

12-11-2009

Dynamic Message Sign and Diversion Traffic Optimization

Jizhan Gou

Follow this and additional works at: <https://scholarsjunction.msstate.edu/td>

Recommended Citation

Gou, Jizhan, "Dynamic Message Sign and Diversion Traffic Optimization" (2009). *Theses and Dissertations*. 1589.
<https://scholarsjunction.msstate.edu/td/1589>

This Dissertation - Open Access is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact scholcomm@msstate.libanswers.com.

DYNAMIC MESSAGE SIGN AND DIVERSION TRAFFIC OPTIMIZATION

By

Jizhan Gou

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Civil Engineering
In the Department of Civil and Environmental Engineering

Mississippi State University

December 2009

DYNAMIC MESSAGE SIGN AND DIVERSION TRAFFIC OPTIMIZATION

By

Jizhan Gou

Approved:

Li Zhang
Professor of Civil and
Environmental Engineering
(Director of Dissertation)

Thomas D. White
Professor of Civil and
Environmental Engineering
(Committee member)

William H. McAnally
Associate Professor of Civil and
Environmental Engineering
(Committee member)

Mingzhou Jin
Associate Professor of Industrial
and Systems Engineering
(Committee member)

James L. Martin
Professor of Civil and
Environmental Engineering
Graduate Coordinator in the
Department of Civil and
Environmental Engineering

Sarah A. Rajala
Dean of the Bagley College of Engineering

Name: Jizhan Gou

Date of Degree: December 11, 2009

Institution: Mississippi State University

Major Field: Civil Engineering

Major Professor: Dr. Li Zhang

Title of Study: DYNAMIC MESSAGE SIGN AND DIVERSION TRAFFIC OPTIMIZATION

Pages in Study: 215

Candidate for Degree of Doctor of Philosophy

This dissertation proposes a Dynamic Message Signs (DMS) diversion control system based on principles of existing Advanced Traveler Information Systems and Advanced Traffic Management Systems (ATMS). The objective of the proposed system is to alleviate total corridor traffic delay by choosing optimized diversion rate and alternative road signal-timing plan. The DMS displays adaptive messages at predefined time interval for guiding certain number of drivers to alternative roads. Messages to be displayed on the DMS are chosen by an on-line optimization model that minimizes corridor traffic delay. The expected diversion rate is assumed following a distribution.

An optimization model that considers three traffic delay components: mainline travel delay, alternative road signal control delay, and the travel time difference between the mainline and alternative roads is constructed. Signal timing parameters of alternative road intersections and DMS message level are the decision variables; speeds, flow rates, and other corridor traffic data from detectors serve as inputs of the model.

Traffic simulation software, CORSIM, served as a developmental environment and test bed for evaluating the proposed system. MATLAB optimization toolboxes have been applied to solve the proposed model. A CORSIM Run-Time-Extension (RTE) has been developed to exchange data between CORSIM and the adopted MATLAB optimization algorithms (Genetic Algorithm, Pattern Search in direct search toolbox, and Sequential Quadratic Programming). Among the three candidate algorithms, the Sequential Quadratic Programming showed the fastest execution speed and yielded the smallest total delays for numerical examples.

TRANSYT-7F, the most credible traffic signal optimization software has been used as a benchmark to verify the proposed model. The total corridor delays obtained from CORSIM with the SQP solutions show average reductions of 8.97%, 14.09%, and 13.09% for heavy, moderate and light traffic congestion levels respectively when compared with TRANSYT-7F optimization results. The maximum model execution time at each MATLAB call is fewer than two minutes, which implies that the system is capable of real world implementation with a DMS message and signal update interval of two minutes.

Key Words: Dynamic Message Signs (DMS), Traffic Control, Integrated Corridor Management, CORSIM RTE, MATLAB, and Signal Optimization.

DEDICATION

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Li Zhang for his invaluable advice, guidance, and support during my doctoral research.

I also wish to express my thankfulness to other advisory committee members: Dr. Thomas D. White, Dr. William H. McAnally and Dr. Mingzhou Jin for their helpful comments, suggestions and support in my research.

I would like to extend my appreciation to Dr. Dennis D. Truax, Head of Civil and Environmental Engineering Department and, Dr. James L. Martin, Graduate Coordinator of Civil and Environmental Engineering Department for their encouragement and concern during my doctoral study.

Last but not the least, I would like to appreciate the Hearin Fellowship from Bagley College of Engineering for one-semester financial support.

TABLE OF CONTENTS

DEDICATION.....	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES.....	vii
LIST OF FIGURES.....	ix
ACRONYMS	xi
CHAPTER	
I. INTRODUCTION	1
Problem Statement	1
Objectives and Approaches	3
Objectives	3
Approaches	6
Significance of the Study	7
Scope and Limits of the Study	8
II. LITERATURE REVIEW	10
Real-Time Traffic Information System and Benefits	10
ATIS and AWIS.....	11
ADAPTIR	11
IntelliZone.....	12
TIPS	13
Real-Time CMS Control	14
CHIPS	15
Other ATIS system	16
Effectiveness and Benefits of ATIS.....	17
DMS Message Content and Driver Compliance	21
Signal Control and Optimization.....	24

Conventional Models.....	24
Signal Optimization	25
Adaptive Signal Control	27
Software Package TRANSYT-7F	32
Integrated Corridor Control.....	33
Summary of Literature Review	37
III. METHODOLOGY	40
Model Construction.....	40
System Parameters	41
Model Structure	42
Mainline Delay.....	43
Speed Prediction	49
Alternative Road Signal Control Delay	52
Alternative road Travel Time Difference	53
Decision Variables	54
State Variables.....	56
Driver Compliance.....	56
Solutions to Model	61
Algorithms Introduction.....	62
Sequential Quadratic Programming (SQP)	64
Pattern Search	68
Generic Algorithm	71
Algorithms Evaluation.....	75
IV. SYSTEM IMPLEMENTATION.....	80
System Structure	81
Techniques Employed in the Implementation	85
Rolling Horizon	85
Penalty Factor	87
Analytical Difference Equations Utilization in SQP Model Solution	88
Total Control Delay.....	89
Total Freeway Travel Delay	97
Total and Average Freeway Queue Delay	98
Detour Delay.....	101
Pseudo Code of System Implementation.....	102
CORSIM Call Points for RTE.....	102
RTE Coding in Visual Studio 2005	106
Model Coding in MATLAB.....	110

V.	RESULT ANALYSIS AND SYSTEM EVALUATION	113
	Cases and Scenarios	113
	Example Network	115
	CORSIM Running Control	122
	Variables	123
	Network State Variables	124
	Independent Decision Variables	124
	Constraints	126
	Case Implementations in RTE.....	129
	Functions Called at Call Points	130
	Model Validation	135
	Result Analysis	137
	CORSIM MOE Outputs.....	138
	Analysis of Single CORSIM Run	139
	Solution Analysis of Multiple CORSIM Runs	141
	Statistic Solution Analysis of Multiple CORSIM Runs	142
	Execution Time	153
	Diversion Volume to DMS Level Approximation Effects.....	161
	Application of Penalty Factor	161
	Application of Analytical Difference in SQP.....	163
	Summary	165
VI.	CONCLUSIONS AND FUTURE RESEARCH	167
	Conclusions	167
	Future Research.....	170
	REFERENCES.....	172
	APPENDIX	
	A. PSEUDO CODE OF SYSTEM IMPLEMENTATION.....	182
	B. EXAMPLE OF USER INPUT DATA FILE.....	199
	C. EXAMPLE OF RTE OUTPUT DATA FILE	202
	D. EXAMPLE OF MATLAB OUTPUT DATA FILE	205
	E. EXAMPLE OF CORSIM RTE RUNNING LOG	207
	F. MATLAB ALGORITHM OPTIONS FOR CASE STUDY.....	214

LIST OF TABLES

1	DMS Message Levels and Expected Diversion Probability	60
2	Comparison on Optimal Solutions from a Random Initial X_0 and an Optimal Solution X^* as an Initial X_0 for a New Search	76
3	Comparison on Execution Properties from a Random Initial X_0 and an Optimal Solution X^* as an Initial X_0 for a New Search.....	78
4	Model Constraints Implementation	128
5	CORSIM MOE Comparison between before and after SQP Optimization	144
6	CORSIM MOE Comparison between before and after Pattern Search Optimization	145
7	CORSIM MOE Comparison between before and after GA Optimization.....	146
8	CORSIM Total Delay Comparison among Three Algorithms – Single Run	147
9	SQP Total Delay Comparison between Single Run and Multiple Runs	148
10	SQP Average Speed Comparison between Single Run and Multiple Runs	149
11	Comparison of PS between Single Run and Multiple Runs, Moderate Congestion	150
12	Comparison of GA between Single Run and Multiple Runs, Moderate Congestion	150
13	Statistical CORSIM MOE Comparison among the Three Algorithms	152
14	Running Time Comparison on Different Optimization Execution Type	154
15	Execution Time Analysis of the Three Algorithms*	156
16	Multiple SQP RTE Running Time	157
17	Multiple Running Time of CORSIM.....	158

18	Changes from Diversion Volume to Diversion Level Approximation	159
19	Statistical Comparison of Execution Time among the Three Algorithms.....	160
20	Comparison between Finite Difference and Analytical difference Gradient in MATLAB SQP Execution Properties	164

LIST OF FIGURES

1	General System Modules.....	5
2	Illustration of Detector Effective Segment	46
3	Speed-Density Curve from Van Aerde Model (64)	50
4	Speed-Density Curve Fitting from CORSIM	51
5	Flow-Speed Data from CORSIM Case Study	51
6	Example PDF and CDF of an Extreme Distribution	58
7	Flowchart of Genetic Algorithm.....	73
8	System Structures of Modules.....	82
9	System Structure	83
10	System Data Flow	84
11	Illustration of Rolling Horizon Concept.....	86
12	Illustration of DMS and Signal Rolling Periods.....	87
13	Structure Illustration of CORSIM Call Points.....	104
14	System Implementation Structure	109
15	Example Network.....	117
16	Example Network Code in CORSIM	118
17	Geometric Information and Variable Number of Intersection Node 39.....	119
18	Geometric Information and Variable Number of Intersection Node 41.....	119

19	Geometric Information and Variable Number of Intersection Node 43.....	120
20	Geometric Information and Variable Number of Intersection Node 45.....	120
21	T Intersection, Two-Phase Signal Plan at Node 39	121
22	Four-Leg Intersection, Four-Phase Signal Plan at Node 41, 43, and 45	122
23	CORSIM RTE Configurations for Case Study	131
24	Illustration of SQP Optimization Progress of Signal Parameters and Diversion Volume.....	132
25	Illustration of SQP Optimization Progress of Signal Parameters	132
26	Illustration of Pattern Search Optimization Progress of Signal Parameters and Diversion Volume	133
27	Illustration of Pattern Search Optimization Progress of Signal Parameters	133
28	Illustration of GA Optimization Progress of Signal Parameters and Diversion Volume	134
29	Illustration of GA Optimization Progress of Signal Parameters.....	134
30	Demonstration of TRANSYT-7F Optimization Configuration	136
31	TRANSYT-7F Genetic Algorithm Parameters	137
32	Statistical MOE Comparisons among the Three Algorithms.....	151
33	Statistical Total Delay Comparisons among the Three Algorithms	152
34	Statistical Average Speed Comparisons among the Three Algorithms	153
35	Average Execution Time Comparisons between CORSIM and SQP RTE.....	160
36	Illustration of Pattern Search Optimization Progress with Penalty	162
37	Illustration of Pattern Search Optimization Progress without Penalty Factor	163

ACRONYMS

ASC	Adaptive Signal Control
AHTD	Arkansas State Highway and Transportation Department
ALINEA	Asservissement LINEaired' Entrée Autoroutière
ALPS	Augmented Lagrangian Pattern Search
ATCS	Adaptive Traffic Control System
ATIS	Advanced Traveler Information System
ATMS	Automatic Traffic Management System
AWIS	Automated Work Zone Information System
CALTRANS	California Department of Transportation
CCTV	Closed Circuit Television
CDF	Cumulative Distribution Function
CHIPS	Computerized Highway Information Processing System
CMS	Changeable Message Signs
CORSIM	Corridor Simulation
CSC	Central System Controller
DI	Disutility Index

DLL	Dynamic Link Library
DMS	Dynamic Message Signs
DOT	Department of Transportation
DP	Dynamic Programming
DS	Degree of Saturation
DTA	Dynamic Traffic Assignment
FHWA	Federal Highway Administration
GA	Genetic Algorithm
GP	General Problem
GPS	Generalized Pattern search
HAR	Highway Advisory Radio
ICM	The integrated corridor management
IRP	Interstate Rehabilitation Program
MADS	Mesh ADaptive Search
MCU	Mobile Control Unit
MOE	Measures of Effectiveness
NHS	National Highway System
OPAC	Optimized Policies for Adaptive Control
PCMS	Portable Changeable Message System
PDF	Probability Density Function

PI	Performance Index
PS	Pattern Search
QP	Quadratic Programming
RHODES	Real-time Hierarchical Optimized Distributed and Effective System
RITA	Research and Innovative Technology Administration
RRS	Roadside Remote Stations
RTCMSC	Real-Time CMS Control
RTE	Run Time Extension
SA	Simulated Annealing
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split, Cycle and Offset Optimization Technique
SCOOT	Split Cycle Offset Optimization Technique
SPSA	Simultaneous Perturbation Stochastic Approximation
SQP	Sequential Quadratic Programming
TIPS	Travel Time Prediction System
TRANSYT-7F	TRAffic Network StudY Tool, version 7F
TRB	Transportation Research Board
TxDOT	Texas Department of Transportation
USDOT	U.S. Department of Transportation
VK	Car equivalent flow (SCATS)

VMS	Variable Message Signs
WZSAS	Work Zone Speed Advisory System

CHAPTER I

INTRODUCTION

Problem Statement

Traffic delays caused by non-recurrent events such as work zones, accidents, weather, and other causes are a serious problem worldwide. Work zone delays account for nearly 24 percent of non-recurring delays and roughly contributes to 50 percent of all non-recurring highway congestion with traffic incidents, weather and other special events (1, 2). A study by Ullman in 2004 showed that drivers faced one active work zone every 100 miles, and one lane closure every 200 miles on the National Highway System (NHS) in 2001 (3). In May 2006, the U.S. Department of Transportation (USDOT) estimated that the total business loss due to these delays was approximately \$200 billion, nearly 4 billion hours of wasted time en route and more than 2 billion gallons of fuel wasted annually because of freight bottlenecks and traffic jams, to which the work zones have contributed significantly (4).

Actions have already been taken to manage work zone delays that are a major component of recurrent delay. For example, nighttime construction has increased to lessen work zone delays. In 2002, 22 percent of work zones were designated for work at night, and 18 percent were active work zones all-day or nearly all-day (18 hours or more) (1).

Effective traffic and diversion guidance is an alternative delay alleviation measure for drivers.

With the help of ATIS and ATMS, traffic management plans are also designed to divert traffic onto parallel arterial roads to reduce traffic delays. However, the alternative road traffic information is not as adequate as that on the freeway; hence, it often results in drivers' hesitation and reluctance in their diversion choice.

