67.3	70.3	68.3	72.6	68.5	72.4	74.8	72.3	74.7	73.6
#16 9:45 - 10:00	73.8	69.6	69.5	71.3	71.4	63.3	70.9	70.7	73.1	69.4	73.5	67.5	72.7	69.9	73.5	69.2	71.8	69.4	72.3	67.6	71.0	68.5	72.8	68.5	72.6	74.9	72.2	74.7	73.8
#17 10:00 - 10:15	73.5	69.5	69.5	71.0	71.4	63.3	70.9	70.9	73.4	69.4	73.8	67.7	73.1	69.9	73.9	69.5	72.5	69.4	72.9	67.8	71.7	68.7	73.1	68.7	72.9	74.9	72.3	74.7	74.0
#18 10:15 - 10:30	73.3	69.4	69.5	70.8	71.1	63.1	70.6	71.1	73.6	69.4	74.0	67.8	73.4	70.0	74.1	69.7	72.9	69.4	73.2	68.0	72.2	68.9	73.3	68.8	73.2	75.0	72.4	74.7	74.1
#19 10:30 - 10:45	73.0	69.3	69.5	70.6	70.4	62.8	70.0	71.0	73.6	69.4	73.9	67.8	73.4	69.9	74.1	69.8	73.1	69.4	73.5	68.1	72.5	69.0	73.4	68.8	73.3	75.0	72.5	74.7	74.2
#20 10:45 - 11:00	72.8	69.3	69.5	70.5	69.3	62.2	68.8	70.9	73.3	69.4	73.7	67.7	73.0	69.9	73.8	69.6	72.7	69.4	73.1	67.9	71.9	69.1	72.8	68.5	72.6	74.9	72.3	74.7	73.7
#21 11:00 - 11:15	72.7	69.3	69.5	70.4	67.9	61.4	67.3	70.7	72.9	69.3	73.4	67.6	72.7	69.8	73.6	69.6	72.5	69.4	72.9	67.8	71.6	69.2	72.4	68.4	72.2	74.8	72.3	74.7	73.5
#22 11:15 - 11:30	72.6	69.3	69.5	70.5	66.9	60.8	66.2	70.7	72.7	69.3	73.2	67.5	72.5	69.8	73.5	69.6	72.5	69.4	72.9	67.8	71.7	69.2	72.4	68.4	72.2	74.8	72.4	74.7	73.3

Figure 3.24 Predicted speed contour of the demand estimation run with the lowest total speed error.

Just as with the profile-based encoding, it is important to consider the quality of a calibration in terms of the error by speed regime. Table 3.13 provides a breakdown of the error for the five calibration runs discussed above with an unweighted objective, as well as for five additional calibration runs made with the same 3-distribution encoding and the regime-weighted objective of Equation 3.3. The breakdown for the runs with the unweighted objective again shows that the GA was less accurate in matching the lower speed regimes than the higher speed regimes. For the calibration attempts using the weighted objective function, the distribution of error is much more even. There is a large reduction in the average speed error for both of the lower speed regimes, with the lowest regime seeing nearly a 70% improvement. This reinforces the usefulness of a weighted objective function when the number of periods with high target speeds outnumber those with lower target speeds.

Table 3.13 Breakdown of the mean absolute speed error for each segment in each time period for the three speed regimes (average of 5 GA runs).

Speed Regime	Overall	> 55	< 55	< 45
Calibrated (Unweighted) Mean Speed Error (mph)	3.12	2.71	9.08	10.65
Calibrated (Weighted) Mean Speed Error (mph)	3.56	3.52	4.48	3.35

3.8 Summary and Conclusions

This paper presents a genetic algorithm framework for calibration of demand volumes for the HCM's freeway facilities methodology. Two encoding approaches are developed for use in the framework. First, a profile-based encoding is proposed that creates a search space around a known initial profile and corresponding AADT value. The second encoding requires knowledge of just the AADT for a study period and generates a mixture distribution of random variables that is used to allocate demand volumes for the study period. The two encodings represent alternative approaches to the underlying problem and can be used for different modeling scenarios.

Two computational experiments confirm the framework's ability to calibrate a model based on observed segment speeds. First, for an artificially designed facility with known demands, both encodings are able to calibrate the model starting from incorrect or restricted information. In the testing, the profile-based approach was able to eliminate 98% of the error of the initial incorrect demand allocation, producing a predicted set of speeds that differs on average less 0.5 mph from

the true speeds. Alternatively, the n -distribution approach achieves a similar level of accuracy for the constructed example while concurrently generating allocation profiles using only the AADT values for the facility and information gained from an examination of congestion observed in the target data. While not a real-world example, the experiment of Section 3.7.2 serves to demonstrate the framework's ability to accurately calibrate a model for a full 24-hour study period and generate demand volume allocations closely matching a set of known values.

The second experiment presents a case study using real-world data for a stretch of urban freeway outside of Raleigh, NC. With minimal information known outside of AADT values and a target speed contour, both the profile-based approach and the n -distribution approach are shown to reliably eliminate an incorrectly predicted patch of congestion from the initial calibration, and generate results more consistent with the real world data. Additionally, by introducing a weighted objective function, the accuracy of both approaches improves significantly for segments and time periods with key low target speeds. For the 16 entry and exit points of the case study facility, the GA is able to concurrently manipulate the 400 individual demand volumes and bring the average error in speed down to around just 3 mph, representing a 50% improvement on the initial best guess. Finally, the results of the experiments demonstrate that much of the error remaining after calibration is confined to off-peak analysis periods in the early morning or late at night — periods which do not generally affect an analysis and whose calibration would require true traffic count data.

Overall, this paper presents a novel approach for calibrating demand volume inputs for HCM freeway facilities analyses. The approach improves on the manual's existing trial and error guidance and provides a clear path to a creating a calibrated facility. The approach brings a new degree of flexibility to the process by allowing analysts to incorporate as much or as little information as is available. Further, the automated nature of the process vastly reduces the burden placed on an analyst attempting to utilize the methodology and simplifies an essential step undertaken before any HCM model can be used planning or congestion management applications.

3.9 Future Work

There are a number of avenues for future research based on the work presented in this paper. The two demand calibration approaches introduced are used independently, but a hybrid approach combining the two would allow an analyst increased flexibility based on the information available on a ramp by ramp basis. One downside of the current approach is that it cannot always predict inputs reliably by matching a single performance measure. For example, as mentioned in Section 3.6.1, speeds cannot be used to predict flows for periods when demand is far below capacity. Future work could incorporate a multi-objective fitness function to account for additional performance

measure matching when the data is available. This work tests only a small number of objectives, but more complex functions and weightings can be developed to incorporate factors such as segment lengths and other geometric factors. Additionally, while each proposed encoding has its own set of limitations, they are in many ways complementary approaches. An extension to the framework could make use of both in a two-stage GA or by incorporating a second encoding in the search process through memetic lifetime learning. The demand estimation framework could also be used in conjunction with a similarly constructed capacity-based encoding to make use of additional parameters to further improve the quality of the calibration. Further, if an analyst has target speeds for multiple days, the n -distribution encoding could be used to generate demand profiles for each day, which could then be compared and used to identify the underlying profile shape. Overall, by providing an improved and simplified approach to model calibration allows for better quality analysis to be conducted on top of the core model for applications like congestion management and network design.

CHAPTER

4

A TWO STAGE APPROACH FOR UNIFIED DEMAND AND CAPACITY CALIBRATION FOR CONGESTED FREEWAY FACILITIES

4.1 Introduction

The *Highway Capacity Manual* (2016) is a widely used reference material for conducting freeway analyses. The HCM defines a freeway facility as an extended length of freeway composed of continuously connected basic, weaving, merge, and diverge segments (*Highway Capacity Manual* 2016). A facility is generally a 9 to 12-mile portion of a single freeway, and each analysis is conducted for a contiguous study period typically falling within a single 24-hour time span. A study period is further divided into individual 15-minute Analysis Periods (AP), each which requires a distinct set of inputs.

Each model must be calibrated for a specific study site before it can be utilized for planning or operational purposes. The calibration process is designed to ensure that the methodology reports performance measures consistent with available real-world data under conditions such as recurring daily congestion. Moreover, a well-calibrated core freeway facilities analysis will enhance the quality of extended Travel Time Reliability and Active Traffic and Demand Management (ATDM) analyses

(*Highway Capacity Manual 2016*). Due to the important role of calibration for effective use of the methodology, the upcoming HCM introduced a framework for calibrating the core freeway facility analysis. Since most geometric inputs for a facility (segment length, number of lanes, etc.) are static and known explicitly *a priori*, calibration of a facility is largely done through the manipulation of parameters relating to free flow speed, vehicle demand volumes, and segment capacities. This is reflected in the overall process flow of the proposed calibration procedure as shown in 4.1.

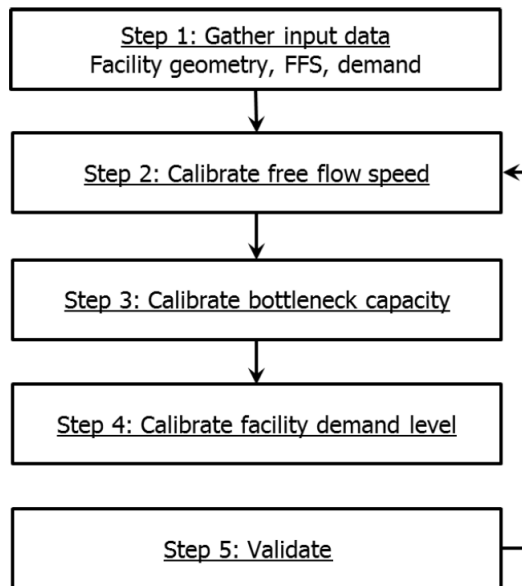


Figure 4.1 Core freeway analysis calibration framework in the 6th edition of the HCM (*Highway Capacity Manual 2016*).

The first two steps of the calibration process are straightforward and predominantly consist of gathering data that are generally readily available. For the first step, geometric aspects of a facility can be determined from maps or photos of the facility, and free flow speeds can initially be based on facility speed limits. Additionally, average annual daily traffic counts (AADTs), which serve as the basis of demand estimation, are often collected and published by each state's department of transportation. For the second step, calibration of free flow speeds (FFS) is typically the most straightforward piece of the calibration process. FFS adjustment can be done simply by observing unimpeded vehicle speeds during uncongested periods.

Steps 3 and 4 represent the most crucial aspects of a model's calibration. In fact, the true crux

of the freeway facilities methodology lies in the relationship between vehicle demand flows and segment throughput capacities. When demand does not exceed capacity, all demand will be met, and performance measures can be estimated independently for each segment of the facility. However, once demand exceeds capacity, a breakdown of flow occurs, and additional sets of equations are required to accurately model the ensuing fallout. The resulting congestion and vehicle queues can span multiple segments, requiring a unified approach for facility analysis. Under these congested conditions, the relationships between demand inputs, bottleneck capacity, and predicted performance measures are highly complex, with even slight variations culminating in sometimes unexpected and wide-ranging effects. Thus it is critically important that facility demands and segment capacities are modeled accurately. The process of defining and adjusting the inputs that determine these quantities presents the biggest challenge to applying the methodology.

Unfortunately, guidance concerning the specific processes for adjusting capacity and balancing demands is sparse in the HCM. Each process involves a large amount of uncertainty due to the vast number of required inputs and the complex underlying relationships. Further complicating matters is the fact that some aspects of both capacity and demand are indeterminate, with neither having a widely available measurement or methodology from which precise direction can be drawn. This in turn precludes the development of any simple procedural approach for the corresponding component of the calibration process. As such, the HCM can currently only recommend a process of trial and error in which slight adjustments to capacity and demand are made repeatedly until an acceptable level of matching is achieved with the target calibration data.

The work presented in the previous chapter strived to address the initial issue of demand estimation and calibration. A genetic algorithm (GA) metaheuristic approach was developed in order to automate the process by framing it as a system identification problem. However, estimating the facility demand only accounts for part of the issue, and a calibration of capacity parameters is also necessary to both maximize the accuracy of and fully utilize the HCM methodology. While the process of obtaining and balancing facility demands is arguably the largest barrier to using the HCM freeway facilities methodology, correctly calibrating the capacity for the facility is equally as important for ensuring the methodology faithfully and accurately models real world conditions.

This paper builds on the previous work using an evolutionary metaheuristic approach. The goal of this paper is to provide an automated calibration approach rooted in well-known mathematical optimization techniques. By incorporating bottleneck calibration and capacity estimation into the system identification problem, the GA approach is extended to include three key capacity parameters: pre-breakdown bottleneck capacity adjustments, the bottleneck capacity drop factor, and the facility wide jam density. The proposed approach serves to unify the processes of demand and capacity calibration by providing a single framework for use with the HCM freeway facilities

computational engine.

4.2 Literature Review

The freeway facilities methodology of HCM uses a modified Cell Transmission Model (CTM) to simulate traffic flow and congestion of freeways. The CTM is a macroscopic approach introduced by Daganzo (1994, 1995). While research has been conducted regarding calibration of the cell transmission model, the work largely corresponds to density estimation and validation as opposed to adjusting capacities. Some of the earliest work on this occurred when Munoz et al. (2003) developed the modified cell transmission model (MCTM) and the switching mode model (SMM). The authors were also able to show that the CTM and the MCTM/SMM were able to estimate densities within freeway facilities with reasonable accuracy. Muñoz et al. (2004) proposed a methodological calibration procedure for the CTM/MCTM, and further showed the methods could accurately estimate density on longer stretches of freeway. More recently, this procedure was further explored by Zhong et al. (2014), but their proposed nonlinear optimization framework showed an inability to reliably obtain globally optimal results under real-world conditions.

The HCM does propose a set of default capacity values for use with the freeway facilities methodology base on different free flow speeds (*Highway Capacity Manual* 2016). However, real world observations stemming from validation of the SHRP 2 L08 project revealed that the observed capacity numbers in the field are usually lower than the proposed values in the HCM (Hadi et al., 2014; Nisbet et al., 2014; Sobolewski et al., 2014; Williges et al., 2014). Elefteriadou & Lertworawanich (2003) conducted a study on capacity and provided three definitions to help with its quantification. However, they also found that the ranges of the value under each of the three definitions could vary largely depending on the location of the study site.

When calibrating a facility, an analyst should choose a set of target measures that represents typical performance for the facility under the conditions the analysis is striving to model (*Highway Capacity Manual* 2016). Conditions modeled can range from normal recurring congestion for weekdays and weekends to highly congested flows for extreme situations. When selecting the target data, the analyst should be careful to ensure that it accurately represents the desired conditions, and does not include non-recurring congestion events such as incidents, severe inclement weather, or work zones. Additionally, overall demands can vary by both month and weekday, and multipliers may be necessary to increase or decrease known AADT values (*Highway Capacity Manual* 2016). For example, demands are generally higher in the summer months as opposed to the winter months.

In theory, each model can be calibrated with regards to any of the outputs provided by the methodology. However, it is likely that in most cases the model will be calibrated in respect to

speeds, travel times, or observed flow volumes. Data can be collected from a number of sources included state and national departments of transportation, as well as from online data bases such as RITIS.org (Maryland CATT Lab, 2008). Using these sources, performance data collected at traffic management centers (TMCs) along roadways is often presented as travel times or speeds. While TMC segments will likely not match with the exact segmentation as required by the HCM methodology, a simple weighted averaging scheme can be applied to determine speeds for the HCM segments.

In the event that the estimated demand inputs do not identify the location and severity of a bottleneck observed in the field, the guidance of the HCM simply recommends reducing the capacity of a segment by a set increment until the bottleneck appears (*Highway Capacity Manual* 2016). Analogously, if an unwanted bottleneck appears in the model, the guidance recommends the capacity of a segment be increased by a set increment until the bottleneck no longer appears. Beyond this, the guidance of the manual provides little to no insight as to how to adjust capacity to match predicted speeds to those observed under real world conditions. Hence in many cases, the capacity calibration process comes down to trial and error until the analyst can achieve a satisfactory level of accuracy.

4.3 Capacity Calibration Parameters

Capacity calibration revolves around three key parameters. The first is a set of pre-breakdown capacity adjustments known as capacity adjustment factors (CAFs). While the HCM includes procedures for estimating the capacity of a segment based on its type and geometry, capacity values for specific facilities may vary from these default values. A CAF can then be applied as reduction on the estimated base capacity. Pre-breakdown capacity adjustment is particularly important for bottleneck segments, where the actual capacity may differ more widely from the HCM default value.

The effect of different pre-breakdown capacities on the speed-flow diagram is demonstrated below in Figure 4.2. On the chart, the vertical axis represents the speed of vehicles in the segment, and the horizontal axis represents the corresponding flow observed in a segment. Two alternatives for pre-breakdown capacity are shown as dashed lines. The major effect of differing pre-breakdown capacities is the flow rate at which breakdowns occur. Lowering the pre-breakdown capacity of a segment will cause breakdowns in the segment to occur at lower flow rates. Speeds will also drop off more sharply as the capacity of the segment is reached. Additionally, a lower capacity value results in an increase of the breakdown formation shockwave speed, and a decrease for the breakdown recovery shockwave speed.

The second key parameter is the capacity drop percentage, which models the phenomenon where the queue discharge rate of a segment following a breakdown is generally lower than the

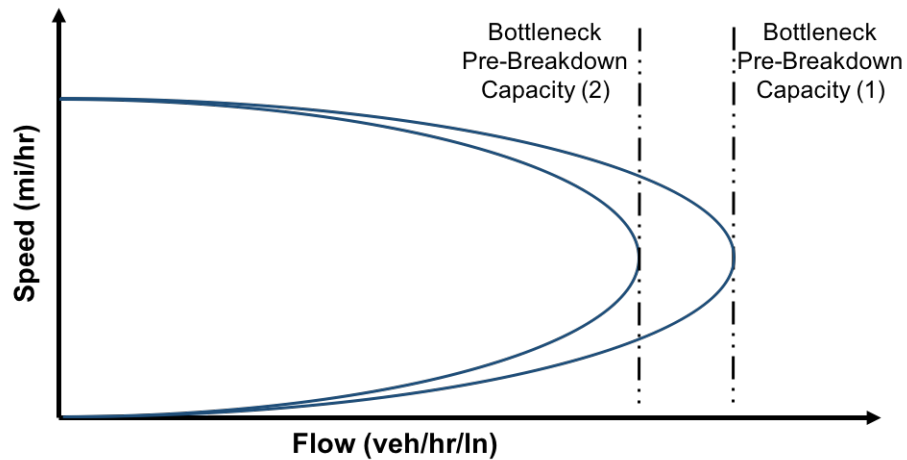


Figure 4.2 Effects of different pre-breakdown bottleneck capacities (*Highway Capacity Manual* 2016).

pre-breakdown capacity (*Highway Capacity Manual* 2016). Once demand has exceeded capacity at a bottleneck segment, the capacity drop percentage is applied resulting in a steeper loss of capacity and a reduced segment throughput. The effects of different queue discharge rates are shown in Figure 4.3. Note that the values for alpha are exaggerated to make the effects more easily visible, but in reality the drop in queue discharge rate will be much smaller. Similar to differing pre-breakdown capacities, a lower queue discharge rate (larger capacity drop percentage) increase the speed of the breakdown formation shockwave, and reduces the speed of the recovery shockwave. The two parameters differ in that the capacity drop percentage is applied only AFTER the pre-breakdown capacity has been exceeded, and does not have any effects on when breakdowns occur with respect to varying demand values. The HCM notes that the body of literature devoted to studying the capacity drop percentage reports a variety of values range from slightly negative to as much as 20%, but recommends a default value of 7% for most facilities (*Highway Capacity Manual* 2016).

Further, the capacity drop percentage is particularly noteworthy in terms of calibration for an additional reason. As previously mentioned, the freeway facilities methodology is divided into two subsets based on whether or not demand has exceeded capacity at some point along the facility. This effect represents a “jump” in the methodology, and much of the model’s sensitivity to parameters revolves around this tipping point. The capacity drop factor further exacerbates this “jump” as it increases the distance covered by the discontinuity. For example, consider a segment with a capacity of 4,800 veh/hr/ln and an hourly demand volume of 5,000 veh/hr/ln. If no capacity drop is considered, this corresponds to a demand to capacity (D/C) ratio of around 1.04. However, if the facility has a capacity drop factor of 7%, the capacity drops to 4,464 veh/hr/ln and D/C ratio jumps

up to 1.12, indicating the effects of the breakdown will be much more severe.

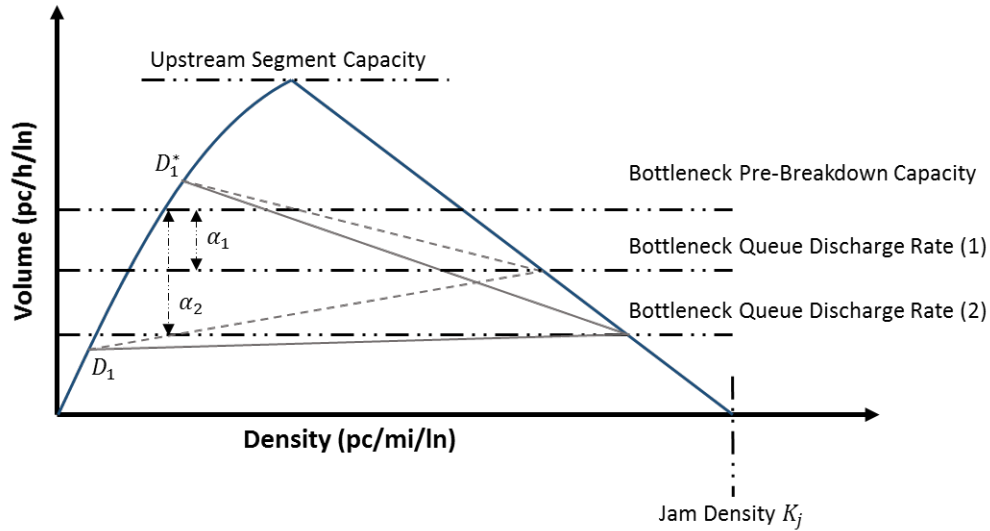


Figure 4.3 Effects of two different capacity drop percentages α_1 and α_2 (*Highway Capacity Manual 2016*).

Lastly, the jam density of a segment plays a large role in determining the size and impact of any queues that form. The jam density of a segment, denoted K_j , represents the maximum allowable density of stopped vehicles on that segment. Alternatively, it can be thought of as representing the minimum allowed spacing between vehicles of a fully queued segment (*Highway Capacity Manual 2016*). Figure 4.4 shows the effects of two different values for jam density on the volume-density diagram. Larger values for the jam density reduce speeds of both the breakdown formation and recovery shockwaves. This is due to the length of the queue growing more slowly as the queue itself is allowed to be denser. As such, jam density play its largest role in determining the queue length, and its impact is most noticeable in variations of individual segment's speeds in and around the queue resulting from a bottleneck. In most cases, jam density affects the facility travel time less than both pre-breakdown capacities and the capacity drop percentage.

4.4 Genetic Algorithm Overview

Metaheuristic approaches are often employed for problems in which the underlying mathematical relationships of a system are poorly understood or prohibitively complex. Unlike classical

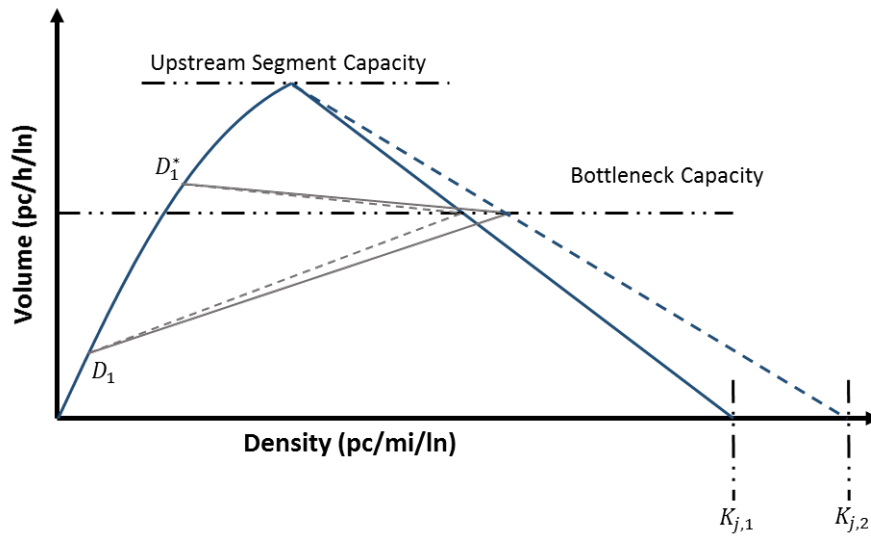


Figure 4.4 Effects of varying jam density values $K_{j,1}$ and $K_{j,2}$ on the volume density diagram (*Highway Capacity Manual* 2016).

approaches that require the use of a gradient or similar quantifiable concept of change, a meta-heuristic approach does not need to explicitly consider the underlying mathematics of the system being modeled in order to identify a solution. In particular, evolution based metaheuristics such as genetic algorithms have been widely studied and have been found to be very effective at solving optimization problems related to system identification.