Many state DOTs disseminate accurate work zone traffic delay estimation as public information, such as 511 Travel Information System, real-time traffic information on website, and Highway Advisory Radio (HAR) to reduce delays. With the help from public information, drivers can make better decisions on departure time shifts, mode change, or even cancellation of a trip. Therefore, these information would benefit work zone mobility and possibly improve safety as well.

Dynamic Message Signs (DMS), variable speed limits, dynamic early/late merge lanes, variable message signs (VMS), and other ITS technologies have also been used to improve work zone traffic flow. On the other hand, in past decades, arterial road intersection signal control and timing plan optimization techniques have been studied extensively, and significant advancements have been made on these fronts. As a result, reductions in vehicle stops, delays, congestion level, and improved efficiency on arterial roads have been achieved. Consequently, safety, air quality, and energy conservation have also been improved by these techniques.

In May 2006, the USDOT funded a 5-year initiative to support the practice in the Integrated Corridor Management (ICM). The ICM optimizes the use of the existing infrastructure and aims to use full capacity on the metropolitan corridors through an integrated transportation management. As a result, the operational and technical integration of the freeway and arterial network among the initiative objectives

will benefit the freeway corridor users by diverting an optimal traffic volume onto the arterial road (4). The proposed DMS control system is an application of the ICM concept that only involves highway management and does not include other surface transportation modes such as the railway. However, this proposed system will also contribute to the ICM even though this study is confined to the freeway corridor.

However, improvements of the ATIS and ATMS will benefit drivers more. For example, the diversion information can coordinate with alternative road signal plans to maximally utilize corridor capacity and reduce corridor delay.

In an ATIS, information such as travel time, travel distance and delay for both mainline and alternative roads is provided by Dynamic Message Signs (DMS). In order to meet the challenge of diverting freeway traffic to an alternative route at the systems optimum level, a DMS control system is proposed in this dissertation, which is a combination of ATIS and ATMS. It possesses features of a control system because it provides not only general travel information like travel time, distance to the end of work zone but also an adaptive road choice guidance message that encourages and monitors diversions.

Objectives and Approaches

Objectives

This dissertation aims at utilizing the whole corridor capacity (capacity from both mainline and alternative roads) to reduce corridor delay. As stated in Chapter Two, adaptive DMS message display is a practical and efficient approach to achieve this goal. The purpose of DMS diversion control is to find an

ideal diversion volume that can be accommodated by newly optimized signal timing plans on alternative roads. The total corridor delay time will be reduced after model suggested diversion volume is realized. The eventual diversion volume is achieved by selecting a DMS message (belonging to certain level with regarding to expected diversion value) to lessen the total delays on both the mainline and alternative road.

In order to reduce corridor delay in the above-mentioned method, a delay optimization model is necessary and it should be solved by efficient algorithms. A traffic simulation system sponsored by FHWA, CORSIM, can provide simulated traffic data and work as a platform to evaluate the proposed model. On the other hand, CORSIM provides RTE to communicate with external programs such as the proposed algorithms implemented in MATLAB (for example, CORSIM alternative road signal cycle length, and splits updating from MATLAB during the simulation).

The system architecture of the proposed approach is shown in Figure 1. There are three text box groups at the left, middle and right. The center group of four boxes is scope of the proposed work in this dissertation. They provide an optimization model and solution. The roles of the left four boxes and the right three boxes are to provide data and evaluation.

Generally, the proposed system has two functional parts: model and its solution (center part of Figure 1), and data source and system evaluation platform (left or right part of Figure 1). For data source and system evaluation, two options are available: traffic simulation in the right three boxes, and real-world implementation in the left four boxes. CORSIM simulation is chosen in this dissertation to provide data and evaluation platform for reasons of time and cost.

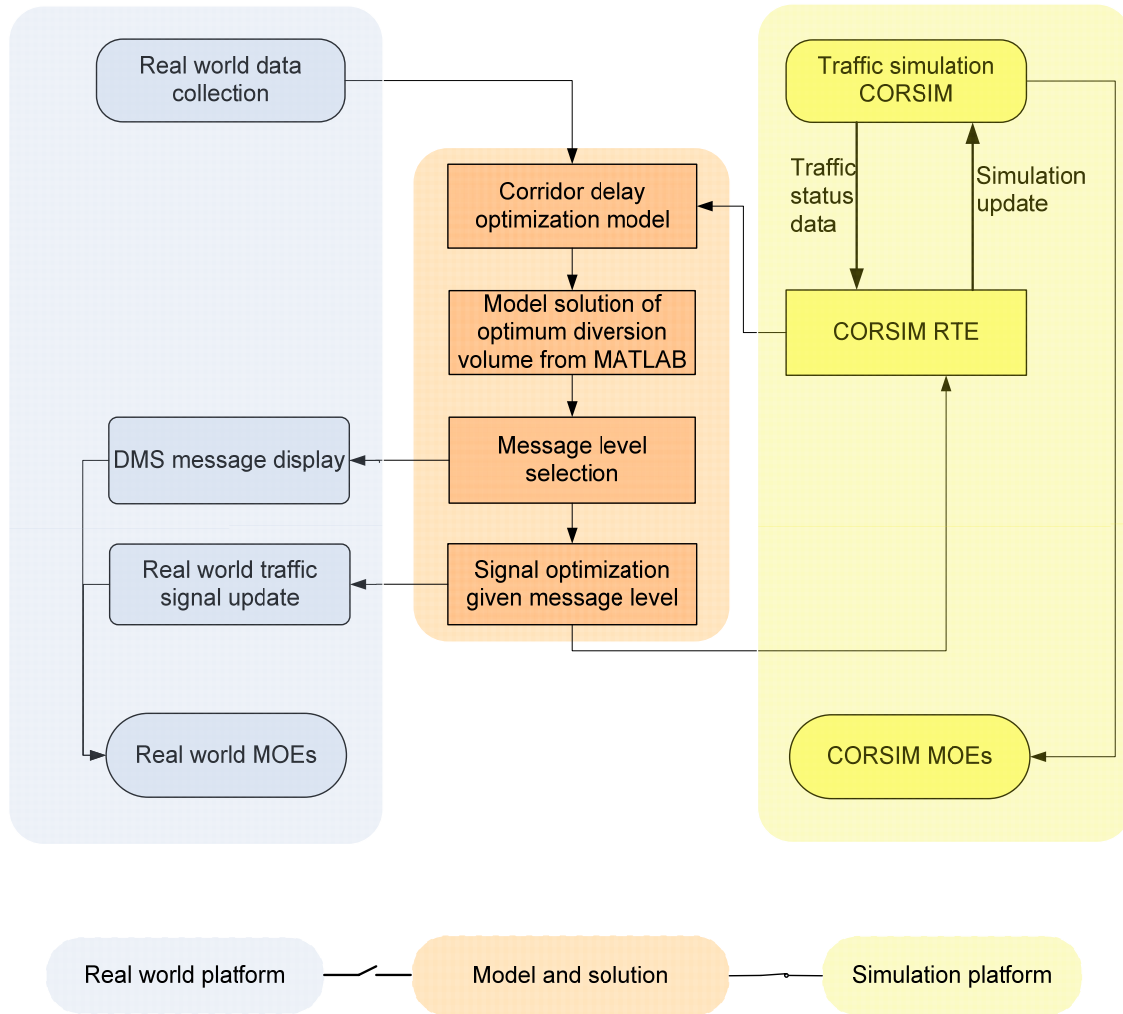


Figure 1 General System Modules

The effectiveness of the proposed system is verified by comparing results with other traffic signal optimization software, such as TRANSYT-7F. The example network running under TRANSYT-7F optimized signal plan serves as the benchmark case. The network performance of TRANSYT-7F optimization and the proposed system is evaluated by HCM 2000 (126) MOEs such as travel delay and average speed. The travel delays discussed in this dissertation refer to freeway traffic delays, alternative

road signal control delays and the travel time difference between mainline and alternative roads. The average speed is for the whole corridor area rather than the average speed from mainline and alternative road only.

Approaches

A corridor delay optimization model considering three types of delays (mainline, alternative road control, and normal travel time difference between mainline and alternative road) is presented in this dissertation. Ideally, system input data (network traffic state) should be collected from freeway and alternative road by traffic detectors in the real world. However, CORSIM simulation is used as a proof-of-concept study and evaluation platform in this study due to time and cost reason. CORSIM provides Run Time Extension to let users program and communicate with other software (for example, MATLAB). In order to solve the model, three algorithms from MATLAB optimization toolbox are selected as candidate algorithms.

The effectiveness of the proposed system has been validated at the end of this dissertation by comparing the study results with the existing popular signal optimization software package TRANSYT-7F over the CORSIM platform. CORSIM MOE outputs, total traffic delays and average network speed between the before and after DMS control cases are extensively compared under different traffic conditions.

Because the DMS messages could vary case by case, DMS message levels instead of detailed messages are defined with consideration of the driver's compliance. Because of the uncertainty of the

driver's compliance with a message, the driver's compliance with DMS guidance is described by a distribution for each DMS message level. For example, an average of 30 percent of drivers is expected to choose diversion from a level four DMS message given an extreme value distribution. The eventual diversion volume might be different from the expected volume even when the same DMS message is displayed repeatedly. Therefore, the proposed study requests signal optimization to deal with these diversion volume variances.

In general, the proposed system is an integrated real-time diversion control system, which simultaneously deals with diversion volume optimization, DMS message level selection, and alternative road signal coordination and optimization. Message level is selected once the expected diversion volume is found. The message is then picked from the predefined message pool for the expected diversion rate.

Significance of the Study

The literature review has found that DMS messages and information in the current Advanced Traveler Information Systems (ATIS) are mainly about mainline information and the alternative road traffic information is often provided inadequately or partially. Studies show that motorists' diversion decisions are indeed affected by the alternative road information (6, 7, and 8). As a result, providing travelers reliable and accurate information of both mainline and alternative roads will help delay alleviation.

In addition, by making diversion control coordinate with the alternative road traffic signal as well as optimizing the signal to accommodate the diversion volume for better corridor capacity utilization, the

proposed system can reduce the total corridor traffic delays and increase the corridor travel speed, throughput and safety.

The proposed freeway diversion system focuses on diverting traffic onto alternative road by the DMS message. This system not only provides real-time information but also make the message adaptive to driver compliance. Each message displayed on DMS is associated with an expected diversion rate following the predefined message levels and messages.

The diversion control is achieved by selecting the suitable message level to guide certain diversion volume that can be accommodated by the optimized alternative road signal, which will result in less delay for the whole corridor.

Scope and Limits of the Study

Obviously, it is unnecessary to divert traffic when a mainline road is not congested. On the other hand, it is unnecessary to divert traffic onto an alternative road when severe congestion has already appeared on both the mainline and alternative roads. The proposed system will work best under near saturated or over-saturated mainline traffic conditions while the alternative road still has available spare capacity to accommodate a certain portion of the mainline traffic.

The proposed DMS diversion control system is not a regulatory traffic device such as a traffic signal because the DMS only provides messages to guide drivers and cannot guarantee the expected diversion. Nevertheless, in the proposed system DMS does not have the same role as in the existing ATIS.

It does intend to manage traffic by providing a specifically selected message with an expectation that drivers are smart to choose a shorter travel time and thus to contribute to a specific diversion volume.

Because the traffic networks in the real world are too complicated for a model to include all aspects, research scope of this dissertation is defined and some assumptions are made to simplify the studied network. The network used in this study is a freeway corridor that has one freeway and one alternative road. Intersections on the alternative road are equipped with pre-time signal controllers.

Assumptions made for this dissertation are listed as follows:

The number of DMS boards and locations, and number of detectors are fixed.

The driver compliance distribution for each DMS message level is assumed to follow a particular distribution, Extreme Value distribution.

Alternative road signal system is pre-timed. If it is an adaptive signal control system optimizing the alternative road signal is not necessary since the adaptive signal controller should adjust the signal for unforeseen traffic conditions theoretically. If it is an actuated signal control system, different decision variables (such as green time extension, maximum green time, minimal green time and so forth) and control delay calculation need to be re-modeled.

For the mainline sections, only unidirectional capacity decrease is considered. Traffic in the opposite direction is assumed not to affect the capacity-decreased side.

HCM2000 intersection control delay estimation is adopted for alternative road control delay estimation.

There is no initial queue for intersection lane groups at the start of the analysis period.

CHAPTER II

LITERATURE REVIEW

The proposed diversion control system is based on the existing real-time traffic information system, signal optimization and driver compliance with DMS message. In this chapter, the existing real-time Advanced Traffic Information System (ATIS) and Automated Work Zone Information System (AWIS), signal optimization algorithms, driver compliance with diversion message, adaptive signal control system and integrated corridor control system are reviewed for the model construction and algorithms references. The literature review is summarized at end of this chapter.

Real-Time Traffic Information System and Benefits

Freeway traffic congestion is mainly caused by oversaturated traffic volume, work zones, accidents and weather. Research has been conducted in this area to relieve traffic delays. One strategy is to provide traffic information to the public and guide travelers to route choice and or shift route. Many Advanced Traveler Information Systems (ATIS) and Automated Work Zone Information System (AWIS) have been deployed and evaluated before. Studies of existing systems will benefit the proposed system in real-time information processing, transmission, messages displayed to the public, driver compliance with DMS messages and so forth.

ATIS and AWIS

There are different types of ATIS systems classified based on the purpose and use. ATIS systems for particular purposes such as speed control and lane merge control are beyond scope of this dissertation and are excluded from the literature review. ATIS studies with field evaluation are reviewed in detail and are described in the following sections.

Automated Work Zone Information System (AWIS) is a type of ATIS system and is specifically used for work zone traffic information and management. In 2005, Lachhwani and Horowitz (9) compared 16 different systems in 27 deployments throughout the United States and made an extensive evaluation. An evaluation criterion for portable ATIS in work Zones was set up based on existing experiences. At the same time, an eleven-step procedure was recommended for traffic engineers to choose and configure AWIS. Five systems from Lachhwani and Horowitz's study were reviewed in details and listed in the following sections.

ADAPTIR

ADAPTIR was developed by Scientex Corporation (124). It consists of Highway Advisory Radio (HAR), Portable Changeable Message System (PCMS), Central System Controller (CSC), and Roadside Remote Stations (RRS). Radar sensors keep measuring speeds at multiple locations upstream of the work zone; and RRS receives data from the radar sensors and passes on to CSC. Then, CSC will calculate the speed difference and delay. A message to be displayed on the PCMS and broadcast by HAR is chosen and updated based on the CSC calculated results.

One or more PCMSs are placed upstream of the work zone to display real-time traffic information for drivers. The logic of the message display is that CSC calculates the difference between two continuous speeds and chooses messages to display on PCMS. If the calculated difference is smaller than 10 mph, nothing will be displayed on the PCMS. If the difference is greater than 10mph, the delay will be calculated. If the estimated delay is less than 5 minutes, only the reduced speed message will be displayed. If the estimated delay is between 5 to 30 minutes, delay information will be displayed. The diversion advice will be displayed when the delay is greater than 30 minutes (10). Nevertheless, the diversion advice is given without alternative road information available; the drivers will take the alternative road at their own risk of more delays there.

ADAPTIR was deployed in a Nebraska work zone on I-80 between Lincoln and Omaha in 1999 to provide warnings to drivers of slower speeds and delays within the work zone and encourage them to take diversion route if possible. Evaluation from deployment of Nebraska showed that the system had mixed results for vehicle diversion and the locations of PCMS, which influenced the effectiveness of the speed advisory messages. In an Arkansas deployment, they found that a 600-second PCMS message update interval is too long. The estimated delay error is greater than the acceptable criteria, 5 minutes, for all the cases (11).

IntelliZone

IntelliZone is developed by Quixote Transportation Technologies Inc. (125). It is an advanced ITS platform that allows engineers to manage and display real-time messages to road users when congestion

appears. IntelliZone consists of microwave sensors, a Mobile Control Unit (MCU), and Dynamic Message Signs (12). All the units are linked by wireless communication.

The detectors record speed, volume and occupancy of each lane at a 30-second interval. Then, the system estimates the volume-weighted average speed over all lanes each three minutes. The average speed is displayed in 10 mph ranges (for example 20-30 mph) on the DMS. The speed messages alternate with message “Actual Speeds Ahead” on the message board. There will be no messages displayed on the DMS if the average speed is greater than 50 mph. A message “Stopped Traffic” will be displayed on the message board to warn drivers about the low speed or stopped traffic if the average speed is less than 20 mph.

This system has been deployed in Missouri and Wisconsin for field evaluation. Evaluation from Missouri showed that 66.3 percent drivers slowed down following the message on the DMS. However, 16.9 percent of drivers would not follow the advice from the message. On the other hand, deployment in Wisconsin did not show good diversion influences. Only 3 out of 186 drivers diverted because of the DMS message (13).

TIPS

Travel Time Prediction System (TIPS) is a portable, automated, real-time and smart work zone information system for predicting and displaying travel time or delay information to drivers. This system collects real-time traffic flow data at 30-second intervals by several nonintrusive sensors such as Radar and so forth. It transmits data by wireless communication to an on-site personal computer. TIPS was specially

developed for freeway work zone applications, however, it is suitable for freeway recurring congestion that is similar to work zone.

TIPS system has been deployed in Ohio and Wisconsin for evaluation. In Ohio, The messages are displayed in two phases; one phase is for travel time to the end of work zone, and the other phase is the distance to the end of work zone. For a six-second message update cycle, four seconds were allocated to travel time to the end of work zone and 1.5 seconds is used for travel distance to the end of work zone. Analysis of the collected field data indicates that 88 percent of the travel time information is within ± 4 minutes of the predicted time. TIPS travel time was concluded to be accurate. In Wisconsin, float cars are used to verify the TIPS time prediction. It is found that more than 70 percent of predicted times are within 5 minutes of floating car travel times. However, the original TIPS sign does not provide guidance for alternative route selection and the later improved TIPS can provide alternative road information. For example, in Milwaukee, WI, TIPS provides alternative road information. Pant who conducted the OHIO TIPS test recommended the Ohio Department of Transportation to implement TIPS in suburban or rural freeway after evaluating the system (14, 15, 16, and 17).

Real-Time CMS Control

Real-Time Changeable Message Signs Control (RTCMSC) system consists of a video detection system, and a portable CMS. The objective of this system is to provide drivers work zone information and encourage alternative road usage when congestion arises on a mainline route.

The RTCMSC diversion advice logic is based on speed detection. Video detection system detects and compares speeds of three consecutive freeway vehicles. If their speeds are all below 20 mph, then the diversion advice message will be shown on the CMS. Otherwise, general work zone information messages such as distance to the end of work zone and driving advisories will be displayed.

RTCMSC has been deployed in Nebraska and evaluated based on its effectiveness in diverting traffic to the alternative road. Analysis shows that the RTCMSC performed in diverting drivers to an alternative route under work zone congestion situation. However, only about 4.5 percent of drivers diverted onto the alternative road when the diversion message was displayed on CMS. Traffic volumes were not statistically changed during the study (18).

CHIPS

The Computerized Highway Information Processing System (CHIPS) is developed by ASTI Transportation Systems. It has been deployed in Arkansas and California for evaluation (19). In the California deployment, there are a set of predefined messages in CHIPS. Three RTMS Radars are used for data collection and three to five CMSs are set up for information display. Scenarios are designed based on combinations of RTMS Radar queue detection. Generally, all messages are displayed on various CMS under different scenarios.

During the two days for evaluation, significant diversion volume is observed. The travel time on the alternative road is measured to be shorter than that of mainline when the mainline faces traffic congestion. This makes the diversion decision more reasonable (20, 21). In Arkansas deployment, a direct

correlation is found between the CMS message change, and highway traffic volume change. The number of vehicles taking the alternative route increases by a factor of two for all vehicles and a factor of nine for truck traffic (22).

Other ATIS system

The California Department of Transportation (Caltrans, 23) deployed an automated Work Zone Information System (AWIS) in October 2004 to mitigate traffic disruptions. The AWIS updates DMS every 10 minutes with one of the six predefined messages according to the detected work zone congestion.

The six predefined messages are classified into three message levels according to the estimated travel time over 2.8 miles of freeway segment. When the estimated travel time exceeds 50 minutes, the alternative road traffic condition will be checked. If the alternative road traffic condition is free flow speed, the estimated travel time and detour advice will be displayed on the VMS board. When the estimated travel time is below 50 minutes or even below 25 minutes, only the estimated travel time will be displayed.

The estimated travel time is calculated based on the average travel speed by multiplying it with the adjust factor that is related to different speed range. The adjust factor is 1 for speed higher than 45 mph, 1.1 for speed between 30mph to 45 mph, 1.25 for speed between 15 to 30 mph, and 1.4 for speed below 15 mph.

In 2002, a study of the Work Zone Speed Advisory System (WZSAS) is conducted by Geza (24) in Nebraska. It was found that 71 percent out of 1,203 online survey respondents would take the alternative road. Delay time message was desired by 71 percent of the respondents against 51 percent on traffic speed.

The WZSAS is an en-route traveler information system that provides real-time speed advisory information by Changeable Message Signs (CMS) located before the diversion point at upstream of work zone. The WZSAS encourages drivers to divert onto alternative road when work zones are congested. The logic of the CMS message selection is straightforward. The system takes the speed and volume data from detectors before the CMS sign. Then, the average speed of that point is computed and a CMS message is then picked based on the computation result. No message will be displayed if the speed is greater than 55mph; delay warning message will be displayed (without detailed delay time) if the speed is less than 15 mph. Otherwise, the average speed will be displayed.

Effectiveness and Benefits of ATIS

The existing ATIS and AWIS systems have been proved effective from many studies. An AWIS system of the California Department of Transportation (Caltrans) in October 2004 reduced average weekday traffic volume in the work zone by 16 to 19 percent. The AWIS operation contributed to a maximum of 50 percent overall decreases in delay during weekday peak hours (3).

From 2000 to 2005, The Arkansas State Highway and Transportation Department (AHTD, 22) carried out the Interstate Rehabilitation Program (IRP), ADAPTIR, and CHIPS were installed on interstate rehabilitation projects in Arkansas. The effectiveness evaluation for these two systems is also conducted by comparing the estimated travel time with actual travel time. From the evaluation, it is found that 14 travel times out of the 144 trial runs exceed the acceptable error of 5 minutes. The system is 89.71 percent accurate from the travel time comparison.

Guo and Srinivasan (25) investigated the commuters' daily urban route departure time switching given real-time information. It is found that "trip time variability and lateness propensity" increased under high congestion. What is more, it is also found that departure time shifting influences daily variance more than route shifting. It concludes that transportation control measures such as staggered work hours, telecommuting, compressed workweek, and supply of real-time information can improve trip time and network reliability by the collected data.

Diakaki et al. (27) developed an Integrated Traffic-Responsive Urban Control system that involves signal, ramp metering, and VMS control (a detailed description of this system is in the integrated corridor control section). The evaluation from preliminary simulation and field implementation is positive for both particular sites and corridors in general. The system is modeled as a linear-quadratic control problem based on a store-and-forward type of modeling. Simple feedback from ramp metering and VMS is considered in the model. From 345 received questionnaires, it is also found that 40 percent of motorists changed routes according to VMS messages. The travel time decreases if compared with the non-diversion situation. Meanwhile, the alternative road throughput has 23 percent increases revealed by the detector collected volume data. The system increases the mainline throughput by 6 percent during the congested hours. At the same time, network travel time is reduced by 10 percent. Nevertheless, this system does not have real-time modification of alternative road traffic signal cycles and offsets, mainline travel time estimation revealed to drivers, coordination among ramp metering, signals, and VMS. In 7 out of 27 intersections on the alternative road, they are equipped with adaptive signals that can respond to the diversion volume and

leaves the rest 20 intersections unchanged. Therefore, it still has some space for further system performance improvement.

In April and May of 2004, the Texas DOT (TxDOT, 28) conducted a survey to find driver desired traffic information in Houston area. The results indicate that real-time travel time information posted on VMS does infer drivers' route choice. About 85 percent of respondents indicate that their route changes are based on the VMS message. What is more, respondents also show high interests to see the travel time information on the VMS board.

North Carolina also conducted some work zone related ITS studies. Results from these studies of intelligent transportation systems in North Carolina work zones (29) indicates that the work zone delay messages reduce maximum queue length from 2.94 miles to 1.3 miles (56 percent decrease) and time delay from 14.8 minutes to 6.6 minutes (a 55 percent decrease).

Vasudevan et al. (30) evaluated performances of commuters' on-time arrivals when route-specific travel time information and general radio advisory information are provided. It was found that drivers who use route-specific travel time information instead of wide-area traffic radio advisories could improve on-time rate by 5 percent and 13 percent during afternoon and morning peak hours respectively in Washington DC. The drivers who listen to radio advisory information have the advantage of making en-route road choices. However, because of the vague, imprecise and incomplete route coverage, they still have a lower on-time rate than the users having route-specific travel time information.

In 1998, U.S. traffic engineers toured Europe to investigate ITS technologies deployment and operation. From the tour, they found that overall accident rates were decreased as much as 23 percent by

detecting congestion, initiating lane controls, and warning drivers of slow traffic. In Amsterdam (31), freeway management systems reduce overall accident rate by 23 percent, serious accident rate by 35 percent, and secondary accident rate by 46 percent. In Germany, variable speed limit signs and lane control signals reduce accidents by 20 percent.

Logi et al. (32) investigated the use of a real-time, knowledge-based decision support tool, “Traffic Control Manager” on alternative route selection for nonrecurring congestion by Dynasmart simulation. From the simulation results, it is found that the tool can help reduce average travel time by 2 to 29 percent. During the highest and lowest levels of demand period, improvement is smallest among all the simulation scenarios. It is also found that it becomes harder to find measures to balance the demand and capacity when the network has less spare capacity available or traffic demand is close to saturation.

Ran et al. (33) conducted a survey in December 2001 to evaluate the impacts of a traveler information system. Two hundred and twenty one (number spelled) out of the 428 mailed out questionnaires were returned and analyzed. The data analysis shows that approximately 68 percent, and 72 percent of respondents adjusted their travel routes based on the information provided by VMS during winter months (December – March) and non-winter months (April – November), respectively. Twelve percent of respondents adjusted their travel routes more than five times a month during the winter months, while it is higher in non-winter months with a diversion rate of 18 percent.

Dumke et al. (34) conducted a work zone surveillance system study at “Big I” interchange in Albuquerque, New Mexico. The surveillance system mainly consisted of Closed Circuit Television cameras (CCTV), Dynamic Message Signs (DMS), portable changeable message signs (CMS) and the

“BIG I” Internet web site. Evaluation after approximately one year of operations showed that the average clearance time on the “BIG I” is 20 minutes faster than the historical average clearance time, 45 minutes, which is a 44 percent reduction. In addition, the average response time is less than eight minutes after system implementation.

All the aforementioned studies show benefits from ATIS and encourage further studies to improve its effectiveness and benefits. One-step further based on the ATIS is to help make diversion decisions for drivers over the system wide data and information. With this capability, the ATIS system changes to a preliminary diversion control system.

DMS Message Content and Driver Compliance

Many studies have been conducted on VMS and DMS message contents and driver compliance with the VMS and DMS messages. Survey and field data collection for before and after cases are the main methods for this type of research. Studies from Peeta (6), Benson (7), and Polydoropoulou et al. (8) show that quantitative, accurate and reliable information about alternative routes played a critical role in encouraging alternative route selection.

Peeta (6) conducted a survey in Indiana for identifying key factors that influence driver’s route diversion decisions given VMS messages. Data from 248 respondents was analyzed and results show that 53 percent of the respondents will divert if more than 10 minutes delay happens on a mainline route and 70 percent of them will divert under adverse weather. The messages content on VMS is classified into eight categories in the survey. The willingness to divert ranges 12.5 percent for general descriptive message

category to 71.4 percent for the accurate information with a clear detour strategy category. However, 20.2 percent will not take the alternative road if only the location of the accident is provided. Rather than using specified delay time for diversion decisions, Mahmassani et al. (35) employs a simulation model to investigate congested urban traffic under real-time information and indicates that diversion will happen when the time saving is 20 percent longer than the remaining trip time.

Benson's survey (7) through 517 computer-assisted telephone interviews in 1994 found that 60 percent of the respondents would take an alternative road suggested by VMS under heavy congestion. On the other hand, the reasons for not taking the alternative road were also surveyed. Three main reasons were identified as the fear of getting lost from the diversion, doubt about the alternative road conditions and motorists choosing their own alternative road.

Polydoropoulou (8) conducted a survey in 1993 for en-route response to unexpected traffic congestion and found that a diversion rate ranging from 27 percent to 61 percent for the five ATIS information Scenarios. The most significant increase occurred when quantitative real-time or predictive information on both mainline and alternative roads are available. Travel information on alternative routes was asserted to be critical for diversion decision from this study.

Field deployments and evaluations for different systems at multiple locations show different diversion rates. Wisconsin TIPS's deployment and evaluation shows a maximum of 10 percent diversion exclusively from TIPS during peak hours (9, 14). However, only 3 out of 186 drivers reported that they diverted because of the IntelliZone signs in Wisconsin (9). Brown Real-Time system results in only 4.5 percent diversion, nevertheless, its effectiveness still gets statistically significant with 95 percent

confidence (18). Evaluation of the Travel Messenger system in North Carolina shows that 40 percent of drivers are often influenced while 46 percent are sometimes influenced by the VMS (9). Similar to North Carolina's diversion rate, 40 percent of drivers were found changing route because of VMS recommendations in Glasgow, Scotland (27).

French DOT (36) estimated from the year 1992 traveler survey results (8000 mailings and 100 telephone follow-ups) that 50 percent of drivers will divert to free flow alternative roads given a congested mainline condition. However, only 3 to 5 percent of drivers will divert to the less congested road if comparative information for both mainline and alternative road is given. Further study for queue length and diversion probability reveals that diversion probabilities are 0.01, 0.07, 0.11, 0.17, and 0.31 when queue length range is [0, 0.5), [0.5, 1), [1, 2), [2, 3], and [3, 4] KM, respectively.

As contrasted with diversion rate of 16.3 percent in San Francisco, Seattle has a diversion rate of 5.8 percent in home-to-work trips and 13.7 percent in work-to-home trips. At the same time, study in Chicago shows that 42.5 percent of surveyed drivers divert from their normal routes to avoid traffic delays because of quantitative travel time information from television and radios (24, 37, and 38). A study in London by Chatterjee et al. (39) indicates 24 percent diversion when drivers face delays. At the same time, study from Wisconsin shows diversion rate between 7 percent and 10 percent depending on the location and the day-of-week (40).