A genetic algorithm (GA) seeks to mimic the processes of evolution and natural selection in order to evolve an initial pool of candidate solutions over time until an optimal or near-optimal solution is found. In a GA, each candidate solution is represented as an individual organism, with the decision variables of the problem encoded as the organism's genes or chromosomes. An evaluation function is defined in order to calculate the fitness of a candidate solution based on these "genes". Both the encoding and fitness function are chosen such that the problem is framed as a minimization or maximization of fitness. This allows fitness-based "competition" to be used in order to make determinations on the quality of one candidate solution versus another. In this way, the fitness function is analogous to the objective function of classical optimization techniques such as linear programming.

A GA begins with an initial population of randomly generated organisms. The search process operates by "evolving" this population over time, with each new generation of organisms providing an opportunity to identify a better solution. Successive generations are created through competitive

selection and crossover of organisms of the previous generation. The evolution process consists of competition-based "mating" of organisms in order to create "offspring" organisms. This is done through the use of a crossover operator, which creates the offspring organisms by "crossing-over" and combining the parents' genetic material into new chromosomes. For example, if one parent's candidate solution was encoded as the binary string "1111," and the other parent's was encoded as "0000," a simple one-point crossover strategy could create two offspring organisms with the chromosomes encoded as "1100" and "0011," respectively. Additional operators such as mutation are utilized to help prevent premature convergence to undesirable local optima. The overall process of a genetic algorithm repeats until either an optimal solution is found or a maximum number of iterations is reached.

The most critical aspect of a GA is the encoding of a candidate solution's decision variables as the genes of an organism. The key underlying assumption of the GA approach is that high-quality solutions contain common "building blocks" found in the encoded genes of the organisms (Mitchell, 1998). Maintaining these blocks within a population of organisms, and combining them with other "good" blocks should then theoretically lead to the discovery of better solutions. Blocks are combined through the use of a crossover operator representing mating and reproduction of organisms. This process produces a number of "offspring" organisms each containing parts of each parent's genetic material and consequently the encoded decision variable values. While real-valued encodings are occasionally used, most encodings consist of some type of binary representation of the solution. This allows for a straightforward realization of genetic building blocks (e.g. small binary patterns such as "010" or "110"), and provides a simple notion of sharing genetic material through exchanging binary digits.

In most cases in which a metaheuristic approach such as a GA is necessary to solve a problem, the solution space is likely to be highly nonlinear, nonconvex, and discontinuous. This makes it extremely difficult for any search process to directly or smoothly move towards a globally optimal solution. Thus, most GA implementations make use of at least one additional operator to guard against premature convergence to locally optimal solutions. The most common of these operators is the mutation operator, which randomly alters an organism's individual genes or chromosomes in order to introduce additional genetic diversity into the population.

4.5 Unifying Demand and Capacity Calibration

Overall, there are three key aspects of capacity that should be considered when calibrating an HCM freeway facilities model, each of which affects the predicted performance measures in a different manner. First, pre-breakdown bottleneck capacity adjustments determine the demand

level at which breakdowns occur, and have inverse effects on the shockwave formation and recovery speeds. Second, a reduced queue discharge rate via a capacity drop factor models a known real-world phenomenon and provides an additional control on the breakdown shockwave formation and recovery speeds that only affects conditions post-breakdown. Lastly, the jam density of the facility controls the length of any queues that form, which is reflected in variations in the speeds of individual segments in and around the vehicle queue.

As previously mentioned, a genetic algorithm metaheuristic approach was developed to automate the process of demand estimation and adjustment. The approach is able to estimate new or adjust existing demand volume inputs such that the performance predicted by the methodology more accurately matched that of a set of target real-world conditions. While it proved to be an improvement over the current trial and error process, its focus only on demand left it with some inherent limitations. On its own, demand calibration is best suited for matching observed throughput values and travel times. This may be sufficient for some facilities, but in general, and especially when calibrating for individual segment speeds, it is better served when complemented with capacity calibration. By incorporating capacity parameters that control additional relevant aspects of congestion analysis (e.g., breakdown occurrence, formation speed, queue length, and breakdown recovery), the power and flexibility of the system identification framework can be maximized.

The proposed framework consists of two-stage approach of successive calibration instances, each of which operates on a different set of variables during the GA search process. The first round of calibration focuses on demand estimation, and incorporates the capacity drop factor alongside the demand volume input variables. The second round utilizes the determined values for these inputs, and operates instead on the pre-breakdown bottleneck adjustments and the jam density. The objective function for both instances is a weighted combination of error in individual segment speeds and error in the total facility travel times for each analysis period. However, the ratio of the weights will be different for the successive calibration rounds, which will be discussed in more detail in the following sections. Finally, as with the initial demand calibration work, the framework is intended for use with FREEVAL, the HCM freeway facilities computational engine. Figure 4.5 shows the high-level process flow of the unified two-stage approach.

4.6 Fitness Evaluation

The goal of the calibration problem is to ensure that the HCM's methodology predicts performance measures consistent with those observed in the real world. As such, a straightforward choice for quantifying an organism's fitness is to compute the sum of the absolute differences between the predicted and real world values for a given performance measure. The overall objective of the GA can

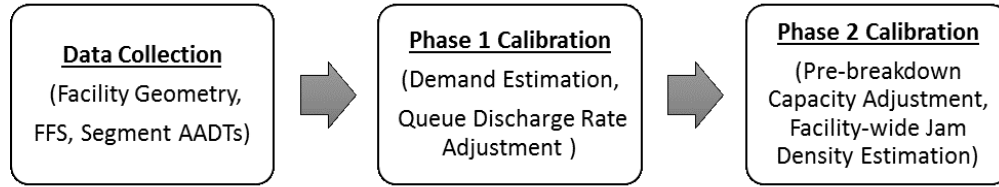


Figure 4.5 High-level process flow of the unified two-stage calibration process.

then be framed as minimizing an organism's fitness, with a fitness value of 0 indicating an optimal solution in which the predicted and real-world performance measures match exactly.

Since speeds and travel times are two of the more commonly available types of real world observed data, as well as key quantities for model calibration, it makes sense to center the fitness evaluation around these two performance measures. While a simple objective function could seek to minimize the error around just one of these measures, the dual nature of the calibration approach can benefit from minimizing a linearly weighted combination of the two, simulating a basic multi-objective approach.

First let $T(p)$ represent the observed facility travel time of period p , and let $\bar{T}(p)$ represent the predicted facility travel time for period p . Similarly, let $v_{i,p}$ represent the observed speed in segment i in analysis period p , and let $\bar{v}_{i,p}$ represent the predicted speed for segment i in analysis period p . The overall goal is to minimize the absolute error in between observed and predicted values for each quantity across all segments $(1, \dots, NS)$ and for all time periods $(1, \dots, P)$. For two linear weights λ_1 and λ_2 , the fitness function $F(x)$ for a given organism can be formulated as shown in equation 4.1.

$$F(x) = \lambda_1 \sum_{p=1}^P |T(p) - \bar{T}(p)| + \lambda_2 \sum_{i=1}^{NS} \sum_{p=1}^P |v_{i,p} - \bar{v}_{i,p}| \quad (4.1)$$

While this represents the fitness function in its simplest form, it can be modified in a near endless number of ways and tailored to a specific goal or application. For example, the work developing the demand estimation framework found that it was beneficial to weight errors occurring where targets speed are lower more heavily than those occurring where the target speeds are higher. There are a variety of ways to do this, but a simple linear piecewise weighting scheme was proposed in the demand estimation work that achieved satisfactory results. In this scheme, errors where the target speed is less than 55 mph are weighted as much as twenty times more heavily than where the target speed is greater than 55 mph. This allows the GA to focus on the replicating the congested conditions, as opposed to trying to perfectly match free flow speeds in uncongested periods. While

free flow speed is a constant value in the HCM methodology, it can fluctuate widely in the real world depending on the type of driver (e.g. one who speeds versus one who strictly obeys the speed limit) passing through the observation zone in a given time period. Thus the fitness function can be reformulated with this modification as shown in equations 4.2 and 4.3.

$$F(x) = \lambda_1 \sum_{p=1}^P |T(p) - \bar{T}(p)| + \lambda_2 \sum_{i=1}^{NS} \sum_{p=1}^P w_{i,p} |v_{i,p} - \bar{v}_{i,p}| \quad (4.2)$$

$$\text{where } w_{i,p} = \begin{cases} 20 & \text{if } v_{i,p} < 55 \\ \max\{75 - v_{i,p}, 0\} & \text{if } v_{i,p} \geq 55 \end{cases} \quad (4.3)$$

In addition to allowing the objective function to consider multiple performance measures, the weights λ_1 and λ_2 serve an additional purpose for the two-stage calibration process. They allow the focus of the optimization to be shifted between the two quantities based on the phase of the calibration process. The first phase of the calibration process centers on demand estimation, and hence should place a heavier weight on matching the target facility travel times. Alternatively, the second phase of the calibration process centers on pre-breakdown capacity adjustments and jam density, and matching individual segments speeds should be a higher priority and given a heavier weight. The exact ratio of the weights can be adjusted for each particular calibration instance, but a simple two-to-one ratio can serve as a default starting point.

4.7 Genetic Encoding of the Capacity Calibration Parameters

The GA framework of the original demand calibration work requires that all decision variables be represented as a string (or strings) of binary digits. As such, a binary encoding process analogous to that of the profile based demand adjustments is used to incorporate capacity calibration parameters into the existing framework. This type of encoding necessitates an assumption of a known range of feasible values for each parameter, which requires that both a minimum and a maximum possible value be specified. The interval created by this range can then be discretized using the maximum integer value allowed by the number of binary digits of the encoding. In this way, the binary value of a gene will represent an integer value, which can then be projected into the discretized search space. To demonstrate, if the search space for a pre-breakdown CAF is the interval $[0.7, 1.01]$, and a 5-digit binary encoding is used, the range is discretized into 31 intervals, which gives a step size of $(1.01 - 0.7) / (2^5 - 1) = 3.1 / 31 = 0.01$ (projection into the range is inclusive, hence why there are $2^5 - 1 = 31$ intervals as opposed to $2^5 = 32$). For a gene expressed by the binary string "010100", it

is converted to base 10 as an integer value of 20, and the resulting CAF is $0.7 + (20 * 0.01) = 0.90$ (a simplified range is used for demonstration purposes). Analogous approaches are used to encode and project both the capacity drop factor and the facility-wide jam density.

The specified allowable range for each parameter can vary for each facility and should be chosen using the analyst’s judgment and knowledge of the facility. However, any range should not be too large. For example, pre-breakdown CAFs indicating a drop in capacity of 30% or more are likely to be unrealistic and may indicate that other inputs of the methodology need to be adjusted. Table 4.1 shows a set of recommended ranges for each of the three parameter types. It should also be noted that the number of binary digits can be varied to increase or decrease the resolution of the CAF search space. For example, a 5-digit binary encoding divides the search interval into $2^5 = 32$ values, while a 16-digit binary encoding divides the interval into 65,536 values.

Table 4.1 Recommended ranges for the capacity calibration parameters.

Parameter	Minimum	Maximum
Pre-breakdown CAF	0.80	1.05
Capacity Drop Factor (%)	0	10
Jam Density (veh/mi/ln)	180	220

With a binary representation for each parameter defined, the capacity parameters next need to be incorporated into the GA framework. This requires adding the capacity drop factor to the existing demand calibration process, and creating an additional process for estimating the pre-breakdown capacity adjustments and the facility-wide jam density. Since the demand calibration piece uses uniform crossover and mutation, both of which consider individual binary digits independently, the capacity drop factor can be calibrated in conjunction with demand by simply appending its binary representation to the encoded string of demand values or adjustments. On the other hand, since the framework now functions as a two-stage process, an additional encoding must be utilized for the set of pre-breakdown CAFs and the jam density.

In a freeway facilities analysis, each individual segment of a facility has a representative capacity value and a corresponding CAF. This research assumes that capacity of a segment is fixed over time, and thus the number of CAF decision variables considered is equal to the number of segments. The capacity drop factor is considered to be a facility-wide value, and hence constitutes one decision variable. For a facility with NS segments, the set of NS CAF binary encodings are then concatenated into a single binary string. This provides a simpler (albeit analogous) representation as opposed

to working with an array of binary strings. Additionally, the binary encoding of the jam density can simply be appended to the end of the single string representation, without altering how GA operators approach it beyond extending its length. Figure 4.6 demonstrates the process of encoding a set of three pre-breakdown capacity adjustments and the facility-wide jam density into a single binary string.

Table 4.2 Parameter minimum values and step sizes for the encoding example.

Parameter	Minimum Value	Step Size
Pre-breakdown CAF	0.80	0.01
Jam Density (veh/mi/ln)	180	1

	Pre-breakdown Capacity Adjustments			Jam Density
	Seg. 1	Seg. 2	Seg. 3	-
Actual Value	1.0	0.95	1.0	205
# Steps Above Minimum	20	15	20	25
5 Digit Binary Encoding	10100	01111	10100	11001
Final Encoded String	10100011111010011001			

Figure 4.6 Example of encoding three pre-breakdown CAFs and the facility-wide jam density into binary strings based on the values of Table 4.2.

As a final note on pre-breakdown capacity adjustment factors, this approach initially extends their use to all segments, as opposed to just at bottleneck segments as proposed in the HCM's calibration procedure. The reason for this is two-fold. First, this enables the framework to make small adjustments to segment capacities along the entire facility as the default estimates may be slightly off. However, and more importantly, the second reason is that there is uncertainty in the location of a bottleneck, and its location is likely relatively unknown for an uncalibrated facility. By allowing the GA metaheuristic to adjust the CAFs for every segment, it gives it the freedom to identify bottleneck location(s) while also determining the magnitude of the capacity adjustment(s). Further, while the GA may in some instances adjust capacities when it has no effect on performance measures

(e.g., when demand is well below capacity), a simple post-processing check can be employed to replace these adjustments with an unadjusted value of 1.0.

4.8 Selection and Crossover

The primary drivers of the evolutionary process of a GA are the selection and crossover operators. The selection operator is used to choose the two organisms (candidate parameter adjustments in our case) that will undergo crossover in order to produce new offspring organisms. There are a number of ways to implement the selection process including random selection, roulette wheel selection, and tournament selection. This research makes use of a simple k -tournament selection strategy in which k solutions are randomly picked from the current generation, and the one with the best (lowest) fitness is selected to be a parent. This process is repeated twice to determine the two parent organisms to undergo crossover. The value of k can be varied to improve the convergence rate of the GA.

As with selection, there are a variety of crossover strategies and a vast amount of research conducted in regards to how these strategies impact final solution quality and the speed of convergence. Three commonly used crossover strategies were implemented and tested as part of this research. These included a one and two point crossovers, as well as uniform crossover. Figure 4.7 demonstrates each of the three crossover strategies. However, initial testing indicated that the uniform crossover strategy was by far the most effective. Using this strategy, a random number is generated uniformly between 0 and 1 for each binary digit of the encoding. If the generated value is less than a pre-specified mixing percentage (e.g. 0.5 would indicate 50% mixing), the binary digit is swapped between the two strings. As such, for the case study of the subsequent section, a uniform crossover strategy was always used for the Phase 2 capacity calibration process.

4.9 Mutation

A uniform mutation strategy was implemented for use in the bottleneck capacity calibration GA framework to help prevent premature convergence to a locally optimal solution. As its name would suggest, a uniform mutation strategy is very similar to the previously described uniform crossover strategy. However, unlike crossover, uniform mutation operates just on a single set of genes in order to randomly introduce changes such that a diverse population of candidate solutions is maintained. Further, instead of a mixing percentage hovering around 50%, mutation of individual genes (e.g., binary digits) is generally specified to occur at a much lower frequency, known as the mutation rate. During the mutation process for a binary string encoding, a random number is again generated

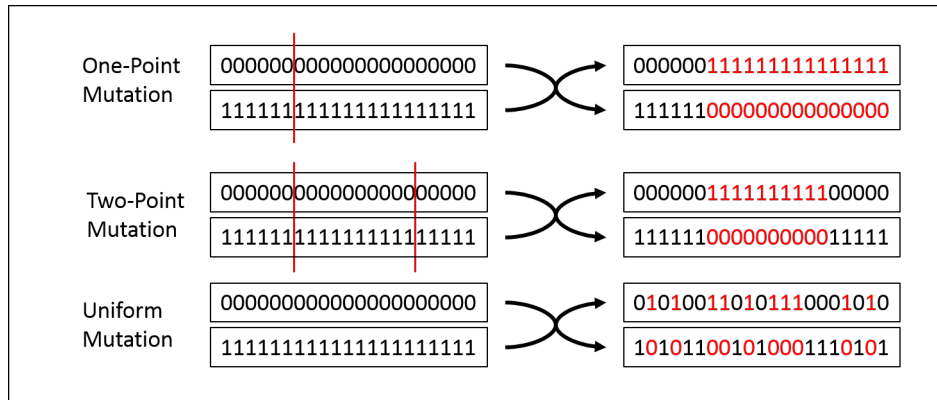


Figure 4.7 Demonstration of three common crossover operators.

uniformly between 0 and 1 for each binary digit. If the number is less than the specified mutation rate, the corresponding binary digit is flipped to its binary alternative. Specifically, a 0 becomes a 1, and a 1 becomes a 0. The optimal mutation rate varies for each specific problem, but research has generally shown that for binary GA encodings, a rate of 0.02 to 0.05 has the most beneficial effect on convergence and final solution quality (Mitchell, 1998).

4.10 Computational Experiments and Numerical Results

A case study for testing the approach with real word data was developed for analysis based on a section of I-540 outside of Raleigh, NC. The goal of the case study is to produce a calibrated model by estimating the demand volume inputs and adjusting the core capacity parameters using the genetic algorithm framework. The facility consists of 14.5 miles of largely three-lane urban freeway in the westbound direction, including seven on-ramp segments and eight off-ramp segments. The facility as modeled using the HCM methodology is shown in Figure 4.8. The facility was then populated with initial demand input values based on AADT estimates available through the North Carolina Department of Transportation and combined with a best estimate of the local hourly demand behavior for the study period (North Carolina Department of Transportation, 2016).

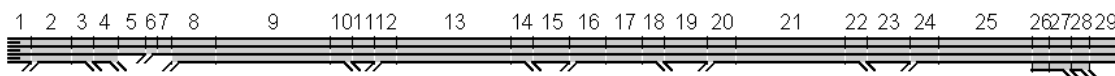


Figure 4.8 HCM segmentation of the 14.5 mile section of I-540 westbound near Raleigh, NC.

A set of representative speeds for a five-and-a-half-hour study period between 6:00am and 11:30 am on a Tuesday in August 2014 was collected from RITIS.org and are shown in the top contour of Figure 4.9 (Maryland CATT Lab, 2008). The bottom contour of Figure 4.9 shows the speed contours initially predicted by the HCM methodology, which clearly has a very different congestion regime as opposed to the target data. The analysis periods in which the congestion occurs are similar, but the target data shows low speeds in segments 8 to 23, while the uncalibrated facility shows congestion only in segments 2 to 6. A summary of the average speed error for different speed regimes is given in Table 4.3. A list facility travel times based on the target travel speeds for each analysis period can be found in Table 4.4. The table also shows the travel times based on the predicted speeds of the uncalibrated facility. It is interesting to note that the travel times are surprisingly similar to the target travel times, though this is unlikely to be the case for most facilities with such clear differences between target and uncalibrated speeds.

Table 4.3 Summary of the average absolute speed errors for the uncalibrated facility.

Error	All Speeds	$v_{i,p} < 65$	$v_{i,p} < 55$	$v_{i,p} < 45$	$v_{i,p} < 35$
Uncalibrated	6.12 mph	12.91 mph	17.21 mph	21.83 mph	36.86 mph

Based on the large differences seen in the speed contours, it is clear that this facility must be calibrated before it can be used for any study or decision-making analysis. Despite having predicted travel times not far off from the target values, it is still desirable to run the model through the demand calibration process (Phase 1) due to the clear misrepresentation of congestion seen in the speeds. The following two sections present the results of applying the two stage calibration process.

Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29
#1 6:00 - 6:15	65.4	72.6	72.6	72.6	73.0	73.0	73.0	73.0	73.0	73.0	73.4	74.1	74.1	74.1	73.6	73.7	73.7	73.7	72.4	70.5	70.5	70.5	71.8	71.1	71.1	71.1	71.1	72.8	72.8
#2 6:15 - 6:30	68.9	70.4	70.4	70.4	72.5	72.5	72.5	72.1	72.1	72.1	73.3	73.9	73.9	73.9	74.0	74.3	74.3	74.3	74.1	73.4	73.4	73.4	74.5	74.5	74.5	74.5	74.5	72.7	72.7
#3 6:30 - 6:45	73.2	70.8	70.8	70.8	71.1	71.1	71.1	71.5	71.5	71.5	73.1	72.5	72.5	72.5	73.5	73.1	73.1	73.1	72.5	72.0	72.0	72.0	71.7	73.7	73.7	73.7	73.7	72.9	72.9
#4 6:45 - 7:00	72.5	72.6	72.6	72.6	71.3	71.3	71.3	71.6	71.6	71.6	70.8	71.0	71.0	71.0	72.4	71.7	71.7	71.7	71.7	70.8	70.8	70.8	72.0	72.9	72.9	72.9	72.9	73.5	73.5
#5 7:00 - 7:15	73.7	70.3	70.3	70.3	70.7	70.7	70.7	71.9	71.9	71.9	73.3	71.9	71.9	71.9	72.5	72.5	72.5	72.5	72.5	70.8	70.8	70.8	71.7	72.1	72.1	72.1	72.1	73.7	73.7
#6 7:15 - 7:30	68.6	67.6	67.6	67.6	68.4	68.4	68.4	68.4	68.4	68.4	67.4	67.3	67.3	67.3	69.4	68.3	68.3	68.3	69.8	69.1	69.1	69.1	71.7	72.1	72.1	72.1	72.1	71.7	71.7
#7 7:30 - 7:45	67.9	64.7	64.7	64.7	68.6	68.6	68.6	60.9	60.9	60.9	46.5	48.8	48.8	48.8	63.9	62.8	62.8	62.8	59.0	59.1	59.1	59.1	66.4	70.1	70.1	70.1	70.1	70.9	70.9
#8 7:45 - 8:00	71.3	68.7	68.7	68.7	67.8	67.8	67.8	53.9	53.9	53.9	31.5	38.7	38.7	38.7	45.3	46.4	46.4	46.4	36.0	53.9	53.9	53.9	64.8	68.5	68.5	68.5	68.5	68.5	68.5
#9 8:00 - 8:15	71.3	68.1	68.1	68.1	67.1	67.1	67.1	56.1	56.1	56.1	36.8	41.0	41.0	41.0	36.9	43.9	43.9	43.9	44.2	55.1	55.1	55.1	62.3	67.6	67.6	67.6	67.6	70.2	70.2
#10 8:15 - 8:30	72.7	68.7	68.7	68.7	67.5	67.5	67.5	67.3	67.3	67.3	55.9	43.0	43.0	43.0	35.8	43.8	43.8	43.8	42.2	56.9	56.9	56.9	64.6	69.1	69.1	69.1	69.1	71.1	71.1
#11 8:30 - 8:45	70.3	70.8	70.8	70.8	73.0	73.0	73.0	70.5	70.5	70.5	64.3	59.0	59.0	59.0	57.7	54.9	54.9	54.9	46.7	57.0	57.0	57.0	63.8	69.5	69.5	69.5	69.5	70.2	70.2
#12 8:45 - 9:00	71.8	70.9	70.9	70.9	71.1	71.1	71.1	70.1	70.1	70.1	70.3	70.0	70.0	70.0	63.4	54.9	54.9	54.9	57.8	62.4	62.4	62.4	65.7	68.1	68.1	68.1	68.1	70.0	70.0
#13 9:00 - 9:15	73.3	70.9	70.9	70.9	69.7	69.7	69.7	70.3	70.3	70.3	72.9	72.9	72.9	72.9	72.6	73.0	73.0	73.0	73.1	71.2	71.2	71.2	72.5	72.2	72.2	72.2	72.2	73.2	73.2
#14 9:15 - 9:30	73.8	69.8	69.8	69.8	71.3	71.3	71.3	73.6	73.6	73.6	73.3	72.6	72.6	72.6	72.5	73.0	73.0	73.0	72.6	72.3	72.3	72.3	74.4	73.3	73.3	73.3	73.3	73.9	73.9
#15 9:30 - 9:45	70.1	68.5	68.5	68.5	68.3	68.3	68.3	69.8	69.8	69.8	71.4	72.7	72.7	72.7	73.1	73.4	73.4	73.4	73.9	71.7	71.7	71.7	72.1	72.2	72.2	72.2	72.2	73.0	73.0
#16 9:45 - 10:00	67.2	68.7	68.7	68.7	68.2	68.2	68.2	68.1	68.1	68.1	68.9	70.1	70.1	70.1	71.0	71.1	71.1	71.1	71.1	70.5	70.5	70.5	71.9	71.3	71.3	71.3	71.3	72.9	72.9
#17 10:00 - 10:15	69.8	69.1	69.1	69.1	67.4	67.4	67.4	67.7	67.7	67.7	68.1	69.7	69.7	69.7	69.0	70.7	70.7	70.7	71.4	70.7	70.7	70.7	72.7	72.3	72.3	72.3	72.6	72.6	72.6
#18 10:15 - 10:30	72.5	66.9	66.9	66.9	69.1	69.1	69.1	72.7	72.7	72.7	74.4	74.0	74.0	74.0	72.9	72.1	72.1	72.1	72.3	71.1	71.1	71.1	73.5	74.2	74.2	74.2	74.2	73.9	73.9
#19 10:30 - 10:45	69.9	69.8	69.8	69.8	70.9	70.9	70.9	70.5	70.5	70.5	70.7	71.4	71.4	71.4	71.1	71.6	71.6	71.6	73.7	73.3	73.3	73.3	73.3	73.6	73.6	73.6	73.6	73.9	73.9
#20 10:45 - 11:00	69.0	67.0	67.0	67.0	68.0	68.0	68.0	68.3	68.3	68.3	69.4	69.7	69.7	69.7	69.6	70.5	70.5	70.5	71.4	70.8	70.8	70.8	71.5	71.8	71.8	71.8	71.8	73.1	73.1
#21 11:00 - 11:15	68.7	70.4	70.4	70.4	67.8	67.8	67.8	68.0	68.0	68.0	69.5	70.3	70.3	70.3	70.5	73.5	73.5	73.5	74.1	72.0	72.0	72.0	71.1	71.5	71.5	71.5	71.5	72.5	72.5
#22 11:15 - 11:30	68.0	70.3	70.3	70.3	67.7	67.7	67.7	70.4	70.4	70.4	68.1	70.2	70.2	70.2	73.2	73.5	73.5	73.5	72.9	72.2	72.2	72.2	71.4	72.4	72.4	72.4	72.4	72.7	72.7

Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29
#1 6:00 - 6:15	75.0	71.0	69.4	74.6	71.9	63.3	70.9	71.0	73.5	68.8	74.3	66.9	72.0	69.4	73.5	69.1	71.6	68.5	73.2	67.7	71.7	68.6	73.1	67.9	71.5	74.7	70.4	74.5	74.4
#2 6:15 - 6:30	74.6	70.4	69.3	73.7	69.1	61.6	67.6	70.2	71.6	68.6	73.1	65.6	69.2	68.9	71.6	67.7	68.5	68.3	71.1	66.6	68.7	68.5	71.0	66.7	68.4	73.8	70.0	74.5	73.3
#3 6:30 - 6:45	73.9	69.8	69.2	72.3	65.4	59.0	63.3	69.0	69.0	68.5	71.3	64.1	65.6	65.6	69.1	64.6	64.6	64.6	68.4	64.8	64.8	64.8	68.2	64.4	64.4	72.4	69.6	74.4	71.6
#4 6:45 - 7:00	72.7	69.0	69.1	70.3	60.8	55.1	57.9	65.7	65.7	68.9	61.0	61.0	61.0	65.7	59.7	59.7	59.7	64.8	60.0	60.0	60.0	64.5	59.4	59.4	70.5	69.2	73.6	69.3	
#5 7:00 - 7:15	71.1	66.4	66.9	29.7	35.9	54.1	56.8	64.3	64.3	64.3	67.8	58.5	58.5	58.5	63.9	56.6	56.6	56.6	62.5	56.7	56.7	56.7	62.0	55.8	55.8	69.0	69.0	72.9	67.5
#6 7:15 - 7:30	71.6	62.5	24.6	18.4	35.1	54.2	56.9	64.6	64.6	64.6	68.0	59.1	59.1	59.1	64.4	57.5	57.5	57.5	63.2	57.6	57.6	57.6	62.7	56.9	56.9	69.4	69.0	73.1	68.1
#7 7:30 - 7:45	66.3	36.3	20.8	18.4	35.3	54.2	56.9	64.9	64.9	64.9	68.2	59.7	59.7	59.7	64.8	58.2	58.2	58.2	63.7	58.4	58.4	58.4	63.3	57.7	57.7	69.8	69.1	73.3	68.5
#8 7:45 - 8:00	61.2	34.4	20.8	18.4	35.5	54.2	56.9	65.1	65.1	65.1	68.4	60.2	60.2	60.2	65.2	58.9	58.9	58.9	64.2	59.2	59.2	59.2	64.0	58.7	58.7	70.2	69.1	73.5	68.9
#9 8:00 - 8:15	67.3	51.7	21.3	18.5	35.6	54.2	56.9	65.3	65.3	68.6	60.8	60.8	60.8	65.6	59.5	59.5	59.5	64.7	60.0	60.0	60.0	64.5	59.5	59.5	70.5	69.2	73.6	69.3	
#10 8:15 - 8:30	73.8	69.3	54.2	26.6	36.4	54.2	56.9	65.9	65.9	65.9	69.0	62.0	62.0	62.0	66.5	61.2	61.2	61.2	65.9	61.8	61.8	61.8	65.9	61.5	61.5	71.3	69.4	74.0	70.3
#11 8:30 - 8:45	74.4	70.2	69.3	73.1	67.3	60.2	65.2	69.6	70.3	68.6	72.2	64.9	67.4	67.4	70.3	66.6	66.6	66.6	69.8	65.8	66.8	66.8	69.6	65.8	66.4	73.1	69.8	74.5	72.5
#12 8:45 - 9:00	74.8	70.6	69.4	74.0	69.8	62.0	68.4	70.4	72.1	68.7	73.4	66.0	69.9	69.1	72.1	68.0	69.3	68.4	71.7	66.9	69.4	68.5	71.6	67.0	69.1	74.0	70.1	74.5	73.6
#13 9:00 - 9:15	75.0	70.9	69.4	74.6	71.7	63.1	70.6	70.9	73.3	68.8	74.2	66.8	71.7	69.4	73.3	68.9	71.3	68.5	73.0	67.6	71.4	68.6	72.9	67.8	71.2	74.6	70.4	74.5	74.3
#14 9:15 - 9:30	75.0	71.1	69.4	74.6	72.4	63.5	71.5	71.1	73.7	68.8	74.5	67.1	72.4	69.5	73.8	69.3	72.0	68.5	73.5	67.9	72.1	68.7	73.4	68.1	71.9	74.8	70.5	74.5	74.6
#15 9:30 - 9:45	75.0	71.2	69.4	74.6	73.0	63.9	72.2	71.3	74.1	68.8	74.5	67.3	73.0	69.6	74.1	69.6	72.7	68.6	73.9	68.2	72.7	68.7	73.8	68.4	72.6	74.9	70.6	74.5	74.7
#16 9:45 - 10:00	75.0	71.3	69.4	74.6	73.5	64.2	72.6	71.5	74.4	68.8	74.5	67.6	73.5	69.8	74.4	69.8	73.2	68.6	74.2	68.4	73.3	68.8	74.2	68.6	73.2	75.0	70.7	74.5	74.9
#17 10:00 - 10:15	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	74.0	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.8	68.8	74.5	68.9	73.7	75.0	70.8	74.6	75.0
#18 10:15 - 10:30	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	73.9	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.8	68.8	74.5	68.8	73.7	75.0	70.8	74.6	75.0
#19 10:30 - 10:45	75.0	71.5	69.4	74.6	73.9	64.4	72.7	71.6	74.6	68.9	74.5	67.8	73.9	69.9	74.6	70.1	73.7	68.7	74.5	68.6	73.7	68.8	74.5	68.8	73.7	75.0	70.8	74.6	75.0
#20 10:45 - 11:00	75.0	71.5	69.4	74.6	73.9	64.4	72.7	7																					

Table 4.4 Facility travel times based on the target speeds compared to those from the uncalibrated facility.

Analysis Period	Target Facility Travel Time (minutes)	Uncalibrated Travel Time (minutes)	Difference (minutes)	Percent (Difference)
#1 6:00 - 6:15	11.98	12.12	0.14	1.13%
#2 6:15 - 6:30	11.85	12.43	0.59	4.95%
#3 6:30 - 6:45	11.98	12.93	0.95	7.90%
#4 6:45 - 7:00	12.07	13.71	1.64	13.58%
#5 7:00 - 7:15	12.07	14.86	2.80	23.18%
#6 7:15 - 7:30	12.54	15.61	3.07	24.53%
#7 7:30 - 7:45	14.26	16.00	1.74	12.17%
#8 7:45 - 8:00	16.56	15.95	-0.61	-3.68%
#9 8:00 - 8:15	16.27	15.49	-0.77	-4.76%
#10 8:15 - 8:30	15.45	14.25	-1.20	-7.78%
#11 8:30 - 8:45	13.76	12.67	-1.09	-7.94%
#12 8:45 - 9:00	13.08	12.35	-0.73	-5.59%
#13 9:00 - 9:15	12.07	12.15	0.08	0.65%
#14 9:15 - 9:30	11.92	12.07	0.16	1.32%
#15 9:30 - 9:45	12.13	12.01	-0.11	-0.93%
#16 9:45 - 10:00	12.37	11.96	-0.41	-3.30%
#17 10:00 - 10:15	12.37	11.92	-0.45	-3.66%
#18 10:15 - 10:30	12.00	11.92	-0.07	-0.62%
#19 10:30 - 10:45	12.05	11.92	-0.13	-1.08%
#20 10:45 - 11:00	12.39	11.92	-0.47	-3.76%
#21 11:00 - 11:15	12.26	11.93	-0.33	-2.72%
#22 11:15 - 11:30	12.16	11.93	-0.23	-1.90%
Average	12.89	13.10	0.21	1.60%
Avg Absolute Error	-	-	0.81	6.27%

4.11 Phase 1 Calibration Results

The phase 1 genetic algorithm approach was applied in order to estimate a set of demand volumes for the first calibration stage. Since the uncalibrated speed contours were significantly different than the target speeds, the n-distribution approach was used in conjunction with the known AADT estimates to determine a new set of demand profiles for the facility entry and exit points. With the new unified approach, the capacity drop factor was considered in conjunction with the demand estimation.

Based on the size of the facility and the length of the study period, an initial population size of 200 organisms was specified. Additionally, the recommended 2:1 ratio of weights for λ_1 and λ_2 was used, where the facility travel time was given twice as much weight as the individual segment speeds. Table 4.5 gives a summary of the parameters used to conduct the Phase 1 calibration, including the parameters controlling the estimation of the new demand profiles. An explanation of these distribution parameters can be found in the previous chapter concerning the demand estimation framework.

Table 4.5 Summary of the parameters used for the Phase 1 calibration.

Phase 1 Calibration and GA Run Parameters	Value
Population Size	200
Max Iterations	400
Binary Encoding Digits	5
Crossover Mixing %	50%
Mutation Rate	5%
Capacity Drop %	0%-10%
Demand Distribution Parameters	Range
Random Variable Weight (w_i)	0.05,0.5
Mean 1 (μ_1)	7am - 10am
Mean 1 (μ_1)	Unrestricted
Mean 1 (μ_1)	4pm - 7pm
Std. Deviation (σ_i)	0.5, 5.0
Skewness (α_i)	-10,10

A run of the Phase 1 calibration was successfully able to estimate new demand profiles for the model as well as estimate that the capacity drop factor is 5%. A summary of the average absolute speed errors for different speed regimes can be found in Table 4.6. Additionally, Figure 4.10 shows a comparison of the target speed contour and the new partially calibrated speed contour, while Table 4.7 provides a full breakdown of the remaining error between the target and predicted travel times. It is clear from both the summary table and the speed contours that there was a significant reduction in error for the individual segment speeds. This came despite the error in travel time being weighted significantly more heavily. The overall reduction of error between target and predicted speeds was

around 36%, and for the lowest speed regimes, the reduction in error was around 80%!

From Table 4.7 it can be seen that there was an increase in the error in travel time from the uncalibrated model, but this is only due to the uncalibrated model having an unnaturally close travel time estimate, and cannot be viewed as a negative since the uncalibrated travel time was only “good” when considered independently from the clearly incorrect predicted speeds. It is also important to note that the average predicted travel time is only 3.38% off from the average target travel time. This is well within the recommended accuracy of 10% recommended in the HCM’s current calibration procedure (*Highway Capacity Manual 2016*).

Table 4.6 Summary results for the Phase 1 calibration run. All errors are the average absolute difference in speed.

Error	All Speeds	$v_{i,p} < 65$	$v_{i,p} < 55$	$v_{i,p} < 45$	$v_{i,p} < 35$
Uncalibrated	6.12 mph	12.91 mph	17.21 mph	21.83 mph	36.86 mph
After Phase 1	4.15 mph	7.24 mph	6.55 mph	4.19 mph	7.54 mph
% Improvement	32.09%	43.86%	61.97%	80.78%	79.53%

Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29
#1 6:00 - 6:15	65.4	72.6	72.6	72.6	73.0	73.0	73.0	73.0	73.0	73.0	73.4	74.1	74.1	74.1	73.6	73.7	73.7	73.7	72.4	70.5	70.5	70.5	71.8	71.1	71.1	71.1	71.1	72.8	72.8
#2 6:15 - 6:30	68.9	70.4	70.4	70.4	72.5	72.5	72.5	72.1	72.1	73.3	73.9	73.9	73.9	74.0	74.3	74.3	74.3	74.1	73.4	73.4	73.4	74.5	74.5	74.5	74.5	74.5	72.7	72.7	
#3 6:30 - 6:45	73.2	70.8	70.8	70.8	71.1	71.1	71.1	71.5	71.5	71.5	73.1	72.5	72.5	72.5	73.5	73.1	73.1	73.1	72.5	72.0	72.0	72.0	71.7	73.7	73.7	73.7	72.9	72.9	
#4 6:45 - 7:00	72.5	72.6	72.6	72.6	71.3	71.3	71.3	71.6	71.6	71.6	70.8	71.0	71.0	71.0	72.4	71.7	71.7	71.7	71.7	70.8	70.8	70.8	72.0	72.9	72.9	72.9	72.9	73.5	73.5
#5 7:00 - 7:15	73.7	70.3	70.3	70.3	70.7	70.7	70.7	71.9	71.9	71.9	73.3	71.9	71.9	71.9	72.5	72.5	72.5	72.5	72.5	70.8	70.8	70.8	71.7	72.1	72.1	72.1	72.1	73.7	73.7
#6 7:15 - 7:30	68.6	67.6	67.6	67.6	68.4	68.4	68.4	68.4	68.4	68.4	67.4	67.3	67.3	67.3	69.4	68.3	68.3	68.3	69.8	69.1	69.1	69.1	71.7	72.1	72.1	72.1	71.7	71.7	
#7 7:30 - 7:45	67.9	64.7	64.7	64.7	68.6	68.6	68.6	60.9	60.9	60.9	46.5	48.8	48.8	48.8	63.9	62.8	62.8	62.8	59.0	59.1	59.1	59.1	66.4	70.1	70.1	70.1	70.1	70.9	70.9
#8 7:45 - 8:00	71.3	68.7	68.7	68.7	67.8	67.8	67.8	53.9	53.9	53.9	31.5	38.7	38.7	38.7	45.3	46.4	46.4	46.4	36.0	53.9	53.9	53.9	64.8	68.5	68.5	68.5	68.5	68.5	68.5
#9 8:00 - 8:15	71.3	68.1	68.1	68.1	67.1	67.1	67.1	56.1	56.1	56.1	36.8	41.0	41.0	41.0	36.9	43.9	43.9	43.9	44.2	55.1	55.1	55.1	62.3	67.6	67.6	67.6	67.6	70.2	70.2
#10 8:15 - 8:30	72.7	68.7	68.7	68.7	67.5	67.5	67.5	67.3	67.3	67.3	55.9	43.0	43.0	43.0	35.8	43.8	43.8	43.8	42.2	56.9	56.9	56.9	64.6	69.1	69.1	69.1	69.1	71.1	71.1
#11 8:30 - 8:45	70.3	70.8	70.8	70.8	73.0	73.0	73.0	70.5	70.5	70.5	64.3	59.0	59.0	59.0	57.7	54.9	54.9	54.9	46.7	57.0	57.0	57.0	63.8	69.5	69.5	69.5	69.5	70.2	70.2
#12 8:45 - 9:00	71.8	70.9	70.9	70.9	71.1	71.1	71.1	70.1	70.1	70.1	70.3	70.0	70.0	70.0	63.4	54.9	54.9	54.9	57.8	62.4	62.4	62.4	65.7	68.1	68.1	68.1	68.1	70.0	70.0
#13 9:00 - 9:15	73.3	70.9	70.9	70.9	69.7	69.7	69.7	70.3	70.3	70.3	72.9	72.9	72.9	72.9	72.6	73.0	73.0	73.0	73.1	71.2	71.2	71.2	72.5	72.2	72.2	72.2	73.2	73.2	
#14 9:15 - 9:30	73.8	69.8	69.8	69.8	71.3	71.3	71.3	73.6	73.6	73.6	73.3	72.6	72.6	72.6	72.5	73.0	73.0	73.0	72.6	72.3	72.3	72.3	74.4	73.3	73.3	73.3	73.3	73.9	73.9
#15 9:30 - 9:45	70.1	68.5	68.5	68.5	68.3	68.3	68.3	69.8	69.8	69.8	68.1	71.4	72.7	72.7	72.7	73.1	73.4	73.4	73.4	73.9	71.7	71.7	71.7	72.1	72.2	72.2	72.2	73.0	73.0
#16 9:45 - 10:00	67.2	68.7	68.7	68.7	68.2	68.2	68.2	68.1	68.1	68.1	68.9	70.1	70.1	70.1	71.0	71.1	71.1	71.1	71.1	70.5	70.5	70.5	71.9	71.3	71.3	71.3	72.9	72.9	
#17 10:00 - 10:15	69.8	69.1	69.1	69.1	67.4	67.4	67.4	67.7	67.7	67.7	68.1	69.7	69.7	69.7	69.0	70.7	70.7	70.7	71.4	70.7	70.7	70.7	72.7	72.3	72.3	72.3	72.3	72.6	72.6
#18 10:15 - 10:30	72.5	66.9	66.9	66.9	69.1	69.1	69.1	72.7	72.7	72.7	74.4	74.0	74.0	74.0	72.9	72.1	72.1	72.1	72.3	71.1	71.1	71.1	73.5	74.2	74.2	74.2	73.9	73.9	
#19 10:30 - 10:45	69.9	69.8	69.8	69.8	70.9	70.9	70.9	70.5	70.5	70.5	70.7	71.4	71.4	71.4	71.1	71.6	71.6	71.6	73.7	73.3	73.3	73.3	73.6	73.6	73.6	73.6	73.5	73.5	
#20 10:45 - 11:00	69.0	67.0	67.0	67.0	68.0	68.0	68.0	68.3	68.3	68.3	69.4	69.7	69.7	69.7	69.6	70.5	70.5	70.5	71.4	70.8	70.8	70.8	71.5	71.8	71.8	71.8	73.1	73.1	
#21 11:00 - 11:15	68.7	70.4	70.4	70.4	67.8	67.8	67.8	68.0	68.0	68.0	69.5	70.3	70.3	70.3	70.5	73.5	73.5	73.5	74.1	72.0	72.0	72.0	71.1	71.5	71.5	71.5	72.5	72.5	
#22 11:15 - 11:30	68.0	70.3	70.3	70.3	67.7	67.7	67.7	70.4	70.4	70.4	68.1	70.2	70.2	70.2	73.2	73.5	73.5	73.5	72.9	72.2	72.2	72.2	71.4	72.4	72.4	72.4	72.4	72.7	72.7
Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29
#1 6:00 - 6:15	75.0	72.1	69.3	74.6	75.0	65.7	73.0	72.2	75.0	68.6	74.4	67.9	74.6	70.5	74.8	70.0	73.7	69.4	74.1	68.4	73.3	68.8	74.1	68.7	73.3	75.0	71.3	74.6	74.7
#2 6:15 - 6:30	75.0	71.8	69.4	74.6	75.0	65.5	72.9	72.0	75.0	68.6	74.5	67.5	74.1	70.3	74.5	69.7	73.1	69.4	73.5	68.2	72.6	68.7	73.8	68.5	72.9	74.9	71.5	74.6	74.4
#3 6:30 - 6:45	75.0	71.5	69.6	74.6	74.7	65.0	72.8	71.7	74.7	68.8	74.5	66.7	72.7	70.0	73.3	68.8	71.1	69.3	71.7	67.3	70.3	68.5	72.2	67.8	70.9	74.5	71.4	74.6	73.3
#4 6:45 - 7:00	75.0	71.0	69.6	74.6	73.1	64.0	72.4	71.1	73.8	68.8	74.5	65.3	70.0	69.7	70.9	67.3	67.7	67.7	68.5	65.9	66.5	66.5	69.6	66.7	67.9	73.6	71.3	74.5	71.1
#5 7:00 - 7:15	74.5	70.4	69.6	73.4	69.6	62.0	68.3	70.3	71.9	68.8	73.1	62.9	65.5	65.5	66.8	62.3	62.3	62.3	63.3	60.7	60.7	60.7	65.4	63.0	63.0	71.9	71.2	73.2	67.2
#6 7:15 - 7:30	73.3	69.7	69.5	71.6	64.6	58.7	62.8	69.5	69.7	68.7	71.4	60.1	61.0	61.0	62.6	57.1	57.1	57.1	58.3	55.0	55.0	55.0	61.4	58.4	58.4	70.1	70.1	71.7	63.4
#7 7:30 - 7:45	72.1	69.2	69.5	70.0	60.2	55.1	58.0	67.6	67.6	67.6	69.5	56.1	49.5	42.5	32.7	56.8	50.5	54.4	38.5	56.8	56.8	56.8	63.0	60.3	60.3	70.8	70.8	72.3	64.7
#8 7:45 - 8:00	71.8	69.0	69.1	69.4	57.9	53.0	55.5	66.1	59.6	37.1	24.0	40.0	40.1	34.9	30.2	49.4	48.0	41.2	35.7	56.9	56.9	56.9	63.3	60.6	60.6	71.0	71.0	72.3	64.7
#9 8:00 - 8:15	72.6	69.3	69.5	70.5	60.3	55.2	58.1	67.3	48.1	31.1	20.4	37.1	40.7	35.8	29.5	48.0	45.8	48.5	36.7	56.9	56.9	56.9	63.7	61.1	61.1	71.1	71.1	72.4	65.1
#10 8:15 - 8:30	73.9	70.0	69.6	72.3	65.0	58.9	63.1	69.4	64.4	44.4	22.2	37.0	42.0	44.0	32.6	47.3	45.4	40.3	35.3	56.9	56.9	56.9	63.9	61.2	61.2	71.2	71.2	72.5	65.3
#11 8:30 - 8:45	74.8	70.6	69.6	73.8	69.3	61.8	67.9	70.2	71.8	68.8	66.1	56.7	47.7	39.7	31.3	49.3	42.1	37.6	35.2	56.9	56.9	56.9	64.1	61.5	61.5	71.3	71.2	72.6	65.7
#12 8:45 - 9:00	75.0	71.2	69.6	74.6	72.5	63.6	71.6	71.0	73.6	68.8	74.4	60.9	61.5	62.9	43.7	48.0	41.0	39.5	35.3	56.9	56.9	56.9	64.2	61.5	61.5	71.3	71.1	72.7	66.0
#13 9:00 - 9:15	75.0	71.6	69.5	74.6	74.3	64.6	72.7	71.5	74.5	68.8	74.5	62.7	68.1	68.1	69.4	68.4	62.7	62.3	48.3	59.8	59.8	59.8	66.1	63.6	63.6	72.1	71.2	73.3	67.6
#14 9:15 - 9:30	75.0	71.8	69.4	74.6	74.9	65.2	72.8	71.8	74.9	68.7	74.5	63.6	69.8	69.5	71.0	67.2	67.6	67.6	68.3	65.7	66.0	66.0	70.1	66.8	68.2	73.7	71.5	74.5	71.1
#15 9:30 - 9:45	75.0	72.0	69.2	74.6	75.0	65.5	72.9	72.1	75.0	68.6	74.4	64.5	71.2	69.7	72.3	68.0	69.4	69.3	70.0	66.4	68.0	68.0	71.3	67.3	69.6	74.2	71.6	74.6	72.2
#16 9:45 - 10:00	75.0	72.2	69.1	74.6	75.0	65.7	73.0	72.2	75.0	68.4	74.4	65.0	72.1	69.7	73.1	68.6	69.6	69.4	71.2	67.0	69.4	68.1	72.1	67.7	70.7	74.5	71.5	74.6	73.0
#17 10:00 - 10:15	75.0	72.3	69.1	74.6	75.0	65.8	73.0	72.4	75.0	68.3	74.4	65.3	72.7	69.8	73.6	69.0	71.5	69.4	72.1	67.3	70.5	68.2	72.7	68.0	71.4	74.7	71.7	74.6	73.4
#18 10:15 - 10:30	75.0	72.3	69.0	74.6	75.0	65.9	73.0	72.4	75.0	68.3	74.4	65.6	73.1	69.8	74.0	69.2	72.0	69.4	72.5	67.5	71.0	68.3	73.1	68.1	71.8	74.7	71.6	74.6	73.7
#19 10:30 - 10:45	75.0	72.4	68.9	74.6	75.0	65.9	73.0	72.5	75.0	68.3	74.4	68.0	74.9	70.5	74.8	70.6	74.6	69.5	74.8	68.8	74.2	68.6	74.8	69.2	74.3	75.0	72.0	74.7	74.9
#20 10:45 - 11:00	75.0	72.4	68.9	74.5	75.0	65.9	73.0	72.5	75.0	68.3	74.4	68.2	75.0	70.5	74.8	70.6	74.7	6											

Table 4.7 Facility travel times based on the target speeds compared to those from the facility after Phase 1 calibration.