Driver types such as occasional drivers or frequent drivers for a segment of road also have a different effect on diversion decision. Zwahlen (41) finds from a survey of 660 respondents that frequent drivers have a 20 percent diversion rate, higher than occasional drivers' diversion rate of 13 percent.

Many reasons such as insufficient alternative road information and fears of getting lost on the alternative road make drivers reluctant to choose the alternative road. Nevertheless, information displayed on the VMS plays a critical role for diversion. Different diversion rates from many studies are not comparable without the detailed message contents. However, driver's compliance data will still help this dissertation on the driver compliance distribution and DMS message (levels) design.

Signal Control and Optimization

Current signal control systems can be classified into four categories: isolated intersection, arterial network, area wide surface street network, and integrated corridor signal control. Corresponding to this, the signal optimizations can also be divided into four groups: isolated intersection, surface street off-line and on-line optimization, and integrated corridor optimization (59).

The usage of conventional model solutions like the Dynamic Programming and Conjugate gradient algorithms are described in this section. Introduction of adaptive signal control, other signal optimization methods, and introduction of good off-line signal optimization software TRANSYT-7F is also included in this section.

Conventional Models

Conventional optimization methods are the methods that are solved by mathematic computations, for example, integer programming, hill climbing, Dynamic Programming (DP), or descent gradient searching, Quasi-Newton searching and so on. Nonconventional optimization methods of signal

optimization mainly refer to methods that use Genetic Algorithm (GA), Simulated Annealing (SA) and other similar algorithms.

DP is a good candidate for signal optimization because signal control can be treated as a multistage decision-making process and DP is efficient in dealing with multistage control problem. As early as 1963, Miller first proposed a DP strategy to determine whether to extend a phase at a short fixed time intervals by optimizing the control delay (42, 43). Zijverden (44) and Bang (45, 46) followed his direction and made similar approaches later. Continuous-time DP model and discrete-time DP model for signal optimization were developed by Grafton (47) and Robertson (48), and have made the DP usable in more situations.

Signal Optimization

In recent years, heuristic algorithm such as GA and SA are more popular for signal optimization because of their possible global optimization properties and low requirement of objective function and constraints. Most of those GA and SA studies are for off-line optimization since their execution time cannot satisfy the on-line implementation requirements, especially for a large network.

Lee et al. (55) proposed a genetic algorithm for adaptive signal control. The system includes three components: genetic algorithm optimization module, internal traffic simulation module, and a database management module. Decision variables are the phase number, sequence, and lengths. It is working in a rolling horizon manner (60 seconds signal duration and 10 seconds roll period). Only the first 10 seconds of each cycle is actually implemented because the newest optimal timing plan is generalized every 10 seconds.

The proposed model is solved with rank-based selection combined with an elitism method by binary coding.

Park et al. (56) evaluated three stochastic coordinated actuated traffic signal control optimization methods: Genetic Algorithm, Simulated Annealing, and OptQuest engine and proposed a model for signal optimization. The results when compared with SYNCHRO, TRANSYT-7F, and PASSER indicate that vehicle queue time and control delay from the proposed method outperformed existing programs in optimizing the four basic signal parameters (cycle, green split, offset, and phase sequence). The optimization variables are classified into three groups. Group 1 includes cycle length, phasing sequences, green splits, and offsets. Group 2 includes minimum green, recall, vehicle extensions, detector placements, delayed call, and extended call. Group 3 includes minimum initial, maximum initial, time actuation, time before reduction, time to reduce, and minimum gap. Objective functions related to variable groups are Queue Time (vehicle-minutes) for group 1, and Control Delay (seconds per vehicle) for group 2 and 3.

In addition, other heuristic algorithms have also been developed by some researchers for signal optimizations. Luo et al. (54) explores the Robust Method for signal optimization for pre-timed isolated intersections and found that it can adapt to the variability of traffic flow by using mean and variance of traffic volume as input variables. The Robust Method can outperform TRANSYT-7F and other conventional models using Webster's delay equation for moderate and congested traffic conditions when the arriving flow rates are not stable overtime.

Since arterial intersection control delays are only part of the proposed objective function of this dissertation, more literatures about special intersection signal optimization are not included in this section.

The aforementioned studies showed popular methods, algorithms, and present references of algorithms selection for the proposed diversion control system.

Adaptive Signal Control

Adaptive control is a real-time, demand responsive traffic signal control. It optimizes signal operations based on real-time traffic data from detectors. Adaptive Traffic Control System (ATCS) is a third generation signal system and is the most advanced signal control system so far. It simultaneously optimizes signal splits, offset and so forth based on the detected volumes and results in reduction in delay.

There are many ATCSs used in the real world. Some Adaptive Signal Control (ASC) systems were developed as early as in 1980's (99). The Split, Cycle and Offset Optimization Technique (SCOOT, 100) was widely utilized since its debut in 1981. Sydney Coordinated Adaptive Traffic System (SCATS, 102), Optimized Policies for Adaptive Control (OPAC, 116), Real-time Hierarchical Optimized Distributed and Effective System (RHODES, 117), ACS-Lite (120), and Los Angeles Adaptive Traffic Control System (LA-ATCS, 103) also have been installed and tested in the real world. Meanwhile, some other systems such as MOTION (104), Microprocessor Optimized Vehicle Actuation (119), Prodyn (105) and UTOPIA (106) were proposed and studied without many installations. Only the ones with installation in the real world and received good evaluations were reviewed in this dissertation.

More than 137 SCOOT systems have been installed all over the world, which makes it the most widely deployed ASC system worldwide (118). SCOOT uses split, cycle time, and offset as decision variables to optimize stops and delays at an intersection. Data from detectors were used to predict queue

lengths at the stop line. After a short-term prediction of traffic condition, it modifies the split and offset to reduce accumulated queue lengths. In order to avoid blocking the upstream intersections, offset of an intersection will be computed and employed at a certain time interval. The initial fixed-time plan is not required by SCOOT and it can start from any initial signal settings (49).

Generally, SCOOT has three optimizers (cycle time, green splits, and offsets) and a Performance Index (PI) from delay, stops, and congestion to be selected as minimization objectives. Traffic volumes on all approaches of the intersections in a network are monitored by SCOOT continuously. Changes to the signal timings for minimizing a Performance Index are made once SCOOT had update values. The PI is a composite measure of delay, queue length and stops in a network. These signal-timing changes are small enough to avoid major perturbation in traffic flow and frequent enough to respond to traffic condition changes in time. The update interval can be as short as one-second if required by users.

The cycle time optimizer can change the cycle length at a few seconds increment for time intervals more than 150 seconds. The cycle optimizer will be fully operated at about 90 percent degree of saturation or 80 percent target saturation at an update time interval of 300 seconds. SCOOT uses downstream loop detectors to optimize left turning phases. The split optimizer will make choice among “signal earlier change”, “5 seconds later”, or “as scheduled before” at each stage. For each cycle on all intersections, the offset optimizer will be called to find suitable offset for each intersection.

Accurate prediction of queue and discharge is critical to SCOOT’s computation. In addition, link journey times, maximum queue clear times and queue discharge rates are important to evaluate the SCOOT performance. Based on Hunt’s evaluation (50), SCOOT outperforms fixed-time signals. Hunt also

concluded that SCOOT is more effective under situations like unpredictable capacity, close to demand and short distance to other intersections.

Sydney Coordinated Adaptive Traffic System (SCATS) was originally developed for utilization in Sydney and other Australian cities. Currently it has been tested in more than 50 cities around the world. Similar to the SCOOT, SCATS also adjusts cycle time, split and offsets under the collected real-time traffic data. Its optimization objective is to minimize overall stops and delay. It minimizes traffic stops under light traffic conditions, traffic delay under heavy traffic conditions, and travel time all the time. However, it is not a model-based optimization. On the contrary, it has a plan library for selection. It is a feedback control system with a hierarchical architecture of two levels, strategic and tactical level (102). At the strategic level, a regional computer is used to coordinate signal timings. At the intersection tactical level, optimization is constrained within the strategic limitation. Early termination of a green phase for less than average volumes and phase skip for no demand will be employed to save green time. The savings in green time can be then reallocated to other phases.

Four operation modes are supported by SCATS. The first (normal) mode is the integrated traffic operations. The second (fallback) mode will implement time-of-day plans when hardware failure occurs. The third (isolated) mode starts a local isolated control depending on vehicle actuation. In the fourth mode, flashing yellow or flashing red on all approaches is displayed as the normal signal.

SCATS uses a degree of saturation (DS) and car equivalent flow (VK) for each approach lane to compute cycle length, split and offset respectively (51). DS is the rate of efficiently used phase time divided by available phase time, and it is calculated by the following equation:

$DS = [\text{phase time} - (\text{unused green time})] / \text{available green time}.$

VK is Car Equivalent Flow, and it is calculated by the following equation,

$VK = DS \times \text{green time} \times \text{vehicles per second at maximum flow}.$

Real-time Hierarchical Optimized Distributed and Effective System (RHODES) has the capabilities to respond to stochastic traffic. It uses turning percentage (or estimated OD matrix), queue discharge rates (saturation flow rate), link free flow speed and so on as input variables. The Horizon roll period (user definable, 45 seconds by default), phase also is used as a decision variable. Queue length, link flow, and predicted delay are used as MOE variables (52, 53).

RHODES has a three-level hierarchy architecture. The highest level is the dynamic network-loading model, which considers the slow-varying traffic characteristics. The middle level is the network flow control model, which allocates green time depending on demand patterns. The lowest level of the hierarchy is the intersection control model that controls the phase changes.

Each level of hierarchy mentioned above has two components: the estimation/prediction and the control part. In addition, there are different algorithms for intersection flow prediction, phase optimization, signal split change, platoon prediction and overall optimization calculations.

Parsons Brinkerhoff Farradyne Inc. and the University of Massachusetts at Lowell developed the Optimized Policies for Adaptive Control (OPAC, 116). The OPAC network is divided into sub-networks for optimization purpose. OPAC treats sub-networks independently in the process.

OPAC switches between two models: one for congested networks and the other for uncongested networks depending on different traffic conditions. In the uncongested model, there are two ways to

determine signal timings: fixed-time plans obtained off-line and a virtual cycle length calculated dynamically. The local signal timings are based on detector data retrieved at 15-second interval and predicted data every 60 seconds. They are implemented for a roll period of two to five seconds. In the congested model, maximal throughput of an intersection is the objective of OPAC. Cycle length computation is completed by a central computer and the rest parameters are distributed to local controllers. Hence, it still can run when communications with the central computer fails.

ACS-Lite is the Lite version of Adaptive Control Software (ACS), which is funded by FHWA. ACS Lite is a low cost solution for real-time traffic conditions. Since 90 percent of the traffic signal systems in the United States are the type of closed loop, ACS-Lite was specifically designed to work for the most popular arterial traffic signal system in U.S. – the closed loop (120).

ACS-Lite cycle time changes on a time-of-day schedule. At each optimization interval (about 10 minutes), the ACS-Lite changes the splits and offsets with a small increments (for example 2-5 seconds) to accommodate traffic flow changes.

LA-ATCS is a traffic signal control system with three operation modes: adaptive, time-of-day and operator control (103). LA-ATCS controls a group of intersections (namely, a section). In the adaptive mode, a common section cycle time is determined from the current traffic conditions. Splits are determined based on traffic volumes at each intersection and offsets are optimized to minimize stops. In the time-of-day mode, it is operated on traffic engineer predefined time plans. The operator control mode is for operator taking over the control and it is useful to deal with special events.

All intersections in one section have a common cycle length computed from the highest traffic level intersection. The highest traffic level intersection varied dynamically by the system as the traffic situation changes. Each intersection can have a particular lower and upper bound of cycle length. Splits are determined based on the traffic flow at each approach of an intersection. Offset optimization of a section will be done once splits are determined. Not all the three system operations are required for all the links in a section.

Software Package TRANSYT-7F

TRANSYT-7F is an acronym for TRAffic Network StudY Tool, version 7F. The TRANSYT model was originally developed by Dr. Dennis I. Robertson of the United Kingdom in 1967.

TRANSYT-7F has acted as a benchmark for signal optimization by some researchers (31).

TRANSYT-7F is a traffic simulation and signal timing optimization program that features genetic algorithm optimization of cycle length, phase sequence, splits, and offsets. TRANSYT-7F combines the optimization process (genetic algorithm, and multiperiod and CORSIM optimization) with simulation model. Optimization options available in TRANSYT-7F are: (1) Hill-Climbing method with macroscopic simulation; (2) GA with macroscopic simulation; (3) GA with microscopic simulation CORSIM-direct optimization; and (4) multi-period optimization and CORSIM-direct optimization (57).

Integrated Corridor Control

In May 2006, the U.S. Department of Transportation (USDOT) announced the National Strategy to Reduce Congestion on American Transportation Network (the Congestion Initiative). The objective of the initiative is to demonstrate how ITS technologies can help control major transportation corridors efficiently. One solution that USDOT is pursuing is the concept of integrated corridor management (ICM). A 5-year initiative to help advance the ICM practice was funded by USDOT in 2006. In September of 2006, eight pioneer sites were selected to deploy, and evaluate ICM strategies by USDOT (4).

However, ICM in the Congestion Initiative is a wide concept and it includes operational, institutional, and technical integration of freeway, arterial, regional rail and bus transit Network. Because this dissertation focuses on freeway corridor study, only the operational and technical integration of freeway and arterial network was reviewed in this research. Therefore, integrated corridor control referred in this research is to control freeway and arterial traffic by multiple strategies and systems such as DMS, ramp meter and so on. For integrated corridor control systems, the DMS is an important component to provide information and to achieve object of traffic control as well.

In 1978, Reiss et al. implemented the earliest work on integrated corridor control in the New York Long Island freeway corridor. The diversion objective was to achieve user-equilibrium network state based on predicted O-D volumes. During a control period, a predefined control plan was selected to match the predicted traffic pattern in each studied section. It was tested for both recurrent and nonrecurrent congestion situations. Significant performance benefits were reported in most cases (107, 108, 109, 110, and 111).

Cremer and Schoof (113) constructed a comprehensive nonlinear optimal control model for the integrated corridor control. There are four traffic control measures concurrently in the model: mainline speed limit, off-ramp diversion, on-ramp metering, and arterial signal control. Total delay time minimization was the objective of the model. The model turns out to be a nonlinear, mixed integer optimization problem, and a two-tier heuristic decomposition is utilized to solve the model. There are three algorithms used to solve the four measures. The diversion decision is a sequence of binary decisions and a special branch-and-bound procedure is employed to solve it. The freeway subsystem is solved by a steepest descent and the green time optimization is proposed with a heuristic random search procedure. It is reported from several cases that it is effective and has achieved the expected efficiency.

Chang et al. (115) presented a model for a typical commuting corridor control in 1993, which consists of ramp metering and assignment of g/C (green time to Cycle length) ratios at signalized street intersections. Traffic diversions are taken as predictable values rather than control variables. The diversion volume is predicted under assumption that travel time and queuing length are known. Network traffic flows are formulated as linear functions of the control variables under the assumption that travel times are constant. The objective of the model is to minimize the total travel time of the network. The nonlinear optimization model is transformed into a linear form under some assumptions and is finally resolved by an existing linear programming algorithm.