Analysis Period	Target Facility Travel Time (minutes)	Phase 1 Calibration Travel Time (minutes)	Difference (minutes)	Percent (Difference)
#1 6:00 - 6:15	11.98	11.90	-0.08	-0.67%
#2 6:15 - 6:30	11.85	11.95	0.10	0.87%
#3 6:30 - 6:45	11.98	12.11	0.13	1.05%
#4 6:45 - 7:00	12.07	12.43	0.36	2.95%
#5 7:00 - 7:15	12.07	13.12	1.05	8.72%
#6 7:15 - 7:30	12.54	13.95	1.41	11.27%
#7 7:30 - 7:45	14.26	15.19	0.93	6.51%
#8 7:45 - 8:00	16.56	17.05	0.49	3.01%
#9 8:00 - 8:15	16.27	17.53	1.26	7.78%
#10 8:15 - 8:30	15.45	16.61	1.16	7.51%
#11 8:30 - 8:45	13.76	15.47	1.71	12.43%
#12 8:45 - 9:00	13.08	14.52	1.44	11.04%
#13 9:00 - 9:15	12.07	13.06	0.99	8.23%
#14 9:15 - 9:30	11.92	12.41	0.49	4.13%
#15 9:30 - 9:45	12.13	12.25	0.12	1.00%
#16 9:45 - 10:00	12.37	12.15	-0.22	-1.81%
#17 10:00 - 10:15	12.37	12.08	-0.29	-2.36%
#18 10:15 - 10:30	12.00	12.04	0.04	0.39%
#19 10:30 - 10:45	12.05	11.85	-0.20	-1.72%
#20 10:45 - 11:00	12.39	11.84	-0.55	-4.44%
#21 11:00 - 11:15	12.26	11.83	-0.43	-3.51%
#22 11:15 - 11:30	12.16	11.82	-0.34	-2.85%
Average	12.89	13.32	0.43	3.38%
Avg Absolute Error	-	-	0.62	4.82%

4.12 Phase 2 Calibration Results

Overall, the Phase 1 calibration showed a large improvement over the initial uncalibrated model and reduced many of the largest speed errors. Though there was a small increase in the average facility error, the maximum error for an individual study period was reduced, and the speed contour reflects the target behavior much more faithfully. However, this only represents a partial calibration, and two key aspects of capacity calibration have yet to be addressed. This partially calibrated model can now be run through Phase 2 of the calibration to estimate the set of pre-breakdown capacity adjustments, as well as the adjust the facility wide jam density. A range of 0.85 to 1.05 was specified for the set of pre-breakdown CAFs of each HCM segment. A range of 180 to 220 veh/mi/ln was used for the facility-wide jam density. Based on the size of the facility and the length of the study period, an initial population size of 200 organisms was specified. Table 4.8 summarizes the parameters, and five GA runs were conducted to calibrate the facility.

Table 4.8 Summary of the parameters used for the Phase 2 calibration runs.

Phase 2 Calibration and GA Run Parameters	Value
Population Size	200
Max Iterations	400
Binary Encoding Digits	5
Crossover Mixing %	50%
Mutation Rate	5%
Pre-breakdown CAF Range	0.85 - 1.05
Jam Density Range	180-220 veh/mi/ln

Running Phase 2 calibration successfully estimated a set of pre-breakdown capacity adjustments as well as the facility-wide jam density. The set of pre-breakdown CAFs can be seen in Figure 4.12, and the capacity drop was estimated to be 5%, while the facility-wide jam density was estimated to be 189 veh/mi/ln. A summary of the average absolute speed errors for different speed regimes can be found in Table 4.9. Additionally, Figure 4.11 shows a comparison of the target speed contour and the new partially calibrated speed contour, while Table 4.10 provides a full breakdown of the remaining error between the target and predicted travel times, as well as improvements over the Phase 1 calibration. The Phase 2 calibration show improvement on both the total speed error, as well as for the individual speed regimes. Perhaps most importantly, a model calibrated through the

two stage process showed a reduction in speed error of over 60% for speeds lower than 65mph, and a reduction of 70% for all speeds lower than 55 mph! In terms of travel time, Table 4.10 shows a further reduction in error for most analysis periods. All but one analysis period falls within 10% of the target travel time, and the average error is just 2.1%.

Table 4.9 Summary results for the Phase 1 calibration run. All errors are the average absolute difference in speed.

Error	All Speeds	$v_{i,p} < 65$	$v_{i,p} < 55$	$v_{i,p} < 45$	$v_{i,p} < 35$
Uncalibrated	6.12 mph	12.91 mph	17.21 mph	21.83 mph	36.86 mph
After Phase 1	4.15 mph	7.24 mph	6.55 mph	4.19 mph	7.54 mph
% Improvement	32.09%	43.86%	61.97%	80.78%	79.53%
After Phase 2	3.73 mph	4.85 mph	4.67 mph	3.32 mph	2.62 mph
% Improvement over Phase 1	10.13%	33.05%	28.55%	20.86%	65.28%
% Improvement over uncalibrated	39.05%	62.43%	72.86%	84.79%	92.89%

These results clearly indicate that the model predicts performance far more consistent with the real world performance measures after the two-stage calibration process. While the predicted speeds do not match the real world data perfectly, they clearly show a congested regime far more consistent with the observed recurring congestion. Further, not only did each phase of the process improve upon the previous “best” calibration, but the process overall resulted in a model in which all key calibration parameters have at least been considered, and likely adjusted when necessary.

Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29	
#1 6:00 - 6:15	65.4	72.6	72.6	72.6	73.0	73.0	73.0	73.0	73.0	73.0	73.4	74.1	74.1	74.1	73.6	73.7	73.7	73.7	72.4	70.5	70.5	70.5	71.8	71.1	71.1	71.1	71.1	71.1	72.8	72.8
#2 6:15 - 6:30	68.9	70.4	70.4	70.4	72.5	72.5	72.5	72.1	72.1	72.1	73.3	73.9	73.9	73.9	74.0	74.3	74.3	74.3	74.1	73.4	73.4	73.4	74.5	74.5	74.5	74.5	74.5	72.7	72.7	
#3 6:30 - 6:45	73.2	70.8	70.8	70.8	71.1	71.1	71.1	71.5	71.5	71.5	73.1	72.5	72.5	72.5	73.5	73.1	73.1	73.1	72.5	72.0	72.0	72.0	71.7	73.7	73.7	73.7	73.7	72.9	72.9	
#4 6:45 - 7:00	72.5	72.6	72.6	72.6	71.3	71.3	71.3	71.6	71.6	71.6	70.8	71.0	71.0	71.0	72.4	71.7	71.7	71.7	71.7	70.8	70.8	70.8	72.0	72.9	72.9	72.9	72.9	73.5	73.5	
#5 7:00 - 7:15	73.7	70.3	70.3	70.3	70.7	70.7	70.7	71.9	71.9	71.9	73.3	71.9	71.9	71.9	72.5	72.5	72.5	72.5	72.5	70.8	70.8	70.8	71.7	72.1	72.1	72.1	72.1	73.7	73.7	
#6 7:15 - 7:30	68.6	67.6	67.6	67.6	68.4	68.4	68.4	68.4	68.4	68.4	67.4	67.3	67.3	67.3	69.4	68.3	68.3	68.3	69.8	69.1	69.1	69.1	71.7	72.1	72.1	72.1	72.1	71.7	71.7	
#7 7:30 - 7:45	67.9	64.7	64.7	64.7	68.6	68.6	68.6	60.9	60.9	60.9	46.5	48.8	48.8	48.8	63.9	62.8	62.8	62.8	59.0	59.1	59.1	59.1	66.4	70.1	70.1	70.1	70.1	70.9	70.9	
#8 7:45 - 8:00	71.3	68.7	68.7	68.7	67.8	67.8	67.8	53.9	53.9	53.9	31.5	38.7	38.7	38.7	45.3	46.4	46.4	46.4	36.0	53.9	53.9	53.9	64.8	68.5	68.5	68.5	68.5	68.5	68.5	
#9 8:00 - 8:15	71.3	68.1	68.1	68.1	67.1	67.1	67.1	56.1	56.1	56.1	36.8	41.0	41.0	41.0	36.9	43.9	43.9	43.9	44.2	55.1	55.1	55.1	62.3	67.6	67.6	67.6	67.6	70.2	70.2	
#10 8:15 - 8:30	72.7	68.7	68.7	68.7	67.5	67.5	67.5	67.3	67.3	67.3	55.9	43.0	43.0	43.0	35.8	43.8	43.8	43.8	42.2	56.9	56.9	56.9	64.6	69.1	69.1	69.1	69.1	71.1	71.1	
#11 8:30 - 8:45	70.3	70.8	70.8	70.8	73.0	73.0	73.0	70.5	70.5	70.5	64.3	59.0	59.0	59.0	57.7	54.9	54.9	54.9	46.7	57.0	57.0	57.0	63.8	69.5	69.5	69.5	69.5	70.2	70.2	
#12 8:45 - 9:00	71.8	70.9	70.9	70.9	71.1	71.1	71.1	70.1	70.1	70.1	70.1	70.3	70.0	70.0	70.0	63.4	54.9	54.9	54.9	57.8	62.4	62.4	62.4	65.7	68.1	68.1	68.1	70.0	70.0	
#13 9:00 - 9:15	73.3	70.9	70.9	70.9	69.7	69.7	69.7	70.3	70.3	70.3	72.9	72.9	72.9	72.9	72.6	73.0	73.0	73.0	73.1	71.2	71.2	71.2	72.5	72.2	72.2	72.2	72.2	73.2	73.2	
#14 9:15 - 9:30	73.8	69.8	69.8	69.8	71.3	71.3	71.3	73.6	73.6	73.6	73.3	72.6	72.6	72.6	72.5	73.0	73.0	73.0	72.6	72.3	72.3	72.3	74.4	73.3	73.3	73.3	73.3	73.9	73.9	
#15 9:30 - 9:45	70.1	68.5	68.5	68.5	68.3	68.3	68.3	69.8	69.8	69.8	71.4	72.7	72.7	72.7	73.1	73.4	73.4	73.4	73.9	71.7	71.7	71.7	72.1	72.2	72.2	72.2	72.2	73.0	73.0	
#16 9:45 - 10:00	67.2	68.7	68.7	68.7	68.2	68.2	68.2	68.1	68.1	68.1	68.9	70.1	70.1	70.1	71.0	71.1	71.1	71.1	71.1	70.5	70.5	70.5	71.9	71.3	71.3	71.3	71.3	72.9	72.9	
#17 10:00 - 10:15	69.8	69.1	69.1	69.1	67.4	67.4	67.4	67.7	67.7	67.7	68.1	69.7	69.7	69.7	69.0	70.7	70.7	70.7	71.4	70.7	70.7	70.7	72.7	72.3	72.3	72.3	72.3	72.6	72.6	
#18 10:15 - 10:30	72.5	66.9	66.9	66.9	69.1	69.1	69.1	72.7	72.7	72.7	74.4	74.0	74.0	74.0	74.0	72.9	72.1	72.1	72.1	72.3	71.1	71.1	71.1	73.5	74.2	74.2	74.2	74.2	73.9	73.9
#19 10:30 - 10:45	69.9	69.8	69.8	69.8	70.9	70.9	70.9	70.5	70.5	70.5	70.7	71.4	71.4	71.4	71.1	71.6	71.6	71.6	73.7	73.3	73.3	73.3	73.3	73.6	73.6	73.6	73.6	73.5	73.5	
#20 10:45 - 11:00	69.0	67.0	67.0	67.0	68.0	68.0	68.0	68.3	68.3	68.3	69.4	69.7	69.7	69.7	69.6	70.5	70.5	70.5	71.4	70.8	70.8	70.8	71.5	71.8	71.8	71.8	73.1	73.1	73.1	
#21 11:00 - 11:15	68.7	70.4	70.4	70.4	67.8	67.8	67.8	68.0	68.0	68.0	69.5	70.3	70.3	70.3	70.5	73.5	73.5	73.5	74.1	72.0	72.0	72.0	71.1	71.5	71.5	71.5	71.5	72.5	72.5	
#22 11:15 - 11:30	68.0	70.3	70.3	70.3	67.7	67.7	67.7	70.4	70.4	70.4	68.1	70.2	70.2	70.2	73.2	73.5	73.5	73.5	72.9	72.2	72.2	72.2	71.4	72.4	72.4	72.4	72.4	72.7	72.7	
Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29	
#1 6:00 - 6:15	75.0	72.0	69.3	74.6	75.0	65.7	73.0	72.1	74.5	68.6	74.5	67.8	74.6	70.5	74.2	69.9	74.0	69.4	74.0	68.3	73.8	68.8	73.0	68.5	74.0	75.0	71.2	74.6	74.7	
#2 6:15 - 6:30	75.0	71.8	69.5	74.6	75.0	65.4	72.9	71.9	74.0	68.6	74.5	67.4	74.2	70.2	73.5	69.5	73.4	69.3	73.4	68.0	73.3	68.6	72.4	68.4	73.6	75.0	71.4	74.6	74.4	
#3 6:30 - 6:45	74.9	71.4	69.6	74.6	74.9	64.9	72.8	71.5	72.7	68.7	74.3	66.5	72.8	70.0	71.5	68.5	71.6	69.3	71.5	67.0	71.3	68.4	70.1	67.5	72.1	74.7	71.3	74.6	73.2	
#4 6:45 - 7:00	74.2	70.9	69.6	74.6	73.8	63.7	72.5	70.4	70.1	68.8	72.8	64.9	70.1	68.5	67.9	66.8	68.4	69.1	68.2	65.5	68.1	68.1	66.5	66.3	69.6	74.0	71.2	73.8	70.9	
#5 7:00 - 7:15	72.5	70.3	69.6	73.4	71.0	61.5	70.0	66.3	65.9	68.8	69.7	62.2	65.6	63.1	62.0	63.4	63.4	64.4	62.8	62.7	63.0	63.0	60.8	64.0	65.5	72.5	68.9	72.0	66.8	
#6 7:15 - 7:30	69.9	68.1	67.0	71.6	66.8	57.7	65.3	62.2	61.6	65.7	66.7	57.1	61.0	57.8	56.2	58.5	58.5	59.8	57.6	57.9	57.9	57.9	55.4	61.5	61.6	70.9	66.2	70.0	62.9	
#7 7:30 - 7:45	67.6	65.7	64.2	70.0	63.1	53.5	61.2	58.4	57.7	62.6	63.5	52.2	47.5	42.6	37.1	58.3	58.3	58.7	42.4	58.9	58.9	58.9	56.7	62.2	62.6	71.3	67.0	70.4	63.7	
#8 7:45 - 8:00	67.0	64.8	63.3	69.4	61.1	50.9	59.1	55.8	54.9	62.8	28.9	47.1	39.5	37.7	37.1	58.3	52.7	41.6	40.3	58.9	58.9	58.9	57.2	62.4	62.9	71.5	67.2	70.5	63.8	
#9 8:00 - 8:15	68.5	66.4	65.1	70.5	63.2	53.5	61.3	57.9	56.1	60.4	30.1	46.0	47.8	42.3	37.6	43.7	40.8	38.9	39.7	58.9	58.9	58.9	57.8	62.7	63.4	71.6	67.5	70.7	64.2	
#10 8:15 - 8:30	71.1	69.3	68.3	72.3	67.2	57.9	65.6	62.0	61.5	65.4	39.1	42.3	38.6	41.0	38.6	48.4	45.8	40.0	39.6	58.9	58.9	58.9	57.9	62.8	63.4	71.7	67.5	70.8	64.4	
#11 8:30 - 8:45	73.3	70.5	69.6	73.9	70.8	61.2	69.6	66.2	65.8	68.8	58.8	49.2	42.5	60.4	52.5	52.9	46.8	41.1	39.6	58.9	58.9	58.9	58.3	62.9	63.7	71.8	67.7	71.0	64.8	
#12 8:45 - 9:00	74.6	71.0	69.6	74.6	73.4	63.3	72.4	69.9	69.6	68.8	72.4	60.4	65.8	62.1	61.6	59.1	57.5	50.2	43.0	59.0	59.0	59.0	58.5	63.0	63.8	71.8	67.8	71.1	65.2	
#13 9:00 - 9:15	75.0	71.5	69.6	74.6	74.7	64.5	72.7	71.4	72.1	68.8	74.0	62.3	68.5	66.6	66.2	65.7	66.8	67.5	66.2	64.4	66.1	66.1	65.6	65.8	68.8	73.7	71.2	73.3	69.7	
#14 9:15 - 9:30	75.0	71.8	69.4	74.6	75.0	65.1	72.8	71.7	73.4	68.7	74.5	63.4	70.2	68.6	68.4	66.9	68.7	69.2	68.3	65.4	68.0	67.8	67.4	66.5	70.1	74.2	71.5	73.9	71.1	
#15 9:30 - 9:45	75.0	72.0	69.3	74.6	75.0	65.4	72.9	72.0	74.2	68.6	74.4	64.3	71.5	69.6	70.2	67.8	70.3	69.3	70.0	66.2	69.7	68.0	69.1	67.1	71.2	74.5	71.5	74.4	72.2	
#16 9:45 - 10:00	75.0	72.1	69.2	74.6	75.0	65.6	72.9	72.2	74.6	68.4	74.4	64.8	72.4	69.7	71.5	68.4	71.4	69.3	71.2	66.8	70.9	68.1	70.2	67.5	72.1	74.7	71.5	74.6	73.0	
#17 10:00 - 10:15	75.0	72.2	69.1	74.6	75.0	65.7	73.0	72.3	74.9	68.4	74.4	65.2	72.9	69.7	72.3	68.8	72.2	69.3	72.1	67.2	71.7	68.2	71.1	67.8	72.6	74.8	71.6	74.6	73.4	
#18 10:15 - 10:30	75.0	72.3	69.0	74.6	75.0	65.8	73.0	72.4	75.0	68.4	74.4	65.4	73.3	69.7	72.8	69.1	72.6	69.4	72.5	67.4	72.2	68.3	71.5	68.0	72.9	74.9	71.6	74.6	73.7	
#19 10:30 - 10:45	75.0	72.3	68.9	74.6	75.0	65.9	73.0	72.5	75.0	68.3	74.4	67.1	74.6	70.1	74.4	70.0	74.3	69.4	74.2	68.3	73.9	68.5	73.5	68.7	74.2	75.0	71.8	74.7	74.7	
#20 10:45 - 11:00	75.0	72.3	68.9	74.6	75.0	65.9	7																							

Table 4.10 Facility travel times based on the target speeds compared to those from facility after Phase 2 calibration.

Analysis Period	Target Facility Travel Time (minutes)	Phase 2 Calibration Travel Time (minutes)	Difference (minutes)	Percent (Difference)
#1 6:00 - 6:15	11.98	11.91	-0.08	-0.63%
#2 6:15 - 6:30	11.85	11.96	0.12	0.97%
#3 6:30 - 6:45	11.98	12.14	0.16	1.30%
#4 6:45 - 7:00	12.07	12.49	0.42	3.45%
#5 7:00 - 7:15	12.07	13.22	1.16	9.58%
#6 7:15 - 7:30	12.54	14.15	1.61	12.86%
#7 7:30 - 7:45	14.26	15.32	1.05	7.39%
#8 7:45 - 8:00	16.56	16.47	-0.08	-0.49%
#9 8:00 - 8:15	16.27	16.25	-0.02	-0.10%
#10 8:15 - 8:30	15.45	16.13	0.68	4.39%
#11 8:30 - 8:45	13.76	15.10	1.34	9.74%
#12 8:45 - 9:00	13.08	13.69	0.61	4.70%
#13 9:00 - 9:15	12.07	12.61	0.54	4.48%
#14 9:15 - 9:30	11.92	12.39	0.47	3.94%
#15 9:30 - 9:45	12.13	12.23	0.10	0.84%
#16 9:45 - 10:00	12.37	12.13	-0.25	-1.99%
#17 10:00 - 10:15	12.37	12.06	-0.31	-2.54%
#18 10:15 - 10:30	12.00	12.02	0.03	0.22%
#19 10:30 - 10:45	12.05	11.89	-0.17	-1.39%
#20 10:45 - 11:00	12.39	11.84	-0.55	-4.47%
#21 11:00 - 11:15	12.26	11.83	-0.43	-3.54%
#22 11:15 - 11:30	12.16	11.82	-0.35	-2.86%
Average	12.89	13.16	0.27	2.1%
Avg Absolute Error	-	-	0.48	3.71%

Figure 4.12 shows the set of pre-breakdown CAFs as estimated by the GA algorithm. It is immediately clear that there exists some “noise” within the generated values, and the capacity of some segments has been adjusted even when no adjustment was needed. This can occur due a lack of sensitivity to the parameter for segments whose D/C ratio is significantly below 1.0. In these cases, neither increasing nor decreasing the capacity has much if any effect on the predicted speeds. Consequently, it may be prudent for an analyst to perform a basic sensitivity analysis to re-adjust some of the estimated CAFs back to the default value.

Alternatively, the encoding proposed previously could be adjusted to explicitly constrain the location and/or number of allowed pre-breakdown adjustments. It will be clear for certain facilities, for geometric or other factors, there are a specific number of bottlenecks, or bottlenecks at specific locations, that require pre-breakdown adjustments to be estimated. It is possible to constrain how the set of pre-breakdown CAFs is determined by altering the existing coding for pre-breakdown CAFs proposed earlier. The altered encoding should allow both the location and the value of a pre-specified number of CAFs to be varied by the GA framework. A special case of this encoding can be developed where both the number and location of the CAFs is specified by an analyst, and only the adjustment value is varied by the GA search process.

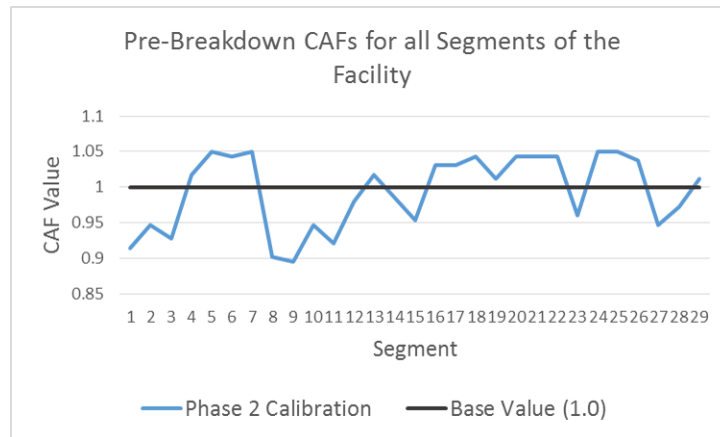


Figure 4.12 Chart showing the set of pre-breakdown CAFs for all segments of the facility as estimated during the Phase 2 calibration process.