An Integrated Traffic-responsive Urban Control system that involves signal control, ramp-metering control, and VMS control was developed by Diakaki et al. (27) in Glasgow, Scotland. A VMS control strategy based on independent feedback control law is applied for equalizing the travel times

between mainline and alternative road. Linear quadratic control and ALINEA local feedback control law are used for arterial signal control and on-ramp metering control respectively. The simulation results as well as the results from the field implementation and evaluations show that the system works for both a specific site and a general corridor.

The control elements used to reduce congestion in the above system are ramp meters, VMSs and signal lights. The on ramp meters are used to regulate the traffic flow entering the freeway and to maintain the occupancy downstream the merging location at a pre-specified level. The variable message signs (VMS) are used to inform drivers of ramp meters' operation status in order to equalize the travel times between two alternative routes, which lead to user optimal conditions. The alternative road signal control is formulated as a linear-quadratic (LQ) problem, whose objective is to balance vehicles within arterial links by varying the split of signal cycles without affecting offsets and cycle duration. All the above controls are based on the current traffic conditions rather than the predicted traffic status.

Zhang (60) formulated a nonlinear integrated traffic control model to utilize the full corridor capacity for accident alleviation. The diversion and signal retiming control measures are solved simultaneously by the gradient projection method. A program is developed to solve the model and the optimization results are compared with TRANSYT-7F and INTEGRATION packages. Generally, the research objective is to find optimal diversion routes, rates and signal timing adjustment for diverting as much traffic off mainline roads as alternative roads can accommodate. Decision variables of the system are green time, off-ramp and on-ramp rates. The objective function is formed as minimization of the summation of the square of the difference between arrival volume and discharging rate from all of the delay

places. A method of feasible directions, and gradient projection, is applied to obtain the optimal solution of the objective function. The system is off-line and is controlled by off-ramp and on-ramp metering. Driver compliance and ramp metering coordination with arterial signals are not included.

Ma et al. (114) conducted a study for coordination between traffic signal timing and ramp metering based on kinematics wave traffic flow model. The integrated control model is solved by an algorithm based on the simultaneous perturbation stochastic approximation (SPSA). The model objective is to minimize total delays in the corridor. The solution from SPSA, Hill-climbing and GA are compared over base cases under two different congestion levels. The SPSA has the highest improvement of level two of 4.8 percent while the GA had 16 percent improvement for level one as the highest one. SPSA does not generate a better solution than GA for level one and a slightly higher (1.6 percent) than GA for level two. Nevertheless, computation iterations of SPSA are far less than GA with a close objective function return, which made SPSA outperform GA on overall performance.

Hamdar et al. (65) developed and explored integrated corridor management strategies based on Dynasmart-P traffic assignment simulation. It is a network wide assignment and simulation given O-D pairs. Among the seven developed strategies, it is found that the best results came from simultaneous implementation of advisory warning VMS, optional detour and signal coordination. The VMS messages used in strategies are classified into four types: speed advisory, congestion warning, optional and mandatory diversion. The driver compliance with the advisory VMS is assumed 50 percent and 100 percent for the advisory warning VMS strategy. For the optional diversion VMS, 47.5 percent drivers chose the diversion. Signal and ramp metering control are the means to forge the strategies in the Dynasmart-P

simulation. The study focuses on evaluation of strategies and simplifies VMS and signal coordination operations.

Jacob and Abdulhai (112) automated an adaptive traffic corridor control using reinforcement learning. It is an adaptive and real-time control from on-ramp, VMS for freeway and arterial corridor under both recurrent and non-recurrent congestion situations. PARAMICS is utilized to train and evaluate the agent off-line. The Q-learning strategy improves the mean speed of corridor by 17.6 percent and decreases the mean travel time by 7.2 percent, which demonstrates the effectiveness of this technique.

Literatures of integrated corridor control show two of the most promising freeway: corridor control tools, ramp metering and dynamic traffic diversion using VMS or DMS. Ramp metering appears to be the most widely studied freeway control strategy and adaptive message change does not receive as many studies as the ramp metering. Researches on integrated corridor diversion control by VMS or DMS do not gain as much attention as ramp metering and arterial network signal optimization. One of the reasons could be the diversion control from VMS or DMS is weaker than that from the ramp metering and signal light; the other reason might be from the high travel speed on freeway and stochastic driver compliance with the VMS or DMS messages under high-speed.

Summary of Literature Review

The Literature review of this study is mainly focused on real-time information system and its benefits, signal optimization, adaptive signal control, diversion control, driver compliance with diversion messages, integrated corridor control and related software package used in this area.

Real-time information systems are popular and have already benefited travelers. However, many ATIS systems only provide information for mainline roads the majority of the time. Some ATIS systems can provide alternative road information through traffic control centers other than the system itself. The characteristic for the ATIS system is to provide information and leave the diversion decision to motorists. Most of the systems are equipped with simple and straightforward time and delay prediction algorithms. The ATIS systems do show great benefits from the real-world deployments and encourage further efforts in this area.

Some researchers study driver compliance with VMS or DMS messages from survey and field data evaluation. Most of the results of driver compliance or attitude towards VMS or DMS messages are provided by diversion rates or percentages. Nevertheless, the relationship between detailed message type, content and driver compliance has not been well explored. A problem of concern is that diversion rates caused by VMS or DMS messages might differ greatly from message to message. If merely a diversion rate is provided without corresponded message content and other related information, it is hard to compare the diversion effects because the message content will affect drivers significantly. This could be a reason that the driver diversion rates vary from one percent to eighty percent from different cases. Because of the difficulties, drivers age, sex, education and income status are not included in most of the diversion rate studies even though these factors do contribute to the drivers' diversion decision which are made randomly.

Signal optimization for isolated intersections, arterial network and adaptive signal control are well studied before. However, coordination or integration with other control methods such as ramp metering, VMS or DMS and so on did not receive enough attentions in the past years.

The former integrated corridor control studies are not all appropriate for on-line and real-time applications, especially for situations that will cooperate with real-time route diversion control. The existing circumstances leave spaces for further studies of on-line cooperated dynamic diversion control.

CHAPTER III

METHODOLOGY

This chapter describes model construction, solution algorithms and initial algorithm evaluations.

In the model construction section, system parameters, different variable types of the model, model constraints and objective function are introduced. In the solution algorithms section, three algorithms employed to solve the model are briefly discussed and evaluated sketchily based on a simple example network. The detailed algorithm evaluations are included in the Chapter Five.

Model Construction

The model structure, parameters, decision variables, objective function and constraints are documented in this section. Driver compliance and speed forecast are other aspects that will be considered later at the model construction stage.

Signal control delay optimization is a critical component of this proposed objective function. Generally speaking, objectives for traffic signal optimization and coordination could include total delay, total number of stops, queue length, throughput, weighted delay, number of stops, total cost of losses, total fuel consumption, total personal delay, and sum of the square of the queue length for maximal throughput and so on. Total delay, total number of stops, and queue length are most frequently used for signal

optimization objectives. Because mitigating traffic delay is the target of this dissertation, the total corridor delay is taken as the objective function to minimize these delays.

System Parameters

Once system parameters are given, a specific traffic network is defined and is ready for optimization study. For the proposed DMS diversion control system, the following parameters need to be provided as system parameters.

G : Collection of lane or lane groups at an intersection i .

A : Set of approaches at an intersection i .

h : Percentage of heavy vehicles in the mainline demand.

DT_j : The j^{th} detector (station) on mainline segment around the study area, it starts to count after the mainline entrance of the studied area.

$DSF_{j,j+1}$: Mainline distance from detector (station) j to $j+1$ (Mile), $j \in [0, m]$, m is the total number of detectors (stations) on the mainline segment.

T : Duration of optimization period (Veh-Hour).

t_{sig} : Rolling period for signal rolling horizon (Second).

t_{DMS} : Rolling period for DMS message rolling horizon (Second). Generally, t_{DMS} should be greater than t_{sig} .

K : Incremental delay factor that is dependent on controller settings. It is constant for fixed time control.

I : Upstream filtering/metering adjustment factor.

X_{Ci} : Critical v/c ratio for the intersection i .

L_i : Lost time per cycle at intersection i (Second).

ρ_{jam} : Jam density, 165 veh/Lane/Mile (vphpl) is adopted in this dissertation, which is obtained from the CORSIM simulation of numerical example network of this dissertation.

c_{iag} : Capacity of lane group g , approach a at intersection i (vph).

S_{iag} : Saturation flow rate for lane group g of approach a at intersection i (vph).

Y : Yellow time.

AR : All red time.

Model Structure

The objective of the proposed model is to minimize total traffic delay of the whole corridor. In the objective function (Equation 1), there are three delay components: mainline delays, alternative road signal control delays, and travel time difference between mainline and alternative roads. The travel time difference could be negative if alternative road travel time is less than mainline roads under normal conditions.

The objective function and its constraints are described in the following sections:

$$\text{Min}(D_{Fa} + D_D + D_I) = \text{Min}(D_{Fa} + D_D + \sum_{i=1}^n D_i) \quad (1)$$

ST.

$$\sum_{p=1}^P (gt_p + Y + AR) = C_i \quad 1 \leq i \leq n$$

The summation of total green time, yellow time and all red

time from all phases at an intersection i should be equal to the cycle length of the intersection i . gt_p represents the green time of a lane group at an intersection and it is a subset of the gt_{iag} .

Y and AR represent yellow time and all red time.

$$gt_{\min} \leq gt_{iag} \leq gt_{\max} \quad \text{Green time boundaries of } i^{\text{th}} \text{ intersection, } 1 \leq i \leq n, n \text{ is the total number of intersection studied.}$$

$$C_{i\min} \leq C_i \leq C_{i\max} \quad \text{Cycle length boundaries of } i^{\text{th}} \text{ intersection. Since all the intersections will share a common cycle length under coordinated operation, only one upper and one lower bound is sufficient.}$$

$$V_{D1} \leq V_{F0} \quad \text{The suggested diversion volume should be less than freeway demand. Strictly speaking,}$$

the diversion volume should also be less than the capacity of off-ramp $V_{D1} \leq C_{off-ramp}$, or

standing queue will spread back onto freeway and may block mainline traffic.

Where the parameters and variables used in the above model are:

n : Number of intersections on and alternative road.

i : The i^{th} intersection on an alternative road.

p : Phase of an intersection.

P : Total Phases of an intersection.

D_{Fa} : Total delay of mainline under diversion control (Veh-Hour).

D_D : Total travel time difference between mainline and alternative road (Veh-Hour).

D_I : Total control delay of all intersections on the alternative road (Veh-Hour).

C_i : Cycle length of the i^{th} intersection (Second). Generally, a common cycle length is used for pretimed signal control.

C_{imin} : Minimal cycle length of i^{th} intersection (Second).

C_{imax} : Maximal cycle length of i^{th} intersection (Second).

gt_{iag} : Green time for lane group g of intersection i (Second).

gt_{min} : Minimal green time of all the intersections (Second).

gt_{max} : Maximal green time of all the intersections (Second).

V_{D1} : Diversion volume from mainline to alternative road (vph)

V_{F0} : Mainline demand (vph), it is forecast based on the detector data from the entrance link.

Mainline Delay

Mainline travel time is calculated by the summation of segment distance divided by average speed in its corresponding segment (a segment's speed is represented by a detector's speed). Mainline travel delays are computed as the travel time difference between the base case and after the DMS control case.

Because of the flow-speed or density-speed relationship, the speed change will show nonlinear relationships with density or flow. This nonlinear relationship is adopted for the mainline speed prediction for the next rolling period. If a detector can cover more than one lane, one detector for a location is enough, for example, RTMS Radar. If a detector can only cover one lane, each lane will have a detector. The location of these detectors is called a detector station.

The delay calculations are shown in Equations 2 to 10. Equations 11 to 13 show calculations of volumes on mainline and alternative roads.

$$T_{Ff} = \frac{\sum_{j=0}^m DSF_{j,j+1}}{v_{Ff}} \quad (2)$$

$$T_{Fb} = \left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{1b}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{jb}} \right] \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{mb}} \quad (3)$$

$$T_{Fa} = \left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{ja}} \right] \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{ma}} \quad (4)$$

T_{Ff} : Travel time of affected mainline segment (segment from the off-exit to the on-exit of mainline road) under free flow speed (Veh-Hour).

v_{Ff} : Mainline free flow speed (mph).

T_{Fb} : Travel time of mainline before utilization of diversion guidance (Veh-Hour).

T_{Fa} : Travel time of mainline after utilization of diversion guidance (Veh-Hour).

$DSF_{j,j+1}$: Distance from detector j to detector $j+1$ (ft).

m : Total number of detectors (stations).

V_{ja} : Mainline speed at detector j after diversion control (after-case, mph), it is the forecasting value for next calculation interval; the speed will be calculated based on speed-density or flow-speed curve.

V_{jb} : Mainline speed at detector j before diversion control (before case, mph).

Freeway travel time of the base case is computed by the total distance between the diverging node and the merging node divided by free flow speed (Equation 2). This travel time is the minimal travel time under ideal traffic conditions. Freeway travel time after a bottleneck, or even after DMS control cases are calculated under similar method except for different speeds. Speeds from different detector stations are used to compute the travel time for the detector station coverage range (assuming speed will change immediately at the edge of detector effective segment). A detector station's volume and speed represent a small segment consisting of half the distance between the current detector and the previous one and half the distance between the current one and the one after it (Equations 3 and 4). Figure 2 illustrates the effective distance of a detector. L_4 in the figure is the effective distance of detector (detector station) number four,

$$L_4 = L_{34}/2 + L_{45}/2.$$

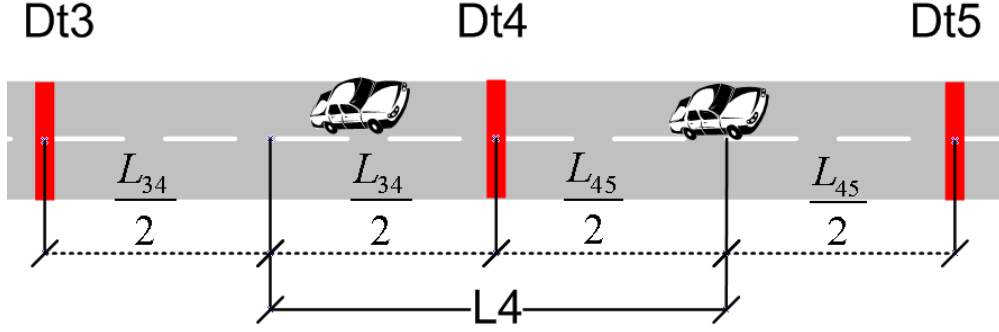


Figure 2 Illustration of Detector Effective Segment

$$d_{Fa} = T_{Fa} - T_{Ff} \quad (5)$$

$$D_{Fa} = d_{Fa} * V_{F1}$$

$$D_{Fa} = (T_{Fa} - T_{Ff}) * V_{F1}$$

$$D_{Fa} = \left[\left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fma}} \right] - \frac{\sum_{j=0}^m DSF_{j,j+1}}{v_{Ff}} \right] \times V_{F1} \quad (6)$$

$$d_{Fb} = T_{Fb} - T_{Ff} \quad (7)$$

$$D_{Fb} = (T_{Fb} - T_{Ff}) * V_{F0}$$

$$D_{Fb} = \left[\left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1b}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fjb}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fmb}} \right] - \frac{\sum_{j=0}^m DSF_{j,j+1}}{v_{Ff}} \right] \times V_{F0} \quad (8)$$

$$D_{Fb} = d_{Fb} * V_{F0} \quad (9)$$

$$\Delta D_{Fab} = D_{Fa} - D_{Fb} \quad (10)$$

d_{Fa} : Average delay of mainline under diversion control (Hour/Vehicle).

D_{Fa} : Total delay of mainline after diversion control (Veh-Hour).

D_{Fb} : Total delay of mainline segment before diversion control (Veh-Hour).

V_{F1} : Mainline volume after diversion (vph).

d_{Fb} : Average delay of mainline without diversion control (Hour/Vehicle).

ΔD_{Fab} : Travel time difference between mainline and alternative road (Hours).

Freeway travel delay for each vehicle is computed from the travel time difference between the base case and the DMS control case. Total delay is computed as the product of unit delay and traffic volume. Calculation of delay for each detector (detector station) effective segment follows the aforementioned approach for computing travel time (Equations 5 to 10).

$$V_{F0} = V_{F1} + V_{D1} \quad (11)$$

$$P_d = P(L_{DMS}) \quad (12)$$

$$V_{D1} = V_{F0} \times P(L_{DMS}) \quad (13)$$

$P(L_{DMS})$: The probability of diversion of a driver when he/she sees a diversion message at the level of L_{DMS} .

V_{F0} : Mainline demand (vph) before diversion (Vehicles).

V_{D1} : Diversion volume from mainline to alternative road, result from a message level and driver behavior distribution (Vehicles).

T_{Fcq} : Travel time of affected mainline segment, use speed-density curve under bottleneck situation and M/M/1 queue model before bottleneck calculation (Veh-Hour).

$P_{(L_{DMS})}$: Possibility of driver following a diversion message (from a certain level). Driver's choice follows Extreme Value distribution.

L_{DMS} : DMS message level.

The above method will not work well for the estimation of accumulated delays when the queue appears. The QuickZone (121) deterministic queue delay computation could be adopted for this situation. The following Equations 14 and 15 show the queue length and delay computation. Queue length of a computation period is the number difference between demand and capacity. Delay from the queue is the product of average queue length and period length.

$$Q_{Ti} = V_{Ti} - c_{Ti} \quad (14)$$

$$D_{Q_{Ti}} = \frac{Q_{Ti-1} + Q_{Ti}}{2} * T_i \quad (15)$$

T_i : i^{th} analysis period length (Hours).

Q_{Ti} : Queue length of analysis period T_i (Veh-Hour).

V_{Ti} : The constrained demand of analysis period T_i (Vehicles).

c_{Ti} : Work zone capacity, discharge rate is used as work zone capacity in this paper (pcphpl).

Q_{Ti-1} : Queue length of analysis period T_{i-1} (Vehicles).

$D_{Q_{Ti}}$: Queue delay of analysis period T_i (Veh-Hour).

In the real world, stochastic queue delay does appear when the demand is even less than the capacity. It is because of the uneven arrival distribution of the traffic. This part of delay also contributes to the total freeway travel delay. Average stochastic delay of a vehicle is calculated by the following Equation 16 (62) deducted from Queue Theory.

$$D_{Q_{Ti}} = \frac{V_{Ti}^2}{c_{Ti}(c_{Ti} - V_{Ti})} * T_i \quad (16)$$

Speed Prediction

Greenshield's traffic model is one of the famous traffic flow models, which requires only free flow speed and jam density for the speed-density relationship curve (61). Because it is too simple to reflect the real-world traffic trend, field collected data often deviates from the theory. As a result, improved models have been studied by some researchers. Van Aerde model (63) is used by some other researchers in practice for its good fitness of field-collected data. The model is shown in Equation 17.

$$\rho = \frac{1}{c_1 + \frac{c_2}{v_{Ff} - v} + c_3 \times v} \quad (17)$$

Where

ρ : Density (veh/km),

v : Speed (km/h),

v_{Ff} : Free flow speed (km/h),

c_1 : Fixed distance headway constant (km),

c_2 : First variable headway constant (km²/h),

c_3 : Second variable distance headway constant (h).

Houde et. al. (64) calibrated Van Aerde model by 1995 metropolitan Washington data and estimated c_1 , c_2 , c_3 to be 0.00512, 0.0144, and 0.000342, respectively. The speed-density curve is depicted respectively as the following Figure 3.

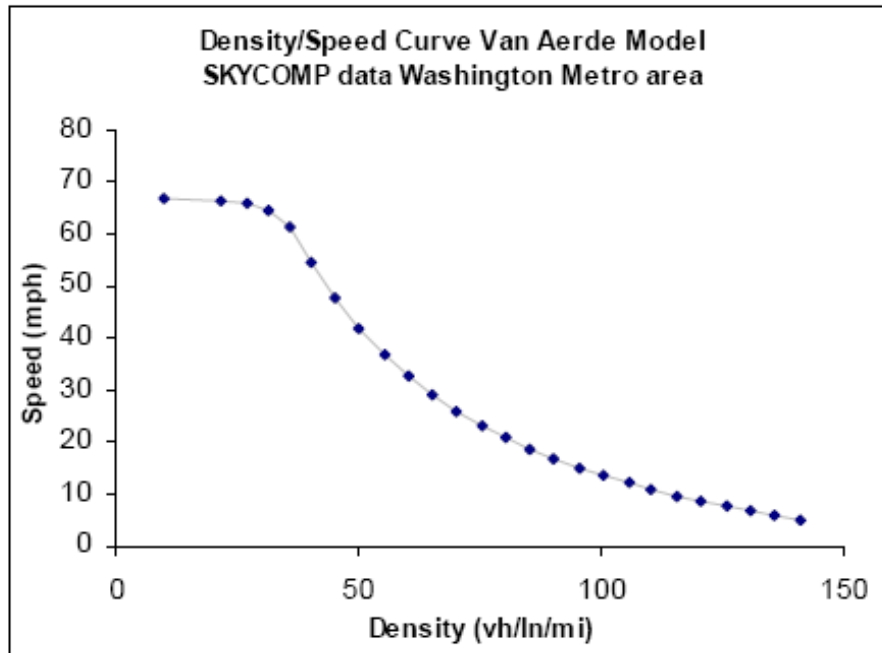


Figure 3 Speed-Density Curve from Van Aerde Model (64)

For this dissertation, the system will have an initial simulation run period for data collection (or a certain length of period in the real world). Once the system has accumulated data, specific c_1 , c_2 , c_3 for each case are fitted and used for later speed estimation. Figure 4 shows the curve fit of Van Aerde model from the CORSIM simulation data of studied example network; the fitted curve represents the field data well. The c_1 , c_2 , and c_3 of the data set are 0.0031990, 0.045, and 0.0003509, respectively.

As an alternative, if the density data is not available, the traditional Greenshield or improved flow-speed model is used for speed estimation from a given flow rate. Figure 5 shows the flow-speed data from CORSIM simulation of studied example networks. The Greenshield parabolic curve fitness for these data is not good enough to represent the relationship among this data. Improved flow-speed model are used to fit the curve and further serves as the base of speed forecast.

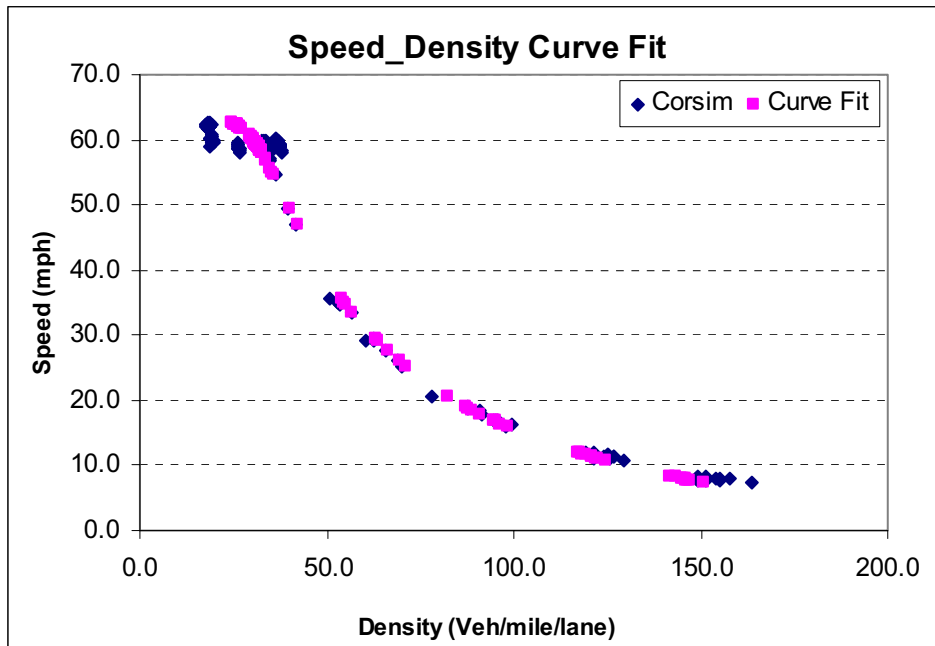


Figure 4 Speed-Density Curve Fitting from CORSIM

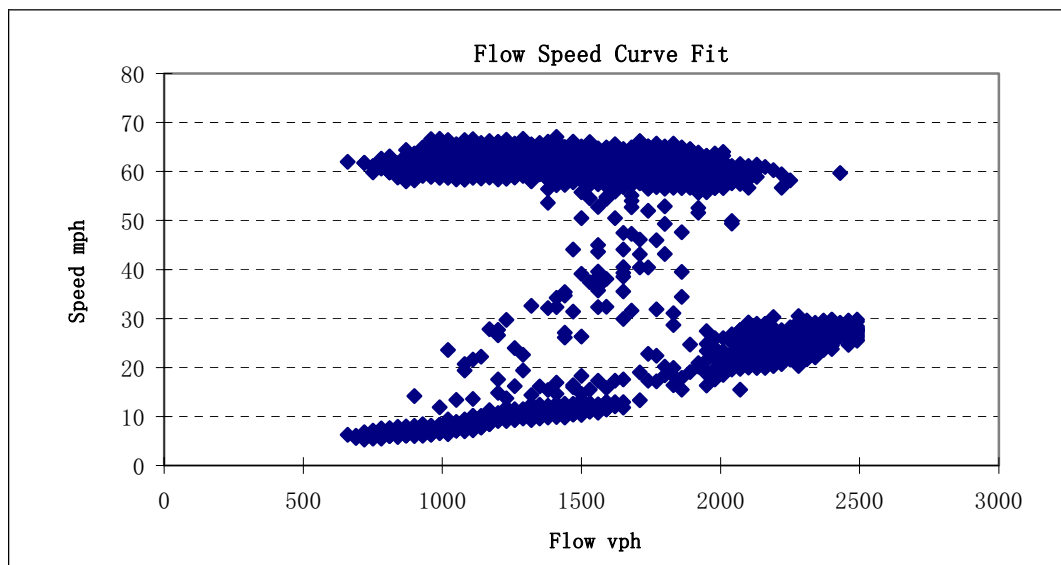


Figure 5 Flow-Speed Data from CORSIM Case Study

Alternative Road Signal Control Delay

A complicated part of objective function delay calculation is the alternative road signal control delay. A series of Webster's control delay equations in Highway Capacity Manual 2000 (HCM2000, 126) are adopted in this dissertation. If a better control delay model is applicable for this dissertation, the Webster's model will be replaced.

$$D_i = \sum_{a=1}^A \sum_{g=1}^G (PF * d_{1iga} + d_{2iga}) \times V_{ga} \quad (18)$$

$$D_i = \sum_{a=1}^A \sum_{g=1}^G \left(\left(\frac{0.50C_i(1 - gt_{iag}/C_i)^2}{1 - \left[\min(1, X_{iag}) \frac{gt_{iag}}{C_i} \right]} + 900T \left[\frac{(X_{iag} - 1) + \sqrt{(X_{iag} - 1)^2 + \frac{8kIX_{iag}}{c_{iag}T}}}{c_{iag}T} \right] \right) \times V_{iag} \right) \quad (19)$$

$$D_I = \sum_{i=1}^I D_i = \sum_{i=1}^I \sum_{a=1}^A \sum_{g=1}^G \left(\left(\frac{0.50C_i(1 - gt_{iag}/C_i)^2}{1 - \left[\min(1, X_{iag}) \frac{gt_{iag}}{C_i} \right]} + 900T \left[\frac{(X_{iag} - 1) + \sqrt{(X_{iag} - 1)^2 + \frac{8kIX_{iag}}{c_{iag}T}}}{c_{iag}T} \right] \right) \times V_{iag} \right) \quad (20)$$

$$C_i = \sum_a \sum_g gt_{iag} + L_i \quad (21)$$

$$X_{iag} = \frac{V_{iag}}{S_{iag}} \frac{C_i}{gt_{iag}} \quad (22)$$

D_i : Total delay of intersection i (Veh-Hour).

D_I : Total delay of intersections on alternative road (Veh-Hour).

PF : Uniform delay progression adjustment factor, which accounts for effects of signal progression. HCM2000 provides formula and table for PF value under different traffic conditions.

V_{iga0} : Volume of intersection i , approach a , group g , before diversion control (vph).

V_{iga1} : Volume of intersection i , approach a , group g , after diversion control (vph).

i : Alternative road intersection number.

d_{1iag} : Uniform control delay, assuming uniform arrivals for intersection i , lane group g and approach a , lane group g (Second/veh).

d_{2iag} : Incremental delay to account for effect of random and over saturation queues for intersection i , approach a , lane group g (Second/veh).

C_i : The i^{th} intersection cycle length (Second) used in pretimed signal. For the coordinated intersections, they share a same cycle length.

gt_{iag} : Effect green time for lane group (Second),
effect green time = green time + yellow – startup lost time – other lost time.

X_{iag} : v/c ratio or degree of saturation for lane group g of approach a at intersection i .

Alternative road Travel Time Difference

Alternative road travel time may be different from mainline roads for the diversion of traffic. The travel time difference will be negative if the mainline road has heavy traffic congestion or the alternative road is shorter than the mainline road (possibly a rare case). The travel time on an arterial road is more difficult to forecast than a freeway. More detectors might be needed to capture the alternative road traffic status. When freeway has demand higher than its capacity, the delay estimation becomes more difficult than that under free flow speed.

$$T_{Fa} = \left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{ja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{ma}} \right]$$

$j=2, 3, \dots m-1$

(23)

$$D_D = [T_{Fa} + D_{QT} - T_D] * V_{D1} \quad (24)$$

$$T_D = \frac{DS_{off}}{v_{off}} + \frac{DS_{on}}{v_{on}} + \sum_{i=1}^I \left(\frac{DSD_{i,i+1}}{v_{Dia}} \right) \quad (25)$$

D_D : Total travel delay of alternative road caused by the distance and speed difference between mainline and alternative road.

$DSD_{i,i+1}$: Distance from intersection i to $i+1$.

v_{Dib} : Alternative road speed at detector i before diversion control.

v_{Dia} : Alternative road speed at detector i after diversion control.