If an analyst specifies that n pre-breakdown CAFs should be determined, the encoding will have $2n$ variables, one segment index variable and one adjustment value for each location. The adjustment can be encoded into a binary string in the same manner as for the full set of adjustments

presented previously, but the array of segment locations must be treated as a real-valued array if it is not fixed by the analyst. Unfortunately, the segment locations cannot be encoded as binary values because a single integer value that is not a power of two cannot readily be represented as a binary string. Unlike floating point values, when dealing solely with integers, step sizes and basic rounding cannot be employed as there is no general way to ensure the representation is balanced across the full search range. However, when dealing with an array of integer values, the individual values can maintain their base 10 representation, and the array can be treated in largely the same manner as a binary string. Most common crossover operators would require no adjustment as they simply switch values between arrays. Mutation operators will require slight alterations as they have to consider the range of values for the integer value using a probability distribution (e.g., uniform or Gaussian) instead of just the two binary possibilities. While real-value GA encodings such as this are not uncommon, they do lack many of the most desirable properties of a binary encoding, and may result in slower convergence.

In the event that the number adjustments is limited to just one segment, an exhaustive search is likely a better option than the GA framework. Further, an exhaustive search is computationally tractable regardless of whether the location of the single adjustment is fixed or allowed to vary. The computational time should not exceed that of the GA approach, and the optimal solution is guaranteed to be found. With a CAF step size of 0.001 and a range of [0.5,1.0], there are 500 potential values for the adjustment. If the location of the adjustment is fixed, the FREEVAL core engine would need to be invoked 500 times, which would have a computational burden equivalent to evolving a population of 100 organisms for 5 generations. This is less than 1% of the effort spent in each of the calibration phases for this case study where a population of 200 organisms was evolved over 400 generations. Even if the location of the adjustment is not fixed, and is allowed to be any of a facility's segments, the burden still remains comparatively small. It is highly unlikely a facility will have more than 50 segments, and if the exhaustive search must be conducted for as many as 50 segments, the computational burden still remains less than 50% of a typical phase of the calibration process. Further, if the step size is limited to 0.01, the burden is reduced by a factor of ten!

This special case of allowing a single CAF to be adjusted was explored for the I-540 case study. The exhaustive search approach for the single pre-breakdown capacity adjustment at an unknown segment location was performed. The search determined that the optimal location for the CAF is at segment 15, and the optimal CAF value is 0.94. Table 4.11 shows a breakdown of the remaining errors in speeds by regime for this special case Phase 2 calibration, and Figure 4.13 shows the new speed contour. While the reduction in speed error is not quite as good as when the full set of CAFs is allowed to be adjusted, the special case Phase 2 still showed significant improvement over the Phase 1 calibration. Further, it provides a more easily justifiable alteration of capacities for the facility.

The adjustment at segment 15 caused the D/C ratio to be above 1.0 for analysis periods 8 and 9 (7:45am-8:15am) leading to the active bottleneck as seen in both the target and predicted contours.

Table 4.11 Summary results for the Phase 1 calibration run. All errors are the average absolute difference in speed.

Error	All Speeds	$v_{i,p} < 65$	$v_{i,p} < 55$	$v_{i,p} < 45$	$v_{i,p} < 35$
Uncalibrated	6.12 mph	12.91 mph	17.21 mph	21.83 mph	36.86 mph
After Phase 1	4.15 mph	7.24 mph	6.55 mph	4.19 mph	7.54 mph
% Improvement	32.09%	43.86%	61.97%	80.78%	79.53%
Phase 2 All CAF	3.73 mph	4.85 mph	4.67 mph	3.32 mph	2.62 mph
% Improvement over Phase 1	10.13%	33.05%	28.55%	20.86%	65.28%
% Improvement over uncalibrated	39.05%	62.43%	72.86%	84.79%	92.89%
Phase 2 Single CAF	3.77 mph	5.77 mph	5.41 mph	3.45 mph	3.73 mph
% Improvement over Phase 1	9.10%	20.26%	17.25%	17.75%	50.57%
% Improvement over uncalibrated	38.27%	55.23%	68.53%	84.19%	89.88%

Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29
#1 6:00 - 6:15	65.4	72.6	72.6	72.6	73.0	73.0	73.0	73.0	73.0	73.0	73.4	74.1	74.1	74.1	73.6	73.7	73.7	73.7	72.4	70.5	70.5	70.5	71.8	71.1	71.1	71.1	71.1	72.8	72.8
#2 6:15 - 6:30	68.9	70.4	70.4	70.4	72.5	72.5	72.5	72.1	72.1	72.1	73.3	73.9	73.9	73.9	74.0	74.3	74.3	74.3	74.1	73.4	73.4	73.4	74.5	74.5	74.5	74.5	74.5	72.7	72.7
#3 6:30 - 6:45	73.2	70.8	70.8	70.8	71.1	71.1	71.1	71.5	71.5	71.5	73.1	72.5	72.5	72.5	73.5	73.1	73.1	73.1	72.5	72.0	72.0	71.7	73.7	73.7	73.7	73.7	72.9	72.9	
#4 6:45 - 7:00	72.5	72.6	72.6	72.6	71.3	71.3	71.3	71.6	71.6	71.6	70.8	71.0	71.0	71.0	72.4	71.7	71.7	71.7	71.7	70.8	70.8	70.8	72.0	72.9	72.9	72.9	72.9	73.5	73.5
#5 7:00 - 7:15	73.7	70.3	70.3	70.3	70.7	70.7	70.7	71.9	71.9	71.9	73.3	71.9	71.9	71.9	72.5	72.5	72.5	72.5	72.5	70.8	70.8	70.8	71.7	72.1	72.1	72.1	72.1	73.7	73.7
#6 7:15 - 7:30	68.6	67.6	67.6	67.6	68.4	68.4	68.4	68.4	68.4	68.4	67.4	67.3	67.3	67.3	69.4	68.3	68.3	68.3	69.8	69.1	69.1	69.1	71.7	72.1	72.1	72.1	71.7	71.7	
#7 7:30 - 7:45	67.9	64.7	64.7	64.7	68.6	68.6	68.6	60.9	60.9	60.9	46.5	48.8	48.8	48.8	63.9	62.8	62.8	62.8	59.0	59.1	59.1	59.1	66.4	70.1	70.1	70.1	70.1	70.9	70.9
#8 7:45 - 8:00	71.3	68.7	68.7	68.7	67.8	67.8	67.8	53.9	53.9	53.9	31.5	38.7	38.7	38.7	45.3	46.4	46.4	46.4	36.0	53.9	53.9	53.9	64.8	68.5	68.5	68.5	68.5	68.5	68.5
#9 8:00 - 8:15	71.3	68.1	68.1	68.1	67.1	67.1	67.1	56.1	56.1	56.1	36.8	41.0	41.0	41.0	36.9	43.9	43.9	43.9	44.2	55.1	55.1	55.1	62.3	67.6	67.6	67.6	67.6	70.2	70.2
#10 8:15 - 8:30	72.7	68.7	68.7	68.7	67.5	67.5	67.5	67.3	67.3	67.3	55.9	43.0	43.0	43.0	35.8	43.8	43.8	43.8	42.2	56.9	56.9	56.9	64.6	69.1	69.1	69.1	69.1	71.1	71.1
#11 8:30 - 8:45	70.3	70.8	70.8	70.8	73.0	73.0	73.0	70.5	70.5	70.5	64.3	59.0	59.0	59.0	57.7	54.9	54.9	54.9	46.7	57.0	57.0	57.0	63.8	69.5	69.5	69.5	69.5	70.2	70.2
#12 8:45 - 9:00	71.8	70.9	70.9	70.9	71.1	71.1	71.1	70.1	70.1	70.1	70.3	70.0	70.0	70.0	63.4	54.9	54.9	54.9	57.8	62.4	62.4	62.4	65.7	68.1	68.1	68.1	68.1	70.0	70.0
#13 9:00 - 9:15	73.3	70.9	70.9	70.9	69.7	69.7	69.7	70.3	70.3	70.3	72.9	72.9	72.9	72.9	72.6	73.0	73.0	73.0	73.1	71.2	71.2	71.2	72.5	72.2	72.2	72.2	72.2	73.2	73.2
#14 9:15 - 9:30	73.8	69.8	69.8	69.8	71.3	71.3	71.3	73.6	73.6	73.6	73.3	72.6	72.6	72.6	72.5	73.0	73.0	73.0	72.6	72.3	72.3	72.3	74.4	73.3	73.3	73.3	73.3	73.0	73.0
#15 9:30 - 9:45	70.1	68.5	68.5	68.5	68.3	68.3	68.3	69.8	69.8	69.8	68.1	71.4	72.7	72.7	72.7	73.1	73.4	73.4	73.4	73.9	71.7	71.7	71.7	72.1	72.2	72.2	72.2	73.0	73.0
#16 9:45 - 10:00	67.2	68.7	68.7	68.7	68.2	68.2	68.2	68.1	68.1	68.1	68.9	70.1	70.1	70.1	71.0	71.1	71.1	71.1	71.1	70.5	70.5	70.5	71.9	71.3	71.3	71.3	72.9	72.9	72.9
#17 10:00 - 10:15	69.8	69.1	69.1	69.1	67.4	67.4	67.4	67.7	67.7	67.7	68.1	69.7	69.7	69.7	69.0	70.7	70.7	70.7	71.4	70.7	70.7	70.7	72.7	72.3	72.3	72.3	72.3	72.6	72.6
#18 10:15 - 10:30	72.5	66.9	66.9	66.9	69.1	69.1	69.1	72.7	72.7	72.7	74.4	74.0	74.0	74.0	72.9	72.1	72.1	72.1	72.3	71.1	71.1	71.1	73.5	74.2	74.2	74.2	74.2	73.9	73.9
#19 10:30 - 10:45	69.9	69.8	69.8	69.8	70.9	70.9	70.9	70.5	70.5	70.5	70.7	71.4	71.4	71.4	71.1	71.6	71.6	71.6	73.7	73.3	73.3	73.3	73.6	73.6	73.6	73.6	73.5	73.5	73.5
#20 10:45 - 11:00	69.0	67.0	67.0	67.0	68.0	68.0	68.0	68.3	68.3	68.3	69.4	69.7	69.7	69.7	69.6	70.5	70.5	70.5	71.4	70.8	70.8	70.8	71.5	71.8	71.8	71.8	71.8	73.1	73.1
#21 11:00 - 11:15	68.7	70.4	70.4	70.4	67.8	67.8	67.8	68.0	68.0	68.0	69.5	70.3	70.3	70.3	70.5	73.5	73.5	73.5	74.1	72.0	72.0	72.0	71.1	71.5	71.5	71.5	71.5	72.5	72.5
#22 11:15 - 11:30	68.0	70.3	70.3	70.3	67.7	67.7	67.7	70.4	70.4	70.4	68.1	70.2	70.2	70.2	73.2	73.5	73.5	73.5	72.9	72.2	72.2	72.2	71.4	72.4	72.4	72.4	72.4	72.7	72.7
Analysis Period	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7	Seg. 8	Seg. 9	Seg. 10	Seg. 11	Seg. 12	Seg. 13	Seg. 14	Seg. 15	Seg. 16	Seg. 17	Seg. 18	Seg. 19	Seg. 20	Seg. 21	Seg. 22	Seg. 23	Seg. 24	Seg. 25	Seg. 26	Seg. 27	Seg. 28	Seg. 29
#1 6:00 - 6:15	75.0	72.1	69.3	74.6	75.0	65.7	73.0	72.2	75.0	68.6	74.4	68.0	74.7	70.6	74.3	70.1	73.9	69.4	74.2	68.6	73.5	68.9	74.3	68.8	73.5	75.0	71.3	74.6	74.8
#2 6:15 - 6:30	75.0	71.9	69.4	74.6	75.0	65.5	72.9	72.1	75.0	68.6	74.5	67.6	74.3	70.4	73.8	69.9	73.3	69.4	73.7	68.3	72.9	68.7	74.0	68.7	73.1	75.0	71.5	74.6	74.5
#3 6:30 - 6:45	75.0	71.6	69.6	74.6	74.8	65.1	72.8	71.7	74.8	68.8	74.5	66.9	73.0	70.1	71.9	69.0	71.5	69.3	72.1	67.5	70.8	68.5	72.6	68.0	71.4	74.6	71.5	74.6	73.6
#4 6:45 - 7:00	75.0	71.1	69.6	74.6	73.4	64.1	72.6	71.2	74.0	68.8	74.5	65.6	70.5	69.7	68.7	67.6	68.3	68.3	69.1	66.2	67.3	67.3	70.2	66.9	68.5	73.8	71.4	74.6	71.5
#5 7:00 - 7:15	74.6	70.5	69.6	73.6	70.1	62.3	68.9	70.4	72.2	68.9	73.3	63.4	66.3	66.3	63.3	63.3	63.3	63.3	64.3	61.8	61.8	61.8	66.2	63.9	63.9	72.2	71.2	73.4	67.9
#6 7:15 - 7:30	73.5	69.9	69.6	72.0	65.4	59.3	63.8	69.7	70.2	68.8	71.8	60.8	62.1	62.1	58.0	58.4	58.4	58.4	59.5	56.4	56.4	56.4	62.4	59.6	59.6	70.5	70.5	72.1	64.3
#7 7:30 - 7:45	72.4	69.4	69.5	70.5	61.3	56.1	59.3	68.3	68.3	68.3	70.1	57.5	52.9	42.3	36.4	56.8	50.6	55.1	38.7	56.8	56.8	56.8	63.0	60.3	60.3	70.8	70.8	72.3	64.7
#8 7:45 - 8:00	72.1	69.2	69.5	69.9	59.1	54.2	56.9	66.9	64.2	57.3	27.8	38.9	41.8	35.4	36.4	49.8	45.4	39.7	35.8	56.9	56.9	56.9	63.3	60.7	60.7	71.0	71.0	72.3	64.8
#9 8:00 - 8:15	72.9	69.5	69.5	70.9	61.4	56.2	59.3	68.0	63.2	43.2	22.4	41.0	43.6	36.2	33.5	47.3	45.8	48.2	36.8	56.9	56.9	56.9	63.8	61.2	61.2	71.2	71.2	72.5	65.2
#10 8:15 - 8:30	74.0	70.1	69.6	72.6	65.8	59.5	64.1	69.6	70.1	68.8	40.2	44.4	45.9	41.5	37.8	47.8	45.2	40.5	35.5	56.9	56.9	56.9	63.9	61.3	61.3	71.2	71.2	72.5	65.3
#11 8:30 - 8:45	74.8	70.7	69.6	74.0	69.9	62.1	68.5	70.4	72.2	68.8	65.4	54.0	55.6	40.2	49.1	45.1	44.6	40.8	35.3	56.9	56.9	56.9	64.1	61.5	61.5	71.3	71.2	72.7	65.7
#12 8:45 - 9:00	75.0	71.2	69.6	74.6	72.8	63.8	71.9	71.1	73.8	68.8	74.5	61.3	65.8	65.8	60.6	59.4	56.9	48.4	36.8	56.9	56.9	56.9	64.2	61.6	61.6	71.3	71.1	72.8	66.1
#13 9:00 - 9:15	75.0	71.6	69.5	74.6	74.5	64.7	72.8	71.6	74.6	68.8	74.5	62.9	68.5	68.5	66.5	66.0	66.0	66.0	69.9	63.7	63.7	63.7	68.7	66.1	66.6	73.2	71.5	74.1	69.8
#14 9:15 - 9:30	75.0	71.9	69.4	74.6	75.0	65.2	72.9	71.9	74.9	68.7	74.5	63.8	70.1	69.6	68.6	67.4	68.0	68.0	68.7	65.9	66.6	66.6	70.4	66.9	68.6	73.9	71.6	74.5	71.4
#15 9:30 - 9:45	75.0	72.1	69.2	74.6	75.0	65.5	72.9	72.1	75.0	68.6	74.4	64.6	71.4	69.7	70.3	68.2	69.9	69.3	70.3	66.6	68.4	68.0	71.5	67.4	70.0	74.3	71.6	74.6	72.4
#16 9:45 - 10:00	75.0	72.2	69.1	74.6	75.0	65.7	73.0	72.3	75.0	68.4	74.4	65.1	72.3	69.8	71.5	68.7	70.9	69.4	71.4	67.1	69.8	68.2	72.3	67.8	71.0	74.5	71.6	74.6	73.2
#17 10:00 - 10:15	75.0	72.3	69.1	74.6	75.0	65.8	73.0	72.4	75.0	68.3	74.4	65.4	72.8	69.8	72.3	69.1	71.7	69.4	72.3	67.5	70.8	68.3	72.9	68.1	71.7	74.7	71.7	74.7	73.6
#18 10:15 - 10:30	75.0	72.3	69.0	74.6	75.0	65.9	73.0	72.5	75.0	68.3	74.4	67.6	74.7	70.3	74.6	70.3	74.2	69.5	74.4	68.6	73.6	68.5	74.5	69.0	73.9	75.0	72.0	74.7	74.8
#19 10:30 - 10:45	75.0	72.4	68.9	74.6	75.0	65.9	73.0	72.5	75.0	68.2	74.4	68.1	75.0	74.8	70.6	74.7	69.5	74.8	68.9	74.3	68.6	74.8	69.3	74.5	75.0	72.0	74.7	75.0	75.0
#20 10:45 - 11:00	75.0	72.4	68.9	74.5	75.0	66.0	73.0	72.5	75.0	68.3	7																		

4.13 Conclusions and Future Work

This paper presents a novel approach for model calibration based on the HCM's freeway facilities methodology. The work expanded upon previous research developing an initial framework for automated demand volume estimation and adjustment for the methodology. The proposed framework fits within the calibration procedure provided in the upcoming 6th edition of the HCM, and avoids the arduous trial and error steps recommended by the manual's current calibration procedure. At the core of the proposed process is a genetic algorithm (GA) metaheuristic approach that can be used to operate on key calibration parameters related to both demand estimation and capacity adjustment. The optimization nature of the metaheuristic approach is designed to find a calibration such that each model can best match performance measures observed in the real world.

The proposed approach has been implemented alongside FREEVAL, the computational engine which faithfully implements the freeway facilities methodology of the HCM. A real world case study was conducted for a section of I-540 outside of Raleigh, NC. After collecting the necessary data to construct the facility and populate its initial inputs, the uncalibrated version of the model predicted an average facility travel time close to the target measure, but individual segment performance measures that varied largely from the target data. This represented a precarious situation in which a cursory examination of performance could lead to utilization of a model that with incorrect behavior. The model was then placed in the two stage calibration procedure to estimate a new set of demands and adjust the capacity parameters. Each phase of the process showed significant improvement over the previous version of the model, and greatly reduced the facility travel time and individual segment speed errors.

Overall, the work presented in this paper was able to accomplish the following:

- Developed a unified approach for model calibration that considers both demand and capacity parameters.
- A genetic algorithm framework was implemented in Java for easy integration with the existing HCM FREEVAL computational engine.
- The calibration approach is flexible and can consider both facility-wide and individual segment performance measures, allowing for a more complete calibration approach.
- Replaces existing challenging calibration process with an automated one that doesn't require in depth knowledge of the inner workings of the methodology.

While the proposed calibration approach is unlikely to perfectly match target performance measures, it was still shown to be able to bring predicted performance to within the accepted

margins of error as outlined in the HCM. More importantly, it is able to do so even with limited available information regarding inputs, and with minimal effort on the part of the analyst. Unlike the challenging and time-consuming process of the current HCM guidance, the framework is largely automated beyond the collection of initial inputs, and only requires insight about the facility location itself, as opposed to needing an in-depth understanding of the workings of the methodology in order to be effective. Thus the framework contributes both by providing more accurate models, and by largely eliminating the most difficult barriers to entry for using the HCM freeway facilities methodology.

The framework provides multiple avenues of investigation for future work. In this paper a simple objective function was used, but a more nuanced weighting approach could be developed to both improve the convergence rate of the genetic algorithm and the quality of the calibration. Additionally, constraints could be introduced into the framework to better control the generation of both demand volumes and capacity adjustments, and could place limits on the number of segments adjusted or incorporate aspects of a facility's geometry. Constraints on pre-breakdown CAFs could serve to identify bottlenecks as well as limit CAF application to just those segments. An in-depth sensitivity analysis could be conducted on the full set of pre-breakdown CAFs, and a local search heuristic could be developed as a post-processing step to help stabilize unnecessary or inaccurate capacity adjustments. Beyond core freeway facilities analysis, the calibration methodology could be extended to develop adjustment factors for non-recurring congestion events. These could be used for analysis of individual scenarios, but would also serve to improve the quality of a full year reliability analysis. Finally, while this work focused entirely on the HCM freeway facilities methodology, the framework can be extended with minor adjustments to be utilized with the underlying cell transmission model and all of its extensions.

CHAPTER

5

CONCLUSIONS AND FUTURE WORK

5.1 Summary and Conclusions

This work presents two optimization approaches for expanding the use of the Highway Capacity Manual's freeway facilities methodology. The approaches address two deficiencies of the method as it is currently presented in the HCM. In Chapter 2, a linear programming formulation is presented that provides a system optimal dynamic traffic assignment model of the HCM methodology. The formulation is developed by breaking down and reconstructing the relationships of the oversaturated regime as a set of linear constraints. The constraints are combined with an objective function that defines the desired optimal properties of the system and utilized within a linear programming model to expand the current limited iterative process. Reformulating the methodology's relationships in this way allows the model to be used for a variety of new types of analyses. Two immediate new applications were explored and their usefulness was demonstrated for example facilities.

- It was found that a "core" analysis of a facility using an objective function minimizing the total number of vehicles in the system at each time step provides insights into optimal patterns of flow for the system.
- These insights are gained through an inspection of the "holding-back" flows allowed by the

relaxation of the node-flow relationships previously strictly enforced in the computational procedure.

- The resulting adjusted flows represent an optimal allocation of demands equivalent to the congestion management strategy of peak demand spreading.
- The set of demands determined by the LP can be used to help guide analysts' decision making when implementing demand spreading strategies.
- By making small adjustments to the core model constraints, and using an objective function minimizing the number of vehicles queued on the mainline, system optimal ramp metering rates are computed for all on-ramps of the facility.
- To avoid simply moving mainline delay to the ramps, constraints are added to the model to control the allowable size of ramp queues.
- Even for the most restrictive case, the ramp metering rates computed by the model for an example reduced the mainline travel time by as much as 13%.

In addition to the above findings, a principal contribution of the linear programming model is that it provides an alternative means of describing the HCM methodology compared to the one defined by the computational procedure. The new representation provides an ideal starting point for new types of analyses conducted on calibrated freeway facility models. Embedding the methodology within an optimization framework greatly broadens the scope of possible analysis and opens it up for use in applications in ways previously precluded by inherent limitations of the "forward" iterative formulation. The two applications explored represent just a small fraction of the increased capabilities afforded by the new representation of the model. While the system optimal nature of the linear program is not a perfectly accurate representation of real world traffic behavior, it still provides a number of valuable insights. Knowing and understanding the best possible behavior of the network can inform both design and operational decisions.

The linear programming formulation does suffer from certain limitations. One limitation is that the relaxation of the strict minimum relationship for node flows allows the holding-back phenomenon to occur. While this phenomenon can be beneficial for some system optimal applications, it still yields less realistic behavior for certain applications than user optimal models. Additionally, just as with all macrosimulation approaches, the LP formulation struggles to correctly account for first-in, first-out flows. Penalties in the objective function or additional constraints can be introduced to help address this, but there is no guarantee that first-in, first-out behavior can be enforced, particularly in the presence of heavy congestion upstream of off-ramp segments.

In Chapters 3 and 4, a novel approach for model calibration is developed by considering the set of methodology parameters as adjustment variables for a system identification problem. The calibration approach uses a genetic algorithm metaheuristic to automate the arduous process of concurrently adjusting large numbers of parameters to ensure the model predicts performances that matches observed real-world conditions. Chapter 3 first presents two encoding approaches for traffic demand estimation and adjustment. Each encoding requires an analyst to have access to average annual daily traffic (AADT) values for each entry and exit node of the facility, and allows general knowledge of demand patterns to be incorporated when available.