V_{D0} : Original alternative road traffic volume on diversion path.

V_{D1} : Diversion volume from mainline to alternative road.

$n1$: Number of alternative road detectors used in alternative road delay estimation.

v_{off} : Off-ramp speed (mph).

v_{on} : On-ramp speed (mph).

DS_{off} : Off-ramp distance (mile).

DS_{on} : On-ramp distance (mile).

T_{Db} : Alternative road moving travel time before diversion control (Veh-Hour).

T_{Da} : Alternative road moving travel time after diversion control (Veh-Hour).

Decision Variables

There are three sets of decision variables directly in objective function; the green time of every phase (including one or more traffic signal groups) is a vector whose size will vary with the number of studied intersections. In other words, the green time variable is not a single value variable; it is a set of

green times for all the studied intersection lane groups. On the other hand, the signal offset – the fourth decision variable – is not directly solved from the objective function because it is not included in the model directly. However, it is also necessary to change it under different traffic conditions. Therefore, a separate signal offset calculation step based on alternative road travel time is proposed during the implementation, though it is not showed in the equations.

L_{DMS} : DMS message level, $L_{DMS} \in [1,8]$. It has predefined diversion probability and the diversion volume caused by a certain DMS level is deduced from the predefined probability.

C : The intersection cycle length for pretimed signal control; there should be a same C for all the intersections $C_i = C$, or $C_i = l * C$ for some intersections, l is an integer (Second).

GT : The set of green times for all the studied intersections, gt_{iag} is the green time for lane group g of approach a at intersection i (Second).

O_i : Signal offset of the i^{th} intersection (Second).

Diversion message level is strictly related to diversion volume. The reason for using the DMS level instead of diversion volume is to make the DMS control more applicable. Nevertheless, the candidate algorithms can only return real number diversion volume directly. Since the diversion volume cannot be used for DMS directly, a predefined table (Table 1) to map the diversion volume and DMS messages (levels) is designed for converting the diversion volume to DMS message level.

State Variables

State variables here refer to those that are used to describe the traffic system state. For example, traffic demand, speed, flow rate, density, and volumes for all intersection approaches are treated as state variable. Most of the state variables are time dependent, which means that they should be updated at certain intervals, such as intersection rolling period, DMS message rolling period and so forth. The corridor traffic status could be collected by traffic detectors and described by the volume and speed variables.

It should be noted that the state variables are shown in the form of vector because there are many locations in the network that need state variable values. In other words, the state of a network cannot be described by several scalar variables; it needs a different series of the same type of variables to represent states all over the network. The state variables of a network spread over a span of time and space.

Driver Compliance

Driver compliance is the positive driver response to traffic information. Compliance is affected by many factors such as driver characteristics, information types, traffic condition and so forth. In this dissertation, the compliance with a message is described by extreme value distributions (the explanation of this distribution is in the following sections). The reason to choose the extreme value distribution to represent the driver compliance with DMS message is based on the fact that drivers are inclined to keep on the original route.

In this dissertation, driver compliance with DMS message is assumed to follow a particular distribution (for example, extreme value distribution) for a given message (message level). The random

variable is the possibility of drivers to take a diversion. The message level instead of detailed message for diver behavior is proposed here, because the messages rely on a particular case and will vary case by case.

Extreme value distribution is also known as log-Weibull distribution or Gumbel distribution.

Extreme value theory is important for assessing the risk for unusual events, such as 100-year floods. The driver behavior towards a diversion is assumed similar to the unusual events, namely, driver will be reluctant to divert. Therefore, the probability that a driver will choose a diversion beyond the expected diversion inclination of a given message level will be rare and its probability will drop sharply beyond the expected probability. In Figure 6, a CDF and PDF example of expectation diversion probability of 80.1 percent is depicted. It indicates that the actual range of distribution is fairly narrow and the overall distribution span is about 20 percent of the possible range. This characteristic is desirable for the driver compliance study, which means that for a specific expected diversion probability, only a few drivers have a greater possibility to divert than the expected diversion possibility. Most of drivers will have a smaller possibility to choose diversion than the expected one.

The parameters of the extreme value are *sigma* and *mu*. Parameter *mu* is the largest possibility density of the distribution and *sigma* is the span parameter that affects the shape of a curve. The bigger the *sigma*, the wider curve shape it is. Figure 6 shows the shape difference among *sigma*, two times of *sigma* and four times of *sigma* with a *mu* value of 0.8 and *sigma* value 0.01. The *sigma* value of 0.01 makes the driver diversion probability closely around the expected value, which means the variance of driver diversion activities are rather small. The *mu* value of 0.8 and the *sigma* value of 0.01 represent an expected diversion value of $\mu - \text{EULER} \times \sigma$; EULER represents the EULER constant that is 0.57722. The

parameter of μ has the highest possible density value and serves as the index of message levels. In other words, the values of μ are bonded with message levels after the following message level definition Table 1 is designed; once one of them is known, the other one can be inferred. Thus, it can guarantee the effectiveness of DMS control.

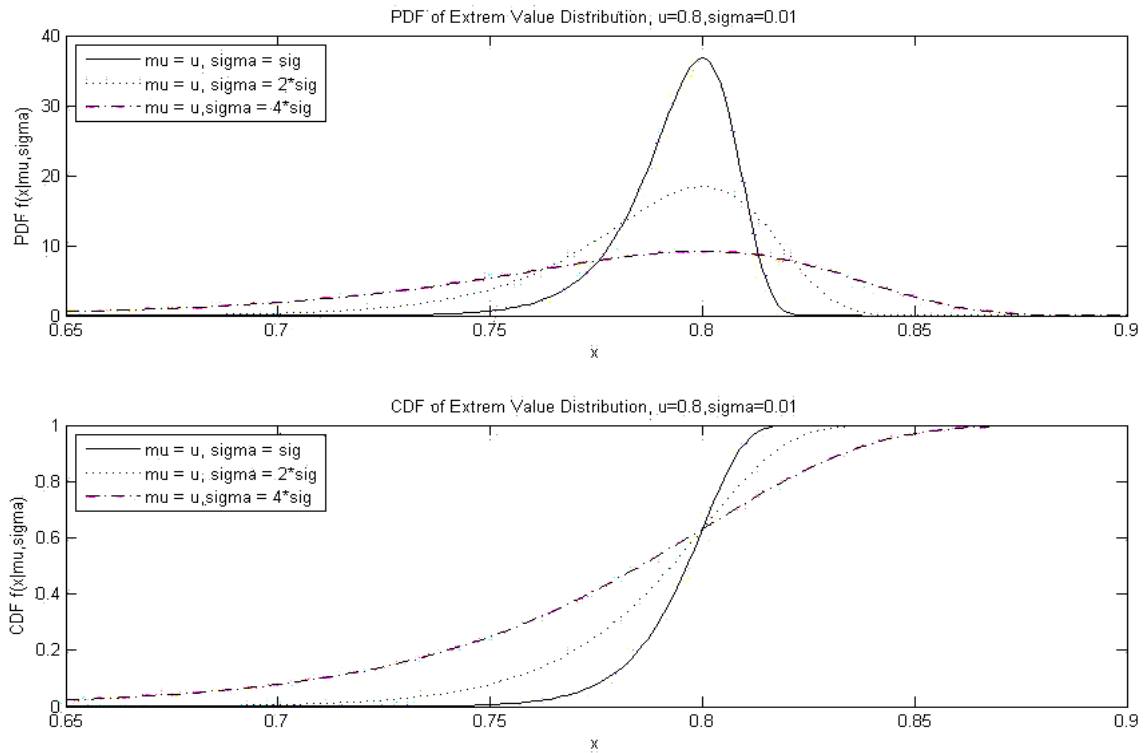


Figure 6 Example PDF and CDF of an Extreme Distribution

All the messages shown on the DMS board are classified into four categories and eight levels. Driver compliance distribution of each level is assumed extreme value distribution with different parameters (such as μ , σ value and so forth, Table 1). The mean value of the extreme value distribution is used as an indicator to choose a message level. In other words, after optimized diversion

volume is returned from optimization model, a DMS level will be selected that the product of its expected diversion value and total demand will be the closest one to the optimized diversion volume. The *sigma* parameter is same for all eight levels, which narrows the distribution to about a 10 percent span of the overall available distribution space and represents a smaller variance of driver diversion choices.

Table 1
DMS Message Levels and Expected Diversion Probability

Message Type	Message Content	Message Example	Expected Diversion Probability	Distribution Parameters
	L			
1 Advisory	1	General descriptive information	2.0%	Mu=0.01 Sigma=0.01
	2	Speed information	3.0%	Mu=0.02 Sigma=0.01
	3	Travel Information	4.0%	Mu=0.03 Sigma=0.01
2 Warning	4	Delay information on mainline	11.0%	Mu=0.10 Sigma=0.01
	5	Delay information on mainline and diversion advice	21.0%	Mu=0.20 Sigma=0.01
3 Optional	6	Delay information on mainline and general information on alternative road	31.0%	Mu=0.30 Sigma=0.01
	7	Delay information on mainline and timesaving from alternative road	41.0%	Mu=0.40 Sigma=0.01
4 Mandatory	8	Mandatory diversion with alternative road information	51.0%	Mu=0.50 Sigma=0.01

Solutions to Model

The proposed model is a nonlinear model with linear constraints and bounds; nonlinear constraint does not exist in the model. Many optimization algorithms can solve the nonlinear functions with linear constraints. MATLAB provides some algorithms such as Sequential Quadratic Programming (SQP), Pattern Search (PS) and Genetic Algorithm (GA) in its Optimization Toolbox™ and Genetic Algorithm and Direct Search Toolbox™ (67, 68). These three algorithms were used to solve the proposed model. Because of the speed requirement from the system's real-time characteristics, the fastest one with the best solution quality¹ among the three algorithms will be used for system implementation in the real world. It is necessary to make an extensive evaluation of algorithm efficiency, performance and solution quality during algorithm selection.

The proposed system is based on real-time information and will be updated at a certain short time interval (for example, two minutes or less). The efficiency of the algorithm is critical here. The maximum algorithm execution time needs to be limited to two minutes for a reasonable frequency of signal updates. A simple network configuration was employed for algorithm efficiency and solution quality evaluation of these three candidate algorithms from MATLAB toolboxes.

¹ solution quality in this dissertation mainly means the degree of corridor delay reduction.

Algorithms Introduction

MATLAB provides a series of algorithms in the Optimization Toolbox™ for different types of problems. At the same time, it also provides Genetic Algorithm and Direct Search Toolbox™ to extend the capabilities of Optimization Toolbox™.

Traditional optimization methods utilize the gradient or higher derivatives to search for an optimal point. Unlike the traditional optimization algorithms, direct search solves optimization problems without any information about the gradient of the objective function. A direct search algorithm searches a set of points around the current point to find a new point that has a better objective function value than the current one. The direct search method can work over non-differentiable or even non-continuous objective function.

There are three algorithms in the Genetic Algorithm and Direct Search Toolbox™: Pattern Search, Genetic Algorithm and Simulated Annealing. Genetic Algorithm is adopted in TRANSYT-7F for signal optimization; therefore, it was also treated as a candidate algorithm for the proposed model in order to compare with the TRANSYT-7F solutions. Simulated Annealing algorithm is not as popular as GA in the signal optimization area and it does not work very well in the optimization domain. Therefore, it was not chosen for evaluation. Pattern Search algorithm is also tested in this dissertation because it is a relatively new algorithm developed in 1997 (81).

The proposed model is a nonlinear problem with linear equity and bounds constraints. It is also a discrete problem and does not have many variables. The function *fmincon* (constrained minimal solution of objective function) in the Optimization Toolbox™ is designed for this type of problems. The *fmincon* is a gradient-based method and it uses SQP algorithm with Hessian matrix for optimal solution search.

In general, an optimization problem can be described by an objective function $f(x)$ that is either minimized or maximized. The problem might be constrained by some equality constraints, inequality constraints and/or bounds. A General Problem (GP) for optimization study is stated as the following (67):

$$\text{Min } f(x) \quad (26)$$

ST.

$$g_i(x) = 0 \quad i=1, \dots, m_e, \quad (27)$$

$$g_i(x) \leq 0 \quad i= m_e+1, \dots, m \quad (28)$$

Where x is the independent variable vector, $f(x)$ is the objective function, and vector function $G(x)$ returns a length m vector of equality and inequality constraints. Boundary is not separately listed here.

However, it can be treated as unequal constraints and transforms. Specifically, the constraints are more detailed as the following equations.

$$c(x) \leq 0 \quad (29)$$

$$ceq(x) = 0 \quad (30)$$

$$A x \leq b \quad (31)$$

$$Aeq x = beq \quad (32)$$

$$lb \leq x \leq ub \quad (33)$$

Where x , b , beq , lb , ub , solution from function $f(x)$, $c(x)$ and $ceq(x)$ are vectors; A and Aeq are matrices. $f(x)$, $c(x)$, and $ceq(x)$ can be either linear or nonlinear functions. Among the constraints, Equations 29 and 30 are nonlinear inequality and equality constraints; Equation 31 and 32 are linear inequality and

equality constraints; Equation 33 is the bounds of independent variables. For the proposed model, there are not nonlinear and linear inequality constraints like Equation 29 and 31.

Sequential Quadratic Programming (SQP)

The SQP algorithm is employed by some solvers like *fmincon*, *fminimax*, *fgoalattain*, and *fseminf* in MATLAB Optimization Toolbox™. *fmincon* is the one used to solve the proposed model. SQP algorithm instead of the *fmincon* is referred from now on for algorithm introduction. SQP transforms the original problem to a series of quadratic programming (QP) sub-problems based on a quadratic approximation of the Lagrangian function. At each step, Hessian matrix is estimated and updated using the BFGS quasi-Newton updating method and a sub-problem is then resolved. The solution of QP sub-problem helps to guide the optimal direction search (86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, and 97).

The *fmincon* is gradient-based method and it can only handle real number types of variables, which implies that it works for continuous problems and constraints that have continuous first and second derivatives. The *fmincon* cannot guarantee global optimal solution and might only give a local optimal solution. However, if the objective and constraints are the so-called convex functions, Karush-Kuhn-Tucker (KKT) equations are both necessary and sufficient for a global solution point. It implies that the *fmincon* may find the global optimal solution for certain convex problems with convex constraints functions.

As stated in the Optimization Toolbox™ 4 User's Guide (67), the principal idea of SQP used in *fmincon* is the QP subproblem formulation based on a quadratic approximation of the Lagrangian function (34).

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \cdot g_i(x) \quad (34)$$

Where $L(x, \lambda)$ is the Lagrange function, $f(x)$ is the objective function, λ is vector of the estimation of Lagrange multipliers and the $g(x)$ is the constraint functions.

The Quadratic Programming (QP) subproblem of the Lagrange function $f(x)$ is described as following and it can be resolved by a QP algorithm.

$$\begin{aligned} \min_{d \in R^n} & \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d \\ \nabla g_i(x_k)^T d + g_i(x_k) &= 0, \quad i = 1, \dots, m_e \\ \nabla g_i(x_k)^T d + g_i(x_k) &\leq 0, \quad i = m_e + 1, \dots, m \end{aligned} \quad (35)$$

The solution of QP problem gives the search directions. Search step length will be determined by a line search procedure to obtain a sufficient decrease in the merit function. The new solution point is then computed by Equation 36.

$$x_{k+1} = x_k + \alpha_k d_k \quad (36)$$

Where α_k is the step length parameter and the d_k is the search direction at step k .