In addition to adjusting demand volumes, the manual recommends that three capacity parameters be considered during the calibration process: pre-breakdown capacity adjustments, queue discharge flow rate, and the facility-wide jam density. Consequently, Chapter 4 presents a unified approach that addresses both the demand and capacity considerations of the full calibration process. A multi-stage approach is developed and breaks the process into two phases, each with distinct decision variables and differing optimization focuses. The first phase operates on demand volumes and the queue discharge rate in conjunction, and an increased emphasis is placed on matching facility wide travel time across analysis periods. The second phase operates on pre-breakdown segment capacity adjustments and the facility-wide jam density. This phase places a higher emphasis on matching speeds observed on individual segments for each analysis period. Findings from the model calibration phase can be summarized as follows:

- It was found that a mixture distribution of n Cauchy or normal random variables can be used to create daily demand profiles that represent realistic traveler behavior and allocate demand in a sufficiently smooth manner.
- On its own, demand estimation and adjustment using the GA framework was shown to significantly improve upon uncalibrated facilities without any manual adjustments of inputs by the analyst.
- Combining the approach for demands with key capacity parameters resulted in an even more effective unified calibration framework. Each phase of the calibration shows significant improvement over both the uncalibrated facility and the previous best available calibration.
- Calibrating for both facility travel time and individual segment speeds in conjunction yields a more accurate model than calibrating facility travel time only and can eliminate cases where closely matching target and predicted travel times are misleading.
- A case study conducted on a stretch of interstate I-540 westbound in Raleigh, NC, demonstrates the effectiveness of the unified two-stage calibration framework. The case study facility saw

average speed error reductions of at least 70% for speeds in congested time periods, and an average reduction of 50% in differences between the observed and predicted travel times for each period.

- When the capacity of every segment along a facility is allowed to be adjusted, it was found that situations may arise where conflicting or unrealistic adjustments occur, likely due to overparameterization.
- An additional approach is presented for cases where it is beneficial to constrain the number of segments whose capacity can be adjusted, and represents a special case of the primary encoding for calibrating pre-breakdown capacity adjustments.
- The special case of a single unknown pre-breakdown CAF is examined in the context of the I-540 case study, and while the reduction in error is not as large (approximately 90% of the full CAF error reduction), the approach still improves upon the first phase of the calibration process and removes some potential issues of overparameterization.

Overall, the automated system identification approach developed in Chapters 3 and 4 represents a major improvement over the existing best practice for model calibration. The genetic algorithm framework greatly reduces the burden placed on the user and closes the gap between model creation and model utilization, a process that represents the biggest challenge for prospective analysts. Uncalibrated or poorly calibrated models have little to no practical use and consequently facility calibration is critical for utilizing the HCM methodology effectively. The improved calibration benefits not only the main analyses presented in the manual (e.g., core, reliability, and ATDM), but also improves current and future extensions to the methodology, such as the linear programming of Chapter 2.

5.2 Future Work

While the work presented in this dissertation has built on the existing methodology to improve and expand the uses of the method, it also introduces or highlights a number of avenues for future research.

The linear programming formulation of Chapter 2 could be adapted for any number optimization problems for congestion management and network design. Integer variables and nonlinear constraints can be introduced accommodate many discrete decision problems. Incorporation of integer variables can also serve as a basis to better model specific dynamic traffic assignment concepts such as user optimal behavior and first-in, first out-flows, both of which remain open research

questions for macrosimulation approaches. Studies can be conducted comparing both the peak spreading and ramp metering strategies to other existing approaches. This would require that a common means of analysis (e.g., implementation of additional strategies into the FREEVAL core computational engine) can be found.

The model calibration work of Chapters 3 and 4 provides opportunities for both improving the inner workings of the genetic algorithm itself, as well as expanding the scope of the framework's application to additional uses. In terms of the former, numerous additional operators can be implemented to bolster the quality of the current evolutionary search process. Aspects of memetic algorithms, or inter-generational learning approaches, could also be explored. On a similar note, only a small number of objective functions were tested for each application, and more complex objectives could serve to improve both the quality of final calibration and the convergence speed of the GA. In terms of demand calibration, the encodings presented in Chapter 3 incorporated only normal and Cauchy distributions, and further work could be done testing the performance of other distributions. On the capacity calibration side, additional constraints can be explored when calibrating the pre-breakdown capacity adjustments. Chapter 4 presented an unconstrained approach along with a simple constrained approach, but more complex schemes could be developed based on geometric or other insights. For example, most likely bottlenecks are expected to occur at on-ramp, lane drop, or weaving segments. These segments would be the most probable candidates for capacity adjustment. An extensive analysis could be conducted on the sensitivity of the full set of pre-breakdown adjustments, and a local search smoothing heuristic could be developed to identify and stabilize factors that have little to no impact on the model.

Future research could also be devoted to additional validation of the computed demand volumes and capacity values. Though challenging to acquire, if data is available, a study comparing the estimated and true capacity parameters could be conducted. Further, a single facility could be calibrated multiple times using target speeds and travel times representing multiple distinct days. The demand volumes and capacity parameter values determined for each calibration could then be compared to determine whether or not a robust set of model parameters has been estimated.

Finally, the genetic algorithm framework in this research dealt strictly with calibrating a facility of the core HCM freeway facilities level. However, with slight modification, the scope of the calibration approach can be greatly expanded. Remaining within the HCM, in the event that data is available, encodings could be developed and studied for calibration of aspects of reliability analysis that spans study periods of months or even years. Additionally, the GA framework could be decoupled from the HCM methodology and used to calibrate similar congested freeway models including, but not limited to, the Cell Transmission model and its numerous extensions.

REFERENCES

- Blum, C. & Roli, A. (2003). "Metaheuristics in combinatorial optimization: Overview and conceptual comparison". *ACM Computing Surveys (CSUR)* **35.3**, pp. 268–308.
- Carey, M. & Watling, D. (2012). "Dynamic traffic assignment approximating the kinematic wave model: System optimum, marginal costs, externalities and tolls". *Transportation Research Part B: Methodological* **46.5**, pp. 634–648.
- Carey, M. et al. (2014a). "Extending travel-time based models for dynamic network loading and assignment, to achieve adherence to first-in-first-out and link capacities". *Transportation Research Part B: Methodological* **65**, pp. 90–104.
- Carey, M. et al. (2014b). "Implementing first-in-first-out in the cell transmission model for networks". *Transportation Research Part B: Methodological* **65**, pp. 105–118.
- Chiu, Y.-C. et al. (2011). "Dynamic traffic assignment: A primer". *Transportation Research E-Circular* E-C153.
- Coello, C. A. C. (2002). "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art". *Computer Methods in Applied Mechanics and Engineering* **191.11**, pp. 1245–1287.
- Daganzo, C. F. (1994). "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory". *Transportation Research Part B: Methodological* **28.4**, pp. 269–287.
- (1995). "The cell transmission model, part 2: Network traffic". *Transportation Research Part B: Methodological* **29.2**, pp. 79–93.
- De Nicolao, G (1997). "System identification: Problems and perspectives". *12th Workshop on Qualitative Reasoning*, pp. 379–386.
- Delbem, A. C. et al. (2012). "Efficient forest data structure for evolutionary algorithms applied to network design". *Evolutionary Computation, IEEE Transactions on* **16.6**, pp. 829–846.
- Doan, K. & Ukkusuri, S. V. (2012). "On the holding-back problem in the cell transmission based dynamic traffic assignment models". *Transportation Research Part B: Methodological* **46.9**, pp. 1218–1238.
- Elefteriadou, L. & Lertworawanich, P. (2003). "Defining, measuring and estimating freeway capacity". *Washington: Transportation Research Board Meeting*.
- Gendreau, M. & Potvin, J.-Y. (2005). "Metaheuristics in combinatorial optimization". *Annals of Operations Research* **140.1**, pp. 189–213.

- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Gomes, G. & Horowitz, R. (2006). "Optimal freeway ramp metering using the asymmetric cell transmission model". *Transportation Research Part C: Emerging Technologies* **14.4**, pp. 244–262.
- Gomes, G. et al. (2008). "Behavior of the cell transmission model and effectiveness of ramp metering". *Transportation Research Part C: Emerging Technologies* **16.4**, pp. 485–513.
- Hadi, M. et al. (2014). "Pilot Testing of SHRP 2 Reliability Data and Analytical Products: Florida Pilot Site". *SHRP 2 Report*.
- Hallenbeck, M. et al. (1997). *Vehicle volume distributions by classification*. Tech. rep.
- Han, L. et al. (2011). "Complementarity formulations for the cell transmission model based dynamic user equilibrium with departure time choice, elastic demand and user heterogeneity". *Transportation Research Part B: Methodological* **45.10**, pp. 1749–1767.
- Highway Capacity Manual* (2016). 6th Edition: A Guide for Multimodal Mobility Analysis. Washington, D.C.: Transportation Research Board.
- Hu, J. et al. (2012). "On linear programs with linear complementarity constraints". English. *Journal of Global Optimization* **53.1**, pp. 29–51.
- Juan, W. et al. (2012). "Genetic algorithm for multiuser discrete network design problem under demand uncertainty". *Mathematical Problems in Engineering* **2012**.
- Karoonsoontawong, A. & Waller, S. (2006). "Dynamic Continuous Network Design Problem: Linear Bilevel Programming and Metaheuristic Approaches". *Transportation Research Record: Journal of the Transportation Research Board* **1964**, pp. 104–117. eprint: <http://dx.doi.org/10.3141/1964-12>.
- Li, Y. et al. (2003). "A Decomposition Scheme for System Optimal Dynamic Traffic Assignment Models". *Networks and Spatial Economics* **3.4**.
- Li, Z.-b. et al. (2014). "Development of control strategy of variable speed limits for improving traffic operations at freeway bottlenecks". English. *Journal of Central South University* **21.6**, pp. 2526–2538.
- Lighthill, M. J. & Whitham, G. B. (1955a). "On Kinematic Waves. I. Flood Movement in Long Rivers". English. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* **229.1178**, pp. 281–316.

- Lighthill, M. J. & Whitham, G. B. (1955b). "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads". *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **229**.1178, pp. 317–345.
- Lim, G. J. et al. (2015). "Reliability analysis of evacuation routes under capacity uncertainty of road links". *IIE Transactions* **47**.1, pp. 50–63. eprint: <http://dx.doi.org/10.1080/0740817X.2014.905736>.
- Lin, D.-Y. et al. (2011). "A Dual Variable Approximation Based Heuristic for Dynamic Congestion Pricing". English. *Networks and Spatial Economics* **11**.2, pp. 271–293.
- Lin, W.-H. & Ahanotu, D. (1995). "Validating the basic cell transmission model on a single freeway link". *PATH technical note; 95-3*.
- Ljung, L. (2010). "Perspectives on system identification". *Annual Reviews in Control* **34**.1, pp. 1–12.
- Lo, H. K. (2001). "A Cell-Based Traffic Control Formulation: Strategies and Benefits of Dynamic Timing Plans". English. *Transportation Science* **35**.2, pp. 148–164.
- Lu, C.-C. et al. (2013). "Dynamic origin–destination demand flow estimation under congested traffic conditions". *Transportation Research Part C: Emerging Technologies* **34**, pp. 16–37.
- Luathep, P. et al. (2011). "Global optimization method for mixed transportation network design problem: A mixed-integer linear programming approach". *Transportation Research Part B: Methodological* **45**.5, pp. 808–827.
- Ma, T. et al. (2015). "Nonlinear multivariate time space threshold vector error correction model for short term traffic state prediction". *Transportation Research Part B: Methodological* **76**.0, pp. 27–47.
- Malveo, A. E. (2013). *Assessing the Impact of Congestion During a Multi-County Evacuation*. Tech. rep. DTIC Document.
- Maryland CATT Lab, U. of (2008). *RITIS:Regional Integrated Transportation Information System*. URL: <https://www.ritis.org/> (visited on 08/11/2016).
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Morbidi, F. et al. (2014). "A new robust approach for highway traffic density estimation". *Control Conference (ECC), 2014 European*, pp. 2575–2580.
- Mudchanatongsuk, S. et al. (2008). "Robust Solutions for Network Design under Transportation Cost and Demand Uncertainty". English. *The Journal of the Operational Research Society* **59**.5, pp. 652–662.

- Munoz, L. et al. (2003). "Traffic density estimation with the cell transmission model". *American Control Conference, 2003. Proceedings of the 2003*. Vol. 5, 3750–3755 vol.5.
- Muñoz, L. et al. (2004). "Methodological calibration of the cell transmission model". *American Control Conference, 2004. Proceedings of the 2004*. Vol. 1. IEEE, pp. 798–803.
- NCDOT (2016). *Planning Level Extensions to NCDOT Freeway Analysis Tools*. URL: <https://connect.ncdot.gov/projects/planning/Pages/ProjDetails.aspx?ProjectID=2015-09>.
- Nisbet, J. et al. (2014). *Pilot Testing of SHRP 2 Reliability Data and Analytical Products: Washington*.
- North Carolina Department of Transportation (2016). *Traffic Surveys*. URL: <https://www.ncdot.gov/projects/trafficsurvey/> (visited on 08/11/2016).
- Pascale, A. et al. (2013). "Estimation of highway traffic from sparse sensors: Stochastic modeling and particle filtering". *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6158–6162.
- Pascale, A. et al. (2014). "Cooperative Bayesian Estimation of Vehicular Traffic in Large-Scale Networks". *Intelligent Transportation Systems, IEEE Transactions on* **15.5**, pp. 2074–2088.
- Richards, P. I. (1956). "Shock Waves on the Highway". English. *Operations Research* **4.1**, pp. 42–51.
- Smilowitz, K. & Daganzo, C. (1999). "Predictability of time-dependent traffic backups and other reproducible traits in experimental highway data". *California Partners for Advanced Transit and Highways (PATH)*.
- Sobolewski, M. et al. (2014). *Pilot Testing of SHRP 2 Reliability Data and Analytical Products: Minnesota*.
- Stankova, K. & De Schutter, B. (2010). "On freeway traffic density estimation for a jump Markov linear model based on Daganzo's cell transmission model". *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 13–18.
- Sumalee, A. et al. (2011). "Stochastic cell transmission model (SCTM): A stochastic dynamic traffic model for traffic state surveillance and assignment". *Transportation Research Part B: Methodological* **45.3**, pp. 507–533.
- Sun, H. et al. (2014). "Dynamic Network Design Problem under Demand Uncertainty: An Adjustable Robust Optimization Approach". *Discrete Dynamics in Nature and Society* **2014**.

- Tampère, C. & Immers, L. (2007). "An Extended Kalman Filter Application for Traffic State Estimation Using CTM with Implicit Mode Switching and Dynamic Parameters". *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 209–216.
- Travis Waller, S. & Ziliaskopoulos, A. K. (2006). "A Combinatorial user optimal dynamic traffic assignment algorithm". English. *Annals of Operations Research* **144**.1. Copyright - Springer Science+Business Media, LLC 2006; Last updated - 2014-08-30, pp. 249–261.
- Ukkusuri, S. V. & Waller, S. T. (2008). "Linear Programming Models for the User and System Optimal Dynamic Network Design Problem: Formulations, Comparisons and Extensions". English. *Networks and Spatial Economics* **8**.4, pp. 383–406.
- Ukkusuri, S. V. et al. (2012). "Dynamic user equilibrium with a path based cell transmission model for general traffic networks". *Transportation Research Part B: Methodological* **46**.10, pp. 1657–1684.
- Waller, S. & Ziliaskopoulos, A. (2001). "Stochastic Dynamic Network Design Problem". *Transportation Research Record: Journal of the Transportation Research Board* **1771**, pp. 106–113. eprint: <http://dx.doi.org/10.3141/1771-14>.
- Williges, C. et al. (2014). *Pilot Testing of SHRP 2 Reliability Data and Analytical Products: Southern California Pilot Site*. Tech. rep.
- Xu, T. et al. (2009). "Study on continuous network design problem using simulated annealing and genetic algorithm". *Expert Systems with Applications* **36**.2, Part 2, pp. 2735–2741.
- Yao, T. et al. (2009). "Evacuation Transportation Planning Under Uncertainty: A Robust Optimization Approach". English. *Networks and Spatial Economics* **9**.2, pp. 171–189.
- Zeng, Q. & Mouskos, K. C. (1997). "Heuristic search strategies to solve transportation network design problems". *Final report to the New Jersey Department of Transportation and the National Center for Transportation and Industrial Productivity*.
- Zheng, H. & Chiu, Y.-C. (2011). "A Network Flow Algorithm for the Cell-Based Single-Destination System Optimal Dynamic Traffic Assignment Problem". *Transportation Science* **45**.1, pp. 121–137. eprint: <http://dx.doi.org/10.1287/trsc.1100.0343>.
- Zhong, R. et al. (2013). "A linear complementarity system approach to macroscopic freeway traffic modeling". *Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on*, pp. 18–23.
- Zhong, R. et al. (2014). "Towards automatic model calibration of first-order traffic flow model". *Control Conference (CCC), 2014 33rd Chinese*, pp. 3423–3428.

Zhu, F. & Ukkusuri, S. V. (2013). "A cell based dynamic system optimum model with non-holding back flows". *Transportation Research Part C: Emerging Technologies* **36**, pp. 367 –380.

Ziliaskopoulos, A. K. (2000). "A Linear Programming Model for the Single Destination System Optimum Dynamic Traffic Assignment Problem". English. *Transportation Science* **34.1**, pp. 37–49.

APPENDIX

APPENDIX

A

FULL LINEAR PROGRAMMING MODEL

A.1 Full Linear Programming Formulation

Sections 2.4.1 - 2.4.5 presented constraints and objective function of the dynamic traffic assignment formulation of the HCM's oversaturated methodology. This section presents model in full.

Minimize:

$$\sum_{i=1}^{N_s} \sum_{t=1}^S \sum_{p=1}^P NV_{i,t,p} \quad (\text{A.1})$$

Subject to:

$$\sum_{t=1}^S \sum_{p=1}^P MF_{0,t,p} + \sum_{i \in \bar{N}} ONRF_{i,t,p} = \sum_{t=1}^S \sum_{p=1}^P MF_{NS,t,p} + \sum_{i \in \bar{F}} OFRF_{i,t,p} \quad (\text{A.2})$$

$$NV_{i,0,p} = KB_{i,p} L_i + UV_{i,S,p-1} \quad \forall i \in N_S, p \in P \quad (\text{A.3})$$

$$NV_{i,t,p} = NV_{i,t-1,p} + MF_{i,t,p} + ONRF_{i,t,p} - MF_{i+1,t,p} - OFRF_{i+1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.4})$$

$$NV_{i,t,p} \leq KJ * N_i^L * L_i \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.5})$$

$$UV_{i,t,p} = MI_{i+1,t,p} - MF_{i+1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.6})$$

$$SF_{i,t,p} = MF_{i+1,t,p} + OFRF_{i+1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.7})$$

$$MI_{i,t,p} = MF_{i-1,t,p} + ONRF_{i-1,t,p} - OFRF_{i,t,p} + UV_{i,1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.8})$$

$$MF_{i,t,p} \leq MI_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.9})$$

$$MF_{i,t,p} \leq SC_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.10})$$

$$MF_{i,t,p} \leq SC_{i-1,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.11})$$

$$MF_{i,t,p} \leq MO1_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.12})$$

$$MF_{i,t,p} \leq MO2_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.13})$$

$$MF_{i,t,p} \leq MO3_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.14})$$

$$MO1_{i,t,p} \leq SC_{i,t,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.15})$$

$$MO1_{i,t,p} \leq MO2_{i,t-1,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.16})$$

$$MO1_{i,t,p} \leq MO3_{i,t-1,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.17})$$

$$ASF_{i,t,p} = \left([KJ * N_i^L] - \frac{NV_{i,t,p} + NV_{i,t-1,p}}{2 * L_i} \right) * \frac{SC_{i,p}}{N_i^L * (KJ - KC)} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.18})$$

$$MO2 = ASF_{i,t,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.19})$$

$$MO3_{i,t,p} \leq MO1_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.20})$$

$$MO3_{i,t,p} \leq MO2_{i+1,t-wtt,p} + OFRF_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.21})$$

$$MO3_{i,t,p} \leq MO3_{i+1,t-wtt,p} + OFRF_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.22})$$

$$MO3_{i,t,p} \leq SC_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.23})$$

$$MO3_{i,t,p} \leq SC_{i+1,t-wtt,p} + OFRF_{i+1,t-wtt,p} - ONRF_{i,t,p} \quad \forall i \in N_S, t \in S, p \in P \quad (\text{A.24})$$

$$OFRF_{i,t,p} = P_D * MF_{i,t,p} \quad \forall i \in \tilde{N}, t \in S, p \in P \quad (\text{A.25})$$

$$\sum_{t=1}^S \sum_{p=1}^P ONRF_{i,t,p} = \sum_{p=1}^P ONRD_{i,p} / 4.0 \quad \forall i \in \tilde{N} \quad (\text{A.26})$$

$$ONRF_{i,t,p} \leq ONRD_{i,t,p} + ONRQ_{i,t-1,p} \quad \forall i \in \tilde{N}, t \in N_S, p \in P \quad (\text{A.27})$$

$$ONRF_{i,t,p} \leq RM_{i,t,p} \quad \forall i \in \tilde{N}, t \in N_S, p \in P \quad (\text{A.28})$$

$$ONRF_{i,t,p} \leq ONRC_{i,t,p} \quad \forall i \in \tilde{N}, t \in N_S, p \in P \quad (\text{A.29})$$

$$ONRQ_{i,t,p} = \sum_{x=1}^{p-1} \left[\frac{ONRD_{i,x}}{4} - \sum_{\tau=1}^S ONRF_{i,\tau,x} \right] + \sum_{\tau=1}^t \left[\frac{ONRD_{i,p}}{Th} - ONRF_{i,\tau,p} \right] \quad \forall i \in \tilde{N}, t \in N_S, p \in P \quad (\text{A.30})$$

A.2 Python and Gurobi Code

```

import time
import model_check
from numpy import zeros
import matplotlib.pyplot as plt
import gurobipy as gbp
__author__ = 'jltrask'

xrange = range

```

```

example_problem = 8

# Importing Facility
fd = model_check.extract(example_problem)
def func_KB(i,p):
    return fd.KB[i][p]
def func_L(i):
    return fd.L_mi[min(fd.NS-1, max(i, 0))]

##### Creating Gurobi Model
# Initializing model instance
hcm = gbp.Model("hcm-test")

##### Creating Segment Variables #####
SFv = [] # Segment flow out of segment i during step t in interval p
ASF = [] # Maximum Allowable Segment flow (replaces MO2)
KQ = [] # Queue density: vehicle density in the queue on segment i in step t in interval p
NVv = [] # NV in segment i at step t in interval p
UVv = [] # Unserved vehicles stored in segment i at the end of step t in interval p
NV_delta = [] # Variables to minimize the difference between observed and computed NV
UV_delta = [] # Variables to minimize the difference between observed and computed UV
SFv_avg = [] # Variables to compute average Segment Flow
NVv_avg = [] # Variables to compute average Number of vehicles
Kv = [] # Segment performance measure density

for el_i in xrange(fd.NS):
    KQ.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='KQ'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    SFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='SF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ASF.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='ASF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    NVv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='NV'+str(el_i)+str(el_t-1)+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S+1)])
    UVv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='UV'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    NV_delta.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='NV_delta'+str(el_i)+str(el_t)+str(el_p))
                    for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    UV_delta.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='UV_delta'+str(el_i)+str(el_t)+str(el_p))
                    for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    SFv_avg.append([hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='SFv_avg'+str(el_i)+'_'+str(el_p)) for el_p in xrange(fd.P)])
    NVv_avg.append([hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='NVv_avg'+str(el_i)+'_'+str(el_p)) for el_p in xrange(fd.P)])
    Kv.append([hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='Kv'+str(el_i)+'_'+str(el_p)) for el_p in xrange(fd.P)])

##### Unserved Vehicles Indicator #####
I_UVv = [] # Array to hold binary indicator
for el_i in xrange(fd.NS): # Segment Variable
    I_UVv.append([])
    for el_t in xrange(fd.S):

```

```

I_UVv[el_i].append([])
for el_p in xrange(fd.P):
    I_UVv[el_i][el_t].append(hcm.addVar(vtype=gbp.GRB.BINARY,
                                        name="I_UV"+str(el_i)+'_'+str(el_t)+'_'+str(el_p)))

##### Creating Expressions for Segment Variables
def SF(i, t, p):
    if t < 0:
        if p == 0:
            return min(fd.SD[max(i,0)][0], fd.SC[max(i-1,0)][0]) (1/fd.Th)
        else:
            return SF(i, fd.S+t, p-1)
    else:
        return SFv[i][t][p]

def NV(i, t, p):
    return NVv[i][t+1][p] # Accounts for t=-1 for initialization

def UV(i, t, p):
    if i < 0:
        return 0.0 # Denied entry is lost
    if p < 0 or (p == 0 and t < 0):
        return 0
    elif t < 0:
        return UV(i, fd.S+t, p-1)
    else:
        return UVv[i][t][p]

def I_UV(i, t, p):
    if p < 0 or (p == 0 and t < 0):
        return 0
    elif t < 0:
        return I_UV(i, fd.S+t, p-1)
    else:
        return I_UVv[i][t][p]

#####
##### Creating Flow Node Variables #####
MFv = [] # Actual mainline flow rate in node i during step t in interval p
ONRFv = [] # Actual ONR flow rate that can cross on ramp node i during step t in interval p
ONRQv = [] # Unment demand that is stored as a queue on the ONR node i during step t in interval p
OFRFv = [] # Actual flow that can exit at OFR node i during step t in interval p
MI = [] # Maximum mainline input: max flow desiring to enter segment i during step t in interval p
MO1v = [] # Max mainline output 1: limited by ONR flow at segment i
MO2v = [] # Max mainline output 2: limited by storage on segment i due to a downstream queue
MO3v = [] # Max mainline output 3: limited by the presence of queued veh at the upstream
DEF = [] # Deficit in flow at segment i at time step t in interval p
ODEF = []

for el_i in xrange(fd.NS+1): ## Node variables bookend each segment, thus there are NS+1 nodes
    MFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                            name='MF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='ONRF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRQv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='ONRQ'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])

```

```

        for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
OFRFv.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                           name='OFRF'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                           for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])
DEF.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                          name='DEF'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                          for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])
ODEF.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                           name='ODEF'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                           for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])
MI.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                         name='MI'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                         for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])
MO1v.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                           name='MO1'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                           for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])
MO2v.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                           name='MO2'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                           for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])
MO3v.append ([[hcm.addVar (vtype=gbp.GRB.CONTINUOUS,
                           name='MO3'+str (el_i)+'_'+str (el_t)+'_'+str (el_p))
                           for el_p in xrange (fd.P)] for el_t in xrange (fd.S)])

##### Creating Expressions for Node Variables
def MF(i, t, p):
    if i < 0:
        return min (fd.mainline_demand[p], fd.SC[0][p]) (1 / fd.Th)
    elif t < 0:
        if p == 0:
            return min (fd.SD[max(i-1,0)][0], fd.SC[i][p]) (1 / fd.Th)
        else:
            return MF(i, fd.S+t, p-1)
    else:
        return MFv[i][t][p]

def ONRF(i, t, p):
    if i < 0:
        return 0.0
    elif t < 0:
        if p == 0:
            return fd.ONRD[i][0] / fd.Th
        else:
            return ONRF(i, fd.S+t, p-1)
    else:
        return ONRFv[i][t][p]

def ONRQ(i, t, p):
    if t < 0:
        if p == 0:
            return 0
        else:
            return ONRQ(i, fd.S-1, p-1)
    else:
        return ONRQv[i][t][p]

def OFRF(i, t, p):
    if i < 0:
        return 0.0
    elif t < 0:
        if p == 0:

```

```

        return 0
    else:
        return OFRF(i, fd.S+t, p-1)
else:
    return OFRFv[i][t][p]

def MO1(i, t, p):
    if t < 0:
        if p == 0:
            return fd.SC[max(0,min(fd.NS-1,i))][p] (1 / fd.Th)
        else:
            return MO1(i, fd.S+t, p-1)
    else:
        return MO1v[i][t][p]

def MO2(i, t, p):
    if t < 0:
        if p == 0:
            return fd.SC[max(0,min(fd.NS-1,i))][p] (1 / fd.Th)
        else:
            return MO2(i, fd.S+t, p-1)
    else:
        return MO2v[i][t][p]

def MO3(i, t, p):
    if t < 0:
        if p == 0:
            return fd.SC[max(0,min(fd.NS-1,i))][p] (1 / fd.Th)
        else:
            return MO3(i, fd.S+t, p-1)
    else:
        return MO3v[i][t][p]

#####

##### Creating ONR Node Only Variables #####
ONRO = [] # Max output flow that can enter the merge from ONR node i during step t in interval p
ONRI = [] # Input flow desiring to enter the merge point at ONR node i during step t in interval p
ONRO_sMax = []
ONRO_sMin = []
for el_i in xrange(len(fd.Ntilde)):
    ONRI.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                            name='ONRI'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                            for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRO.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                            name='ONRO'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                            for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRO_sMax.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                                  name='ONRO_sMax'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRO_sMin.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                                  name='ONRO_sMin'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])

#####

##### Creating OFR Node Only Variables #####
#####

# Integrating variables into model
hcm.update()

```

```

##### Capacity Constraints #####
# SC = fd["SC"] # Retrieves constant capacities, no cap drop at step level
# Returned variable will always be a known constant
def func_SC(i,t,p):
    if t < 0:
        return func_SC(i, fd.S+t, p-1)
    else:
        return fd.SC[max(0,min(fd.NS-1, i))][max(p,0)]/fd.Th
#####

##### Front Clearing Queue #####
# If [SC(i,p)-ONRD(i,p)] > [SC(i,p-1)-ONRD(i,p-1)] && [SC(i,p)-ONRD(i,p)] > SD(i,p)
front_clearing_queue_present = []
for el_i in xrange(fd.NS+1): # fd.NS+1 to account for final node
    if el_i == fd.NS:
        front_clearing_queue_present.append([False for el_p in xrange(fd.P)])
    else:
        front_clearing_queue_present.append([])
        for el_p in xrange(fd.P):
            ##### FCC #####
            if el_p == 0:
                front_clearing_queue_present[el_i].append(False)
            else:
                isFCC = ((fd.SC[el_i][el_p] - fd.ONRD[el_i][el_p]) >
                    (fd.SC[el_i][el_p-1]-fd.ONRD[el_i][el_p-1])) and
                    (fd.SC[el_i][el_p]-fd.ONRD[el_i][el_p] > fd.SD[el_i][el_p])
                front_clearing_queue_present[el_i].append(False)
#####

##### BEGIN MODEL BUILD #####
init_time = time.time()
# Setting objective - Minimize number of vehicles
hcm.setObjective
    gbp.quicksum(gbp.quicksum(gbp.quicksum((fd.NS - el_i)/fd.NS (NV(el_i, el_t, el_p))
    + ONRQ(el_i, el_t, el_p) + ODEF[el_i][el_t][el_p] for el_p in xrange(fd.P))
    for el_t in xrange(fd.S)) for el_i in xrange(fd.NS)), gbp.GRB.MINIMIZE

hcm.update()

##### Init NV #####
#Initialize the Freeway Facility
for el_i in xrange(fd.NS): # NV is a Segment Quantity
    for el_p in xrange(fd.P):
        hcm.addConstr(NV(el_i, -1, el_p) ==
            func_KB(el_i, el_p) func_L(el_i) + UV(el_i, fd.S-1, el_p-1),
            name='NV_Init_'+str(el_i)+str(0)+str(el_p))
print("Init_NV_done")
#####

##### Demand/Flow Conservation #####
# Total Input/Output Conservation
hcm.addConstr(gbp.quicksum(gbp.quicksum(MF(fd.NS, el_t, el_p) + gbp.quicksum(OFRF(el_i, el_t, el_p)
for el_i in xrange(fd.NS+1)) for el_t in xrange(fd.S)) for el_p in xrange(fd.P))
    <= sum(fd.mainline_demand)/4.0 + sum(sum(fd.ONRD))/4.0,
    name='Demand_Flow_Conservation')

# On-Ramp Flow
for el_i in fd.Ntilde:

```

```

hcm.addConstr(gbp.quicksum(gbp.quicksum(gbp.quicksum(ONRF(el_i, el_t, el_p)
for el_p in xrange(fd.P)) for el_t in xrange(fd.S)) for el_i in xrange(fd.NS)) ==
sum(sum(fd.ONRD))/4.0, name='ONRF_Conservation')

##### Begin loop over all nodes #####
for el_p in xrange(fd.P):
    for el_t in xrange(fd.S):
        for el_i in xrange(fd.NS+1):
            ##### Off-Ramp Flow #####
            # Note: OFRF exists at the node at the downstream end of an OFR segment
            # If i is OFR Seg, OFRF is at node i+1 (node at downstream end of segment)
            if el_i-1 in fd.Ftilde: # Check if OFR at node. If i is OFR Seg, OFRF is at node i+1
                # Upper bound based on OFRD and deficit in OFRF from previous periods/steps
                hcm.addConstr(OFRF(el_i, el_t, el_p) ==
                    fd.func_TP(el_i-1, el_t, el_p) (fd.SD[el_i-1][el_p])) # fd.SD[el_i-1][el_p]
            else: # The segment is not an offramp
                hcm.addConstr(OFRF(el_i, el_t, el_p) == 0.0,
                    name='OFRF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(DEF[el_i][el_t][el_p] == 0.0,
                    name='DEF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
            #####

            ##### On-Ramp Flow #####
            # Note: ONRF exists at the node at the upstream end of an ONR segment
            if el_i in fd.Ntilde:
                onr_i = fd.Ntilde.index(el_i) # Convert segment idx to onr var idx
                ##### On-Ramp Input #####
                # ONRI = ONRD + ONRQ
                hcm.addConstr(ONRI[onr_i][el_t][el_p] ==
                    fd.ONRD[el_i][el_p] (1/fd.Th) + ONRQ(el_i, el_t-1, el_p),
                    name='ONRI_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                ##### On-Ramp Output #####
                # ONRO = min(RM, ONRC, sMax)
                hcm.addConstr(ONRO[onr_i][el_t][el_p] <= fd.RM[el_i][el_p],
                    name='ONRO_MIN_RM'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO[onr_i][el_t][el_p] <= fd.func_ONRC(el_i, el_t, el_p)/fd.Th,
                    name='ONRO_MIN_ONRC'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO[onr_i][el_t][el_p] <= ONRO_sMax[onr_i][el_t][el_p],
                    name='ONRO_MIN_sMAX'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                # Sub Minimum: sMin = Min(SC, MF + ONRF, MO3 + ONRF)
                hcm.addConstr(ONRO_sMin[onr_i][el_t][el_p] <= func_SC(el_i, el_t, el_p),
                    name='ONRO_sMIN1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO_sMin[onr_i][el_t][el_p] <=
                    MF(el_i+1, el_t-1, el_p) + ONRF(el_i, el_t-1, el_p),
                    name='ONRO_sMIN2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO_sMin[onr_i][el_t][el_p] <=
                    MO3(el_i, el_t-1, el_p) + ONRF(el_i, el_t-1, el_p),
                    name='ONRO_sMIN3_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                # Sub Maximum: sMax = Max(sMin - MI, sMin/(2 numLanes))
                hcm.addConstr(ONRO_sMax[onr_i][el_t][el_p] >=
                    ONRO_sMin[onr_i][el_t][el_p] - MI[el_i][el_t][el_p],
                    name='ONRO_sMAX1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO_sMax[onr_i][el_t][el_p] >=
                    ONRO_sMin[onr_i][el_t][el_p] (1.0/(2 fd.NL[el_i][el_p])),
                    name='ONRO_sMAX2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                ##### ONRF & ONRQ #####
                # ONRF = min(ONRI, ONRO)
                hcm.addConstr(ONRF(el_i, el_t, el_p) <= ONRI[onr_i][el_t][el_p],
                    name='ONRF_Min_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRF(el_i, el_t, el_p) <= ONRO[onr_i][el_t][el_p],

```

```

        name='ONRF_MinO_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p)
# Update ONRQ
hcm.addConstr(ONRQ(el_i, el_t, el_p) ==
    gbp.quicksum(fd.ONRD[el_i][x]/4.0 - gbp.quicksum(ONRF(el_i, tau, x)
    for tau in xrange(fd.S)) for x in xrange(el_p))
    + gbp.quicksum(fd.ONRD[el_i][el_p]/fd.Th - ONRF(el_i, tau, el_p)
    for tau in xrange(el_t)),
    name='ONRQ_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
else: # Segment is not ONR
    hcm.addConstr(ONRF(el_i, el_t, el_p) == 0.0,
        name='ONRF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(ONRQ(el_i, el_t, el_p) == 0.0)
#####

##### Mainline Input #####
# Step 9: Calculate Mainline Input
hcm.addConstr(MI[el_i][el_t][el_p] == # Mainline Input at node i equals
    MF(el_i-1, el_t, el_p) # Mainline Flow at upstream node plus
    + ONRF(el_i-1, el_t, el_p) # ONR Flow at upstream node minus
    - OFRF(el_i, el_t, el_p) # OFR flow at current node i
    + UV(el_i-1, el_t-1, el_p), # UV of upstream segment at the prev step
    name='MI_E_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####

##### Mainline Output 1 #####
# Calculate Mainline Output 1: MO1 = min (SC-ONRF, MO2, MO3)
hcm.addConstr(MO1(el_i, el_t, el_p) <=
    func_SC(el_i, el_t, el_p) - ONRF(el_i, el_t, el_p),
    name='MO1_Min1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MO1(el_i, el_t, el_p) <= MO2(el_i, el_t-1, el_p),
    name='MO1_Min2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MO1(el_i, el_t, el_p) <= MO3(el_i, el_t-1, el_p),
    name='MO1_Min3_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####

##### Mainline Output 2 #####
##### MO2/ASF #####
# Allowable Segment Flow
if el_i < fd.NS: # Not used for final node
    hcm.addConstr(KQ[el_i][el_t][el_p] ==
        fd.KJ fd.NL[el_i][el_p] - (fd.NL[el_i][el_p] (fd.KJ-fd.KC))
        SF(el_i, el_t-1, el_p)/(fd.SC[el_i][el_p]/fd.Th),
        name='KQ_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(ASF[el_i][el_t][el_p] ==
        (fd.KJ fd.NL[el_i][el_p] -
        (NV(el_i, el_t-1, el_p)+NV(el_i, el_t, el_p))/(2 func_L(el_i)))
        (func_SC(el_i, el_t, el_p)/(fd.NL[el_i][el_p] (fd.KJ-fd.KC))),
        name='ASF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MO2(el_i, el_t, el_p) ==
        ASF[el_i][el_t][el_p] - OFRF(el_i, el_t, el_p),
        name='MO2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
else: # Assume there is no queue downstream of facility end
    hcm.addConstr(MO2(el_i, el_t, el_p) == fd.SC[fd.NS-1][el_p]/fd.Th,
        name='MO2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####

##### MO3 #####
if front_clearing_queue_present[el_i][el_p] == False:
    # If there is no front clearing queue, this value is set to segment capacity
    # It is always considered FCC = False at the final downstream node

```

```

        hcm.addConstr(MO3(el_i, el_t, el_p) == func_SC(el_i, el_t, el_p),
                    name="MO3_NFCQ"+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    else:
        # MO3(i, t, p) <= MO1(i+1, t-WTT, p) - ONRF(i, t, p)
        # MO3(i, t, p) <= MO2(i+1, t-WTT, p) + OFRF(i+1, t-WTT, p) - ONRF(i, t, p)
        # MO3(i, t, p) <= MO3(i+1, t-WTT, p) + OFRF(i+1, t-WTT, p) - ONRF(i, t, p)
        # MO3(i, t, p) <= SC(i, t-WTT, p) - ONRF(i, t, p)
        # MO3(i, t, p) <= SC(i+1, t-WTT, p) + OFRF(i+1, t-WTT, p) - ONRF(i, t, p)
        hcm.addConstr(MO3(el_i, el_t, el_p) <=
                    MO1(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
                    name='MO3_m1'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MO3(el_i, el_t, el_p) <=
                    MO2(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) +
                    OFRF(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
                    name='MO3_m2'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MO3(el_i, el_t, el_p) <=
                    MO3(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) +
                    OFRF(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
                    name='MO3_m3'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MO3(el_i, el_t, el_p) <=
                    func_SC(el_i, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
                    name='MO3_m4'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MO3(el_i, el_t, el_p) <=
                    func_SC(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) +
                    OFRF(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
                    name='MO3_m5'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    #####

    ##### Mainline Flow #####
    #MF(i, t, p) <= MI(i, t, p)
    #MF(i, t, p) <= MO1(i, t, p)
    #MF(i, t, p) <= MO2(i, t, p)
    #MF(i, t, p) <= MO3(i, t, p)
    #MF(i, t, p) <= SC(i, t, p)
    #MF(i, t, p) <= SC(i-1, t, p)
    if (el_i==0):
        hcm.addConstr(MF(el_i, el_t, el_p) == MI[el_i][el_t][el_p],
                    name='MF_m1'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    else:
        hcm.addConstr(MF(el_i, el_t, el_p) <= MI[el_i][el_t][el_p],
                    name='MF_m1'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MF(el_i, el_t, el_p) <= MO1(el_i, el_t, el_p),
                    name='MF_m2'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MF(el_i, el_t, el_p) <= MO2(el_i, el_t, el_p),
                    name='MF_m3'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MF(el_i, el_t, el_p) <= MO3(el_i, el_t, el_p),
                    name='MF_m4'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MF(el_i, el_t, el_p) <= func_SC(el_i, el_t, el_p),
                    name='MF_m5'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
        hcm.addConstr(MF(el_i, el_t, el_p) <= func_SC(el_i-1, el_t, el_p),
                    name='MF_m6'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    #####

    print("Node_Constraints_Added")

    ##### Segment Constraints #####
    # Segment flow is equal to the mainline flow plus off-ramp flow of node at downstream end
    for el_i in xrange(fd.NS):
        for el_t in xrange(fd.S):
            for el_p in xrange(fd.P):
                ##### Max Density #####

```

```

hcm.addConstr(NV(el_i, el_t, el_p) <= fd.KJ fd.NL[el_i][el_p] func_L(el_i),
              name = 'KJ_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
##### Segment FLow #####
hcm.addConstr(SF(el_i, el_t, el_p) == MF(el_i+1, el_t, el_p) + OFRF(el_i+1, el_t, el_p),
              name="3.112"+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
##### Update NV and UV #####
# NV(i, t, p) = NV(i, t-1, p) + MF(i, t, p) + ONRF(i, t, p) - MF(i+1, t, p) - OFRF(i+1, t, p)
# UV(i, t, p) = NV(i, t, p) - [KB(i, t, p) L(i)]
##### NV #####
hcm.addConstr(NV(el_i, el_t, el_p) == NV(el_i, el_t-1, el_p) + MF(el_i, el_t, el_p)
              + ONRF(el_i, el_t, el_p) -
              MF(el_i+1, el_t, el_p) - OFRF(el_i+1, el_t, el_p),
              name="3.113" + str(el_i)+'_'+str(el_t)+'_'+str(el_p))
##### UV #####
hcm.addConstr(UV(el_i, el_t, el_p) ==
              NV(el_i, el_t, el_p) - func_KB(el_i, el_p) func_L(el_i),
              name="3.114"+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
##### UV Check #####
print("Segment_Constraints_Added")

##### Segment Performance Measures #####
for el_i in xrange(fd.NS):
    for el_p in xrange(fd.P):
        hcm.addConstr(SFv_avg[el_i][el_p] ==
                      (fd.Th/fd.S) gbp.quicksum(SF(el_i, tau, el_p) for tau in xrange(fd.S)),
                      name='SFv_avg_'+str(el_i)+'_'+str(el_p))
        hcm.addConstr(NVv_avg[el_i][el_p] ==
                      (1.0/fd.S) gbp.quicksum(NV(el_i, tau, el_p) for tau in xrange(fd.S)),
                      name='NVv_avg_'+str(el_i)+'_'+str(el_p))
        hcm.addConstr(Kv[el_i][el_p] == NVv_avg[el_i][el_p]/fd.L_mi[el_i],
                      name='Kv_'+str(el_i)+'_'+str(el_p))
print("Segment_Performance_Measure_Constraints_Added")
#####

hcm.update()
model_build_time=time.time()
print("Model_Built:_" +str(model_build_time - init_time))

hcm.optimize()
optimize_finish_time = time.time()
print("Model_Solved:_" +str(optimize_finish_time - model_build_time))

f=open('gpex1_ht_out.csv', 'w')
f.write('Segment, _Period, _Step, _NV, _MF, _MI, _MO1, _MO2, _MO3, _ONRF, _OFRF, _DEF, _UV, _SF, _KQ, _ASF\n')
varCount = 0
for p in xrange(fd.P):
    for t in xrange(fd.S):
        for i in xrange(fd.NS):
            varCount+=1
            #s = str(varCount)
            s = str(i)
            s += ",_" + str(p)
            s += ",_" + str(t)
            s += ",_" + str(NV(i, t, p).X)
            s += ",_" + str(MF(i+1, t, p).X)
            s += ",_" + str(MI[i+1][t][p].X)
            s += ",_" + str(MO1(i+1, t, p).X)
            s += ",_" + str(MO2(i+1, t, p).X)

```

```

s += ",_" + str(MC3(i+1, t, p).X)
s += ",_" + str(ONRF(i+1,t, p).X)
#s+= ", " + str(ONRQ(i, t, p).X)
s += ",_" + str(OFRF(i+1,t, p).X)
s += ",_" + str(DEF[i+1][t][p].X)
s+= ",_" + str(UV(i, t, p).X)
s+=",_" + str(SF(i, t, p).X)
s+=",_" + str(KQ[i][t][p].X)
s+=",_" + str(ASF[i][t][p].X)
f.write(s +'\n')
#print(s)

f.close()

# Computing Performance Measures
segFlow = zeros((fd.NS, fd.P))
numVeh = zeros((fd.NS, fd.P))
K = zeros((fd.NS, fd.P))
U = zeros((fd.NS, fd.P))
recalc_V = zeros((fd.NS, fd.P))
print("\nSPEED_DIFFERENCES")
boxes1 = [0 for el in xrange(fd.NS)]
boxes5 = [0 for el in xrange(fd.NS)]
boxes10 = [0 for el in xrange(fd.NS)]
for p in xrange(fd.P):
    for i in xrange(fd.NS):
        if abs(U[i][p] - recalc_V[i][p]) > 1:
            boxes1[i]+=1
            if abs(U[i][p] - recalc_V[i][p]) > 5:
                boxes5[i]+=1
                if abs(U[i][p] - recalc_V[i][p]) > 10:
                    boxes10[i]+=1

x = [el for el in xrange(fd.NS)]
fig = plt.figure(1)
plt.plot(x, boxes1, figure = fig, label='box1')
plt.plot(x, boxes5, figure = fig, label='box5')
plt.plot(x, boxes10, figure = fig, label='box10')
plt.legend(loc=2)
#plt.savefig('abs_err_i40.png')
plt.show()

abs_error = zeros((fd.NS,))
for p in xrange(fd.P):
    for i in xrange(fd.NS):
        segFlow[i][p] = (fd.Th/fd.S) sum([SF(i, tau, p).X for tau in xrange(fd.S)])
        numVeh[i][p] = (1.0/fd.S) sum([NV(i, tau, p).X for tau in xrange(fd.S)])
        K[i][p] = numVeh[i][p]/fd.L_mi[i]
        U[i][p] = segFlow[i][p]/K[i][p]
        temp_segFlow=(fd.Th/fd.S) sum([fd.fSF[i][tau][p] for tau in xrange(fd.S)])
        temp_numVeh = (1.0/fd.S) sum([fd.fNV[i][tau][p] for tau in xrange(fd.S)])
        recalc_V[i][p]=temp_segFlow/(temp_numVeh/fd.L_mi[i])
        abs_error[i]+=abs(U[i][p]-recalc_V[i][p])
        #sum_error+=U[i][p]-recalc_V[i][p]

x = [el for el in xrange(1,12)]
y = [abs_error[el]/12.0 for el in xrange(11)]
fig = plt.figure(1)
#plt.plot(x,y, figure = fig)
plt.bar(x, y, figure = fig)

```

```

plt.title('Average_Absolute_Speed_Difference')
plt.ylabel('Speed_(mph)')
plt.xlabel('Segment')
plt.legend(loc=2)
plt.savefig('gpex_avg_speed_diff.png')
plt.show()

import time
import model_check
from numpy import zeros
import matplotlib.pyplot as plt
import gurobipy as gbp
__author__ = 'jltrask'

xrange = range

example_problem = 23
onrq_capacity_ln = 50
out_fname='RMTTest/i40_hc_'+str(onrq_capacity_ln)+'_rm_rates.csv'

# Importing Facility
fd = model_check.extract(example_problem)
def func_KB(i,p):
    return fd.KB[i][p]
def func_L(i):
    return fd.L_mi[min(fd.NS-1, max(i, 0))]

##### Creating Gurobi Model
# Initializing model instance
hcm = gbp.Model("hcm-test")

##### Creating Segment Variables #####
SFv = [] # Segment flow out of segment i during step t in interval p
ASF = [] # Maximum Allowable Segment flow (replaces MQ2)
KQ = [] # Queue density: vehicle density in the queue on segment i in step t in interval p
NVv = [] # NV in segment i at step t in interval p
UVv = [] # Unserved vehicles stored in segment i at the end of step t in interval p
NV_delta = [] # Variables to minimize the difference between observed and computed NV
UV_delta = [] # Variables to minimize the difference between observed and computed UV
SFv_avg = [] # Variables to compute average Segment Flow
NVv_avg = [] # Variables to compute average Number of vehicles
Kv = [] # Segment performance measure density

for el_i in xrange(fd.NS):
    KQ.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='KQ'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    SFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='SF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ASF.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='ASF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    NVv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='NV'+str(el_i)+str(el_t-1)+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S+1)])
    UVv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='UV'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))