At each major step, Hessian of the Lagrangian function $L(x, \lambda)$ is approximated by positive definite quasi-Newton method BFGS (Equation 37).

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T s_k s_k^T H_k}{s_k^T H_k s_k} \quad (37)$$

Where:

$$s_k = x_{k+1} - x_k \quad (38)$$

$$q_k = \left[\nabla f(x_{k+1}) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(x_{k+1}) \right] - \left[\nabla f(x_k) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(x_k) \right] \quad (39)$$

In the SQP implementation in MATLAB *fmincon* function, $q_k^T s_k$ is kept positive at each update to keep the Hessian positive definite, even though it might be positive indefinite at the solution point. A procedure repeatedly halves the most negative element of $q_k^T s_k$ until it is greater than or equal to a small negative tolerance. If $q_k^T s_k$ is still not positive after the above procedure, q_k will be modified by adding a vector v multiplied by a constant scalar w .

$$q_k = q_k + wv \quad (40)$$

Where:

$$v_i = \nabla g_i(x_{k+1}) \cdot g_i(x_{k+1}) - \nabla g_i(x_k) \cdot g_i(x_k) \quad \text{if } (q_k)_i \cdot w < 0 \text{ and } (q_k)_i \cdot (s_k)_i < 0 \quad i=1, \dots, m$$

$$v_i = 0 \quad \text{Otherwise}$$

The w will be systematically increased until $q_k^T s_k$ becomes positive.

At the same time, a QP problem of search direction in the following form is solved at each major SQP step by active set method (projection method) to provide search direction d for the QP subproblem.

$$\begin{aligned} \min_{d \in R^n} & \frac{1}{2} d^T H_k d + c^T d, \\ & A_i d = b_i, \quad i = 1, \dots, m_e, \\ & A_i d \leq b_i, \quad i = m_e + 1, \dots, m. \end{aligned} \quad (41)$$

There are two phases in the solution procedure. A feasible point is computed in the first phase and iterative sequences of feasible pointes that make the solution convergence are generated in the second phase. The active set \bar{A}_k is an estimation of the active constraints on the constraint boundaries at the solution point. \hat{d}_k is formed based on \bar{A}_k that is updated at each iteration k . A feasible subspace for \hat{d}_k is formed based on Z_k that satisfies $\bar{A}_k Z_k = 0$, which means that Z_k is orthogonal to \bar{A}_k . Z_k is formed

from the last number of inactive constraints in the QR decomposition matrix of \bar{A}_k . In other words, Z_k is obtained from

$$Z_k = Q[:, l+1, m], l < m, \quad (42)$$

Where l is the number of active constraints and the Q satisfies the following equation,

$$Q^T A_k^T = \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (43)$$

After the Z_k is found, the search direction \hat{d}_k is sought from quadratic function of p by the projected gradient method of quadratic function.

$$\hat{d}_k = Z_k p \quad (44)$$

Where p can be obtained by differentiating the following quadratic function and set it to zero.

$$q(p) = \frac{1}{2} p^T Z_k^T H Z_k p + c^T Z_k p \quad (45)$$

$$\frac{\partial q(p)}{\partial p} = \nabla q(p) = Z_k^T H Z_k p + Z_k^T c \quad (46)$$

$$Z_k^T H Z_k p = -Z_k^T c \quad (47)$$

Once \hat{d}_k is obtained, the new solution point is computed by the following Equation 48,

$$x_{k+1} = x_k + \alpha \hat{d}_k \quad (48)$$

Where α have only two choices for the quadratic nature of the objective function. When the constraints are in the active set \bar{A}_k , if one unit along the \hat{d}_k steps to the minimum of the function restricted to the null-space of active set without violation of the constraints, then this is the solution of QP. Otherwise, α will be calculated by the following Equation 49.

$$\alpha = \min_{i \in \{1, \dots, m\}} \left\{ \frac{-(A_i x_k - b_i)}{A_i d_k} \right\} \quad (49)$$

The solution to the QP problem of search direction produces a search direction vector d_k for QP sub-problem of objective function. A new iterate for the QP sub-problem will be formed if α_k is obtained from a merit function. The α_k is determined to produce a sufficient decrease in the merit function (Equation 50).

$$\Psi(x) = f(x) + \sum_{i=1}^{m_e} r_i \cdot g_i(x) + \sum_{i=1}^{m_e} r_i \cdot \max[0, g_i(x)] \quad (50)$$

Where r_i is given by

$$r_i = (r_{k+1})_i = \max_i \left\{ \lambda_i, \frac{(r_k)_i + \lambda_i}{2} \right\}, i = 1, \dots, m \quad (51)$$

The initial r_i is obtained by

$$r_i = \frac{\|\nabla f(x)\|}{\|\nabla g_i(x)\|} \quad (52)$$

Where the $\| \cdot \|$ is the Euclidean norm.

Pattern Search

Pattern Search algorithm is a direct search method that solves optimization problems without any information about the gradient of the objective function and its constraints.

The pattern search tool in Genetic Algorithm and Direct Search Toolbox™ uses the Augmented Lagrangian Pattern Search (ALPS) and Pattern Search algorithm to solve nonlinear constraint problems. At each step, a sub-problem is formulated by combining the objective function and nonlinear constraint

functions using the Lagrangian penalty parameters. A sequence of such optimization problems are approximately minimized using the pattern search algorithm with linear constraints and bounds satisfied (77, 81, 82, 83, 84, and 85).

In fact, there are two pattern search algorithms called the generalized pattern search (GPS) algorithm and the mesh adaptive search (MADS) algorithm. Generalized pattern search (GPS) algorithm was defined and analyzed by Torczon in 1997 (81). The MADS algorithm is a modification of the GPS algorithm. These two algorithms differ in how the set of mesh points are formed and computed. The GPS algorithm uses fixed direction vectors, however the MADS algorithm uses a random selection of vectors to form the mesh.

The main steps of General Pattern Search algorithm is described as the followings:

Let $x_0 \in \mathbf{R}^n$ and initial step length Δ_0 be greater than zero.

For $k = 0, 1, \dots$

 Compute the objective function $f(x_k)$ value at x_k .

 Determine a step s_k .

 Compute ρ_k , $\rho_k = f(x_k) - f(x_k + s_k)$.

 If $\rho_k > \varepsilon$ and $\rho_k > 0$

$$x_{k+1} = x_k + s_k$$

 else if $\rho_k \leq 0$

$$x_{k+1} = x_k$$

 else if $\rho_k \leq \varepsilon$

 stop

 end if

Update C_k and Δ_k

End for

Where

\mathbf{R}^n is the n dimension set of real numbers,

Δ_k is step length,

ε is tolerance precision,

C_k is generating matrix,

B is basic matrix,

s_k is a step of x_k . $s_k = \Delta_k B C_k$.

Pattern Search computes a sequence of points around a solution point. At each step, the algorithm searches a set of points (mesh) around the current point - the point with an improved objective function value at the previous step.

A pattern is defined as a set of vectors $\{v_i\}$ that the pattern search algorithm uses to determine which point to search at each iteration. The set $\{v_i\}$ is defined by the number of independent variables in the objective function, N , and the positive basis set. A mesh is formed by adding a pattern to the current point. If the pattern search algorithm finds a point in the mesh that improves the objective function value at current point, the new point becomes the current point for next iteration.

During the search, actions of polling, expanding and contracting are conducted repeatedly. Polling is the action to search for possible objective function value among the current mesh. The algorithm could stop polling the mesh points when it finds a candidate point whose objective function value is less than that of the current point. Alternatively, after it enumerates all the mesh points for a best candidate point whose objective function value is the least. Expanding and contracting are actions following polling to change the

value of mesh size. After a successful poll (a new point with less objective function value than the current point), the mesh will be expanded to a certain size by multiplying a scalar that is defined by user (>1).

Otherwise, the mesh will be contracted after an unsuccessful poll by multiplying a scalar defined by user (<1).

The searching will stop if some criteria are met. Either of the following criteria will stop the searching:

- The performed iterations reach the user-defined maximal iterations.
- The mesh size is smaller than the user-defined minimal mesh size.
- The total number of evaluated objective function values reaches the user defined maximal number of function evaluations.
- Algorithm running time reaches the user-defined time limit.
- The distance between two X s from consecutive iterations is less than X tolerance while there are no more mesh points to explore.
- Change of objective function from two consecutive iterations is less than the user defined function value tolerance while there are no more mesh points to explore.

Generic Algorithm

Genetic Algorithm was developed by John Holland in 1975. Genetic Algorithm is a kind of evolution algorithm and it uses techniques derived from Darwin's principle of "survival of the fittest". In other words, characteristics that are helpful for survival have higher possibility to pass on to the coming generations because individuals carrying them have a higher possibility to have children. During the evolution, reproduction is critical for the algorithm progress. There are three ways to generate children in GA (78, 79, and 80):

Elite children are the individuals in the current generation with the best fitness values. These individuals will become children in next generation. The number of elites is specified by user during the implementation.

Crossover children are generated by combining parts of a pair of selected parents.

Mutation children are generated by conduction random mutations from a single parent.

At each step, the Genetic Algorithm generates children using parents picked from the current population. Parents are selected based on fitness values; the better fitness value an individual has, the higher possibility to be picked as parents.

Figure 7 shows the outline of the GA algorithm. The algorithm begins with creating (or giving) a random initial population. Then, the algorithm will evaluate the new populations. Finally, the algorithm creates a sequence of new individuals that will form a new generation of populations and starts the next iteration.

At each step, the algorithm uses individuals with better fitness value in current generation to create next population. To create a new population, the algorithm first scores each individual of current population by computing their fitness value. Second, the algorithm selects individuals based on their fitness. After parents are selected, children will be produced from the aforementioned three methods and current population will be replaced by the new generated population.

The GA stop criterion could be one of the followings:

- The number of performed iterations reaches the user-defined maximal iterations.
- Algorithm running time reaches the user-defined time limit.
- The objective function change from two consecutive iterations is less than function change tolerance.

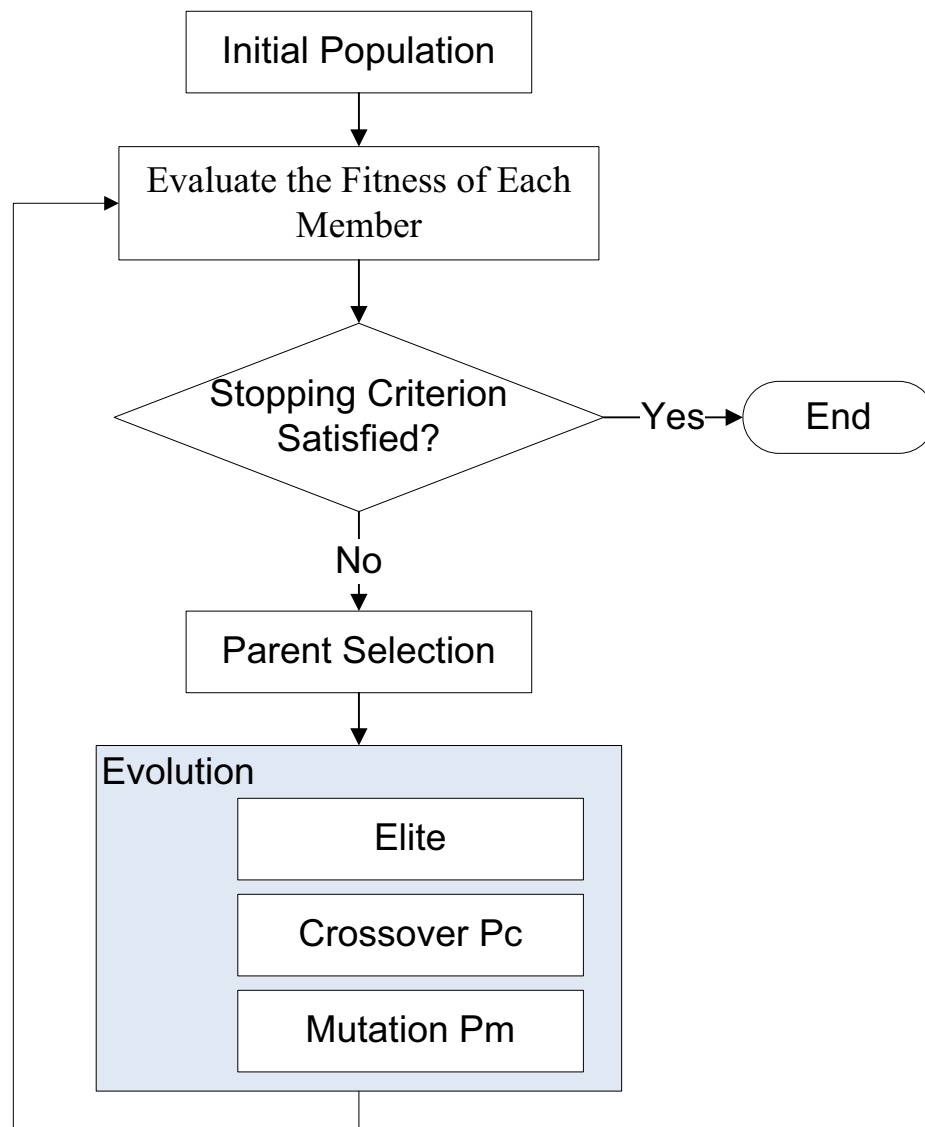


Figure 7 Flowchart of Genetic Algorithm

Instead of jumping from one point to another, Genetic Algorithm searches from the whole population of points. In other words, Genetic Algorithm codes all the independent variables into an individual by either the bits or vector method. Genetic reproduction is based on crossover and mutation probability rather than the deterministic search mechanism. Though Genetic Algorithm is not classified as direct search, it still has some characteristics similar to direct search. It does not use the gradient information to search and does not require the objective function to be continuous. Unlike the SQP algorithm, it is impossible to tell whether a solution point is the global optimal point or not.

For both Pattern Search and Genetic Algorithm in MATLAB, Augmented Lagrange Multiplier Method is used to create a sub-problem at each step (Equation 53). A sub-problem is formulated by combining the objective function and the nonlinear constraints function using the Lagrangian and the penalty parameters. The solutions from sub-problem guide the search direction (67, 68).

$$\Theta(x, \lambda, s, \rho) = f(x) - \sum_{i=1}^m \lambda_i s_i \log(s_i - c_i(x)) + \sum_{i=m+1}^{mt} \lambda_i c_i(x) + \frac{\rho}{2} \sum_{i=m+1}^{mt} c_i(x)^2 \quad (53)$$

Where:

λ_i The Lagrange multiplier, nonnegative.

s_i Nonnegative shift.

ρ Positive penalty parameter.

c_i Constraints.

However, this step does not occur in the solution of proposed model since nonlinear constraints do not exist in the model.

Algorithms Evaluation

The algorithms are initially evaluated from a simplified numerical example in Chapter Three.

There are more detailed solution evaluations based on CORSIM MOEs in Chapter Five. CORSIM simulation acts as a uniform data source and platform for evaluations of all these three algorithms.

For the simple evaluation, all the variables and constraints, etc. are on the same plane as the model solution in the Chapter Five. The only difference