```

```

        for el_p in xrange(fd.P)] for el_t in xrange(fd.S)]]
NV_delta.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                             name='NV_delta'+str(el_i)+str(el_t)+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
UV_delta.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                             name='UV_delta'+str(el_i)+str(el_t)+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
SFv_avg.append([hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                           name='SFv_avg'+str(el_i)+'_'+str(el_p)) for el_p in xrange(fd.P)])
NVv_avg.append([hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                           name='NVv_avg'+str(el_i)+'_'+str(el_p)) for el_p in xrange(fd.P)])
Kv.append([hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                     name='Kv'+str(el_i)+'_'+str(el_p)) for el_p in xrange(fd.P)])

##### Unservd Vehicles Indicator #####
I_UVv = [] # Array to hold binary indicator
for el_i in xrange(fd.NS): # Segment Variable
    I_UVv.append([])
    for el_t in xrange(fd.S):
        I_UVv[el_i].append([])
        for el_p in xrange(fd.P):
            I_UVv[el_i][el_t].append(hcm.addVar(vtype=gbp.GRB.BINARY,
                                                name="I_UV"+str(el_i)+'_'+str(el_t)+'_'+str(el_p)))

##### Creating Expressions for Segment Variables
def SF(i, t, p):
    if t < 0:
        if p == 0:
            return min(fd.SD[max(i, 0)][0], fd.SC[max(i-1, 0)][0]) (1/fd.Th)
        else:
            return SF(i, fd.S+t, p-1)
    else:
        return SFv[i][t][p]

def NV(i, t, p):
    return NVv[i][t+1][p] # Accounts for t=-1 for initialization

def UV(i, t, p):
    if i < 0:
        return 0.0 # Denied entry is lost
    if p < 0 or (p == 0 and t < 0):
        return 0
    elif t < 0:
        return UV(i, fd.S+t, p-1)
    else:
        return UVv[i][t][p]

def I_UV(i, t, p):
    if p < 0 or (p == 0 and t < 0):
        return 0
    elif t < 0:
        return I_UV(i, fd.S+t, p-1)
    else:
        return I_UVv[i][t][p]

#####

##### Creating Flow Node Variables #####
MFv = [] # Actual mainline flow rate in node i during step t in interval p
ONRFv = [] # Actual ONR flow rate that can cross on ramp node i during step t in interval p

```

```

ONRQv = [] # Unmet demand stored as a queue on the ONR at node i during step t in interval p
OFRFv = [] # Actual flow that can exit at OFR node i during step t in interval p
MI = [] # Maximum mainline input: max flow desiring to enter segment i during step t in interval p
MO1v = [] # Max mainline output 1: limited by ONR flow at segment i
MO2v = [] # Max mainline output 2: limited by storage on segment i due to a downstream queue
MO3v = [] # Max mainline output 3: limited by the presence of queued vech at the upstream
DEF = [] # Deficit in flow at segment i at time step t in interval p
ODEF = []

```

```

for el_i in xrange(fd.NS+1): ## Node variables bookend each segment, thus there are NS+1 nodes
    MFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                           name='MF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='ONRF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ONRQv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='ONRQ'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    OFRFv.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='OFRF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                  for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    DEF.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                            name='DEF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    ODEF.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                             name='ODEF'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                 for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    MI.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                          name='MI'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
               for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    MO1v.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                             name='MO1'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                 for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    MO2v.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                             name='MO2'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                 for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
    MO3v.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                             name='MO3'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                 for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])

```

Creating Expressions for Node Variables

```

def MF(i, t, p):
    if i < 0:
        return min(fd.mainline_demand[p], fd.SC[0][p]) (1 / fd.Th)
    elif t < 0:
        if p == 0:
            return min(fd.SD[max(i-1,0)][0], fd.SC[i][p]) (1 / fd.Th)
        else:
            return MF(i, fd.S+t, p-1)
    else:
        return MFv[i][t][p]

def ONRF(i, t, p):
    if i < 0:
        return 0.0
    elif t < 0:
        if p == 0:
            return fd.ONRD[i][0] / fd.Th

```

```

        else:
            return ONRF(i, fd.S+t, p-1)
    else:
        return ONRFv[i][t][p]

def ONRQ(i, t, p):
    if t < 0:
        if p == 0:
            return 0
        else:
            return ONRQ(i, fd.S-1, p-1)
    else:
        return ONRQv[i][t][p]

def OFRF(i, t, p):
    if i < 0:
        return 0.0
    elif t < 0:
        if p == 0:
            return 0
        else:
            return OFRF(i, fd.S+t, p-1)
    else:
        return OFRFv[i][t][p]

def MO1(i, t, p):
    if t < 0:
        if p == 0:
            return fd.SC[max(0, min(fd.NS-1, i))][p] (1 / fd.Th)
        else:
            return MO1(i, fd.S+t, p-1)
    else:
        return MO1v[i][t][p]

def MO2(i, t, p):
    if t < 0:
        if p == 0:
            return fd.SC[max(0, min(fd.NS-1, i))][p] (1 / fd.Th)
        else:
            return MO2(i, fd.S+t, p-1)
    else:
        return MO2v[i][t][p]

def MO3(i, t, p):
    if t < 0:
        if p == 0:
            return fd.SC[max(0, min(fd.NS-1, i))][p] (1 / fd.Th)
        else:
            return MO3(i, fd.S+t, p-1)
    else:
        return MO3v[i][t][p]

#####

##### Creating ONR Node Only Variables #####
ONRO = [] # Max output flow that can enter the merge point from ONR i during step t in interval p
ONRI = [] # Input flow desiring to enter the merge point at ONR node i during step t in interval p
ONRO_sMax = []
ONRO_sMin = []
for el_i in xrange(len(fd.Ntilde)):

```

```

ONRI.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                        name='ONRI'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                        for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
ONRO.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                        name='ONRO'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                        for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
ONRO_sMax.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='ONRO_sMax'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                              for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
ONRO_sMin.append([[hcm.addVar(vtype=gbp.GRB.CONTINUOUS,
                              name='ONRO_sMin'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                              for el_p in xrange(fd.P)] for el_t in xrange(fd.S)])
#####

##### Creating OFR Node Only Variables #####
#####

# Integrating variables into model
hcm.update()

##### Capacity Constraints #####
# SC = fd["SC"] # Retrieves constant capacities, no cap drop at step level
# Returned variable will always be a known constant
def func_SC(i,t,p):
    if t < 0:
        return func_SC(i, fd.S+t, p-1)
    else:
        return fd.SC[max(0, min(fd.NS-1, i))][max(p,0)]/fd.Th
#####

##### Front Clearing Queue #####
# If [SC(i,p)-ONRD(i,p)]>[SC(i,p-1)-ONRD(i,p-1)] && [SC(i,p)-ONRD(i,p)] > SD(i,p)
front_clearing_queue_present = []
for el_i in xrange(fd.NS+1): # fd.NS+1 to account for final node
    if el_i == fd.NS:
        front_clearing_queue_present.append([False for el_p in xrange(fd.P)])
    else:
        front_clearing_queue_present.append([])
        for el_p in xrange(fd.P):
            ##### FCC #####
            if el_p == 0:
                front_clearing_queue_present[el_i].append(False)
            else:
                isFCC = ((fd.SC[el_i][el_p] - fd.ONRD[el_i][el_p]) >
                        (fd.SC[el_i][el_p-1]-fd.ONRD[el_i][el_p-1]))
                and (fd.SC[el_i][el_p]-fd.ONRD[el_i][el_p] > fd.SD[el_i][el_p])
                front_clearing_queue_present[el_i].append(False)
#####

##### BEGIN MODEL BUILD #####
init_time = time.time()
# Setting objective - Minimize number of vehicles
hcm.setObjective(
    gbp.quicksum(
        gbp.quicksum(
            gbp.quicksum((fd.NS - el_i)/fd.NS NV(el_i, el_t, el_p)

```

```

+ UV(el_i, el_t, el_p) + ONRQ(el_i, el_t, el_p)
+ ODEF[el_i][el_t][el_p]
for el_p in xrange(fd.P)) for el_t in xrange(fd.S)) for el_i in xrange(fd.NS)), gbp.GRB.MINIMIZE)
hcm.update()

##### Init NV #####
#Initialize the Freeway Facility
for el_i in xrange(fd.NS): # NV is a Segment Quantity
    for el_p in xrange(fd.P):
        hcm.addConstr(NV(el_i, -1, el_p) == func_KB(el_i, el_p) func_L(el_i)
            + UV(el_i, fd.S-1, el_p-1), name='NV_Init_'+str(el_i)+str(0)+str(el_p))
print("Init_NV_done")
#####

##### Demand/Flow Conervation #####

# On-Ramp Flow
for el_i in fd.Ntilde:
    hcm.addConstr(gbp.quicksum(gbp.quicksum(gbp.quicksum(ONRF(el_i, el_t, el_p)
        for el_p in xrange(fd.P)) for el_t in xrange(fd.S))
        for el_i in xrange(fd.NS)) == sum(sum(fd.ONRD))/4.0, name='ONRF_Conservation')

##### Begin loop over all nodes #####
for el_p in xrange(fd.P):
    for el_t in xrange(fd.S):
        for el_i in xrange(fd.NS+1):
            ##### Off-Ramp Flow #####
            # Note: OFRF exists at the node at the downstream end of an OFR segment
            # If i is OFR Seg, OFRF is at node i+1 (node at downstream end of segment)
            # Check if OFR at node. If i is OFR Seg, OFRF is at node i+1 (node at downstream end)
            if el_i-1 in fd.Ftilde:
                hcm.addConstr(OFRF(el_i, el_t, el_p)
                    == fd.func_TP(el_i-1, el_t, el_p) fd.SD[el_i-1][el_p])
            else: # The segment is not an offramp
                hcm.addConstr(OFRF(el_i, el_t, el_p) == 0.0,
                    name='OFRF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(DEF[el_i][el_t][el_p] == 0.0,
                    name='DEF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
            #####

            ##### On-Ramp Flow #####
            # Note: ONRF exists at the node at the upstream end of an ONR segment
            if el_i in fd.Ntilde:
                # Convert segment idx to onr var idx (for ONRO_A, ONRO_I, ONRF_I)
                onr_i = fd.Ntilde.index(el_i)
                ##### On-Ramp Input #####
                # ONRI = ONRD + ONRQ
                hcm.addConstr(ONRI[onr_i][el_t][el_p]
                    == fd.ONRD[el_i][el_p] (1/fd.Th) + ONRQ(el_i, el_t-1, el_p),
                    name='ONRI_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                ##### On-Ramp Output #####
                # ONRO = min(RM, ONRC, sMax)
                hcm.addConstr(ONRO[onr_i][el_t][el_p] <= fd.RM[el_i][el_p],
                    name='ONRO_MIN_RM'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO[onr_i][el_t][el_p] <= fd.func_ONRC(el_i, el_t, el_p)/fd.Th,
                    name='ONRO_MIN_ONRC'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                hcm.addConstr(ONRO[onr_i][el_t][el_p] <= ONRO_sMax[onr_i][el_t][el_p],
                    name='ONRO_MIN_sMAX'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
                # Sub Minimum: sMin = Min(SC, MF + ONRF, MOB + ONRF)
                hcm.addConstr(ONRO_sMin[onr_i][el_t][el_p] <= func_SC(el_i, el_t, el_p),

```

```

        name='ONRO_sMIN1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(ONRO_sMin[onr_i][el_t][el_p] <=
    MF(el_i+1, el_t-1, el_p) + ONRF(el_i, el_t-1, el_p),
    name='ONRO_sMIN2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(ONRO_sMin[onr_i][el_t][el_p] <=
    MO3(el_i, el_t-1, el_p) + ONRF(el_i, el_t-1, el_p),
    name='ONRO_sMIN3_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
# Sub Maximum: sMax = Max(sMin - MI, sMin/(2 numLanes))
hcm.addConstr(ONRO_sMax[onr_i][el_t][el_p] >=
    ONRO_sMin[onr_i][el_t][el_p] - MI[el_i][el_t][el_p],
    name='ONRO_sMAX1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(ONRO_sMax[onr_i][el_t][el_p] >=
    ONRO_sMin[onr_i][el_t][el_p] * (1.0/(2 * fd.NL[el_i][el_p])),
    name='ONRO_sMAX2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
##### ONRF & ONRQ #####
# ONRF = min(ONRI, ONRO)
hcm.addConstr(ONRF(el_i, el_t, el_p) <= ONRI[onr_i][el_t][el_p],
    name='ONRF_MinI_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(ONRF(el_i, el_t, el_p) <= ONRO[onr_i][el_t][el_p],
    name='ONRF_MinO_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
# Update ONRQ
hcm.addConstr(ONRQ(el_i, el_t, el_p) ==
    gbp.quicksum(fd.ONRD[el_i][x]/4.0 - gbp.quicksum(ONRF(el_i, tau, x)
    for tau in xrange(fd.S)) for x in xrange(el_p))
    + gbp.quicksum(fd.ONRD[el_i][el_p]/fd.Th - ONRF(el_i, tau, el_p)
    for tau in xrange(el_t)), name='ONRQ_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(ONRQ(el_i, el_t, el_p) <= onrq_capacity_ln fd.onr_nl[el_i][el_p])
else: # Segment is not ONR
    hcm.addConstr(ONRF(el_i, el_t, el_p) == 0.0,
        name='ONRF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(ONRQ(el_i, el_t, el_p) == 0.0)
#####
##### Mainline Input #####
# Step 9: Calculate Mainline Input
hcm.addConstr(MI[el_i][el_t][el_p] ==
    MF(el_i-1, el_t, el_p) # Mainline Input at node i equals
    + ONRF(el_i-1, el_t, el_p) # Mainline Flow at upstream node plus
    - OFRF(el_i, el_t, el_p) # ONR Flow at upstream node minus
    + UV(el_i-1, el_t-1, el_p), # OFR flow at current node i
    # UV of up segment at the prev time step
    name='MI_E_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####
##### Mainline Output 1 #####
# Calculate Mainline Output 1: MO1 = min(SC-ONRF, MO2, MO3)
hcm.addConstr(MO1(el_i, el_t, el_p) <=
    func_SC(el_i, el_t, el_p) - ONRF(el_i, el_t, el_p),
    name='MO1_Min1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MO1(el_i, el_t, el_p) <= MO2(el_i, el_t-1, el_p),
    name='MO1_Min2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MO1(el_i, el_t, el_p) <= MO3(el_i, el_t-1, el_p),
    name='MO1_Min3_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####
##### Mainline Output 2 #####
##### MO2/ASF #####
# Allowable Segment Flow
if el_i < fd.NS: # Not used for final node
    hcm.addConstr(KQ[el_i][el_t][el_p] ==
        fd.KJ * fd.NL[el_i][el_p] - (fd.NL[el_i][el_p])

```

```

        (fd.KJ-fd.KC)) SF(el_i, el_t-1, el_p)/(fd.SC[el_i][el_p]/fd.Th),
        name='KQ_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(ASF[el_i][el_t][el_p] ==
    (fd.KJ fd.NL[el_i][el_p] - (NV(el_i, el_t-1, el_p)+NV(el_i, el_t, el_p))/
    (2 func_L(el_i))) (func_SC(el_i, el_t, el_p)/(fd.NL[el_i][el_p] (fd.KJ-fd.KC))),
    name='ASF_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MO2(el_i, el_t, el_p) ==
    ASF[el_i][el_t][el_p] - ONRF(el_i, el_t, el_p),
    name='MO2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
else: # Assume there is no queue downstream of facility end
    hcm.addConstr(MO2(el_i, el_t, el_p) == fd.SC[fd.NS-1][el_p]/fd.Th,
        name='MO2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####

##### MO3 #####
if front_clearing_queue_present[el_i][el_p] == False:
    # If there is no front clearing queue, this value is set to segment capacity
    # It is always considered FCC = False at the final downstream node
    hcm.addConstr(MO3(el_i, el_t, el_p) == func_SC(el_i, el_t, el_p),
        name="MO3_NFCQ"+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
else:
    # MO3(i, t, p) <= MO1(i+1, t-WTT, p) - ONRF(i, t, p)
    # MO3(i, t, p) <= MO2(i+1, t-WTT, p) + OFRF(i+1, t-WTT, p) - ONRF(i, t, p)
    # MO3(i, t, p) <= MO3(i+1, t-WTT, p) + OFRF(i+1, t-WTT, p) - ONRF(i, t, p)
    # MO3(i, t, p) <= SC(i, t-WTT, p) - ONRF(i, t, p)
    # MO3(i, t, p) <= SC(i+1, t-WTT, p) + OFRF(i+1, t-WTT, p) - ONRF(i, t, p)
    hcm.addConstr(MO3(el_i, el_t, el_p) <=
        MO1(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
        name='MO3_m1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MO3(el_i, el_t, el_p) <= MO2(el_i+1, el_t-fd.WTT(el_i, el_p), el_p)
        + OFRF(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
        name='MO3_m2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MO3(el_i, el_t, el_p) <= MO3(el_i+1, el_t-fd.WTT(el_i, el_p), el_p)
        + OFRF(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
        name='MO3_m3_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MO3(el_i, el_t, el_p) <=
        func_SC(el_i, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
        name='MO3_m4_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MO3(el_i, el_t, el_p) <=
        func_SC(el_i+1, el_t-fd.WTT(el_i, el_p), el_p)
        + OFRF(el_i+1, el_t-fd.WTT(el_i, el_p), el_p) - ONRF(el_i, el_t, el_p),
        name='MO3_m5_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####

##### Mainline Flow #####
#MF(i, t, p) <= MI(i, t, p)
#MF(i, t, p) <= MO1(i, t, p)
#MF(i, t, p) <= MO2(i, t, p)
#MF(i, t, p) <= MO3(i, t, p)
#MF(i, t, p) <= SC(i, t, p)
#MF(i, t, p) <= SC(i-1, t, p)
if (el_i==0):
    hcm.addConstr(MF(el_i, el_t, el_p) == MI[el_i][el_t][el_p],
        name='MF_m1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
else:
    hcm.addConstr(MF(el_i, el_t, el_p) <= MI[el_i][el_t][el_p],
        name='MF_m1_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MF(el_i, el_t, el_p) <= MO1(el_i, el_t, el_p),
        name='MF_m2_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
    hcm.addConstr(MF(el_i, el_t, el_p) <= MO2(el_i, el_t, el_p),

```

```

        name='MF_m3_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MF(el_i, el_t, el_p) <= MO3(el_i, el_t, el_p),
        name='MF_m4_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MF(el_i, el_t, el_p) <= func_SC(el_i, el_t, el_p),
        name='MF_m5_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
hcm.addConstr(MF(el_i, el_t, el_p) <= func_SC(el_i-1, el_t, el_p),
        name='MF_m6_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
#####
print("Node_Constraints_Added")

##### Segment Constraints #####
# Segment flow is equal to the mainline flow plus off-ramp flow of node at downstream end
for el_i in xrange(fd.NS):
    for el_t in xrange(fd.S):
        for el_p in xrange(fd.P):
            ##### Max Density #####
hcm.addConstr(NV(el_i, el_t, el_p) <= fd.KJ fd.NL[el_i][el_p] func_L(el_i),
            name = 'KJ_'+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
            ##### Segment FLOW #####
            # i+1 -> uses nodes at downstream end of segment
hcm.addConstr(SF(el_i, el_t, el_p) == MF(el_i+1, el_t, el_p) + OFRF(el_i+1, el_t, el_p),
            name="3.112"+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
            ##### Update NV and UV #####
            # NV(i, t, p) = NV(i, t-1, p) + MF(i, t, p) + ONRF(i, t, p) - MF(i+1, t, p) - OFRF(i+1, t, p)
            # UV(i, t, p) = NV(i, t, p) - [KB(i, t, p) L(i)]
            ##### NV #####
hcm.addConstr(NV(el_i, el_t, el_p) == NV(el_i, el_t-1, el_p) + MF(el_i, el_t, el_p)
            + ONRF(el_i, el_t, el_p) - MF(el_i+1, el_t, el_p) -
            OFRF(el_i+1, el_t, el_p),
            name="3.113" + str(el_i)+'_'+str(el_t)+'_'+str(el_p))
            ##### UV #####
hcm.addConstr(UV(el_i, el_t, el_p) ==
            NV(el_i, el_t, el_p) - func_KB(el_i, el_p) func_L(el_i),
            name="3.114"+str(el_i)+'_'+str(el_t)+'_'+str(el_p))
            ##### UV Check #####
print("Segment_Constraints_Added")

##### Segment Performance Measures #####
for el_i in xrange(fd.NS):
    for el_p in xrange(fd.P):
        hcm.addConstr(SFv_avg[el_i][el_p] ==
            (fd.Th/fd.S) gbp.quicksum(SF(el_i, tau, el_p) for tau in xrange(fd.S)),
            name='SFv_avg_'+str(el_i)+'_'+str(el_p))
        hcm.addConstr(NVv_avg[el_i][el_p] ==
            (1.0/fd.S) gbp.quicksum(NV(el_i, tau, el_p) for tau in xrange(fd.S)),
            name='NVv_avg_'+str(el_i)+'_'+str(el_p))
        hcm.addConstr(Kv[el_i][el_p] == NVv_avg[el_i][el_p]/fd.L_mi[el_i],
            name='Kv_'+str(el_i)+'_'+str(el_p))
print("Segment_Performance_Measure_Constraints_Added")
#####

hcm.update()
model_build_time=time.time()
print("Model_Built:_" +str(model_build_time - init_time))

hcm.optimize()
optimize_finish_time = time.time()
print("Model_Solved:_" +str(optimize_finish_time - model_build_time))

```

```

f = open(out_fname, 'w')
rm_rates = zeros((fd.NS, fd.P))
for p in xrange(fd.P):
    s = ''
    for i in xrange(fd.NS):
        for t in xrange(fd.S):
            rm_rates[i][p]+=ONRF(i,t,p).X
            s = s + str(rm_rates[i][p] 4) +', '
        f.write(s+'\n')
f.close();

# Computing Performance Measures
segFlow = zeros((fd.NS, fd.P))
numVeh = zeros((fd.NS,fd.P))
K = zeros((fd.NS,fd.P))
U = zeros((fd.NS,fd.P))
recalc_V = zeros((fd.NS, fd.P))
print("\nSPEED_DIFFERENCES")
boxes1 = [0 for el in xrange(fd.NS)]
boxes5 = [0 for el in xrange(fd.NS)]
boxes10 = [0 for el in xrange(fd.NS)]
for p in xrange(fd.P):
    for i in xrange(fd.NS):
        if abs(U[i][p] - recalc_V[i][p]) > 1:
            boxes1[i]+=1
            if abs(U[i][p] - recalc_V[i][p]) > 5:
                boxes5[i]+=1
                if abs(U[i][p] - recalc_V[i][p]) > 10:
                    boxes10[i]+=1

x = [el for el in xrange(fd.NS)]
fig = plt.figure(1)
plt.plot(x, boxes1, figure = fig, label='box1')
plt.plot(x, boxes5, figure = fig, label='box5')
plt.plot(x, boxes10, figure = fig, label='box10')
plt.legend(loc=2)
#plt.savefig('abs_err_i40.png')
plt.show()

for p in xrange(fd.P):
    for i in xrange(fd.NS):
        segFlow[i][p] = (fd.Th/fd.S) sum([SF(i,tau,p).X for tau in xrange(fd.S)])
        numVeh[i][p] = (1.0/fd.S) sum([NV(i,tau,p).X for tau in xrange(fd.S)])
        K[i][p] = numVeh[i][p]/fd.L_mi[i]
        U[i][p] = segFlow[i][p]/K[i][p]
        temp_segFlow=(fd.Th/fd.S) sum([fd.fSF[i][tau][p] for tau in xrange(fd.S)])
        temp_numVeh = (1.0/fd.S) sum([fd.fNV[i][tau][p] for tau in xrange(fd.S)])
        recalc_V[i][p]=temp_segFlow/(temp_numVeh/fd.L_mi[i])

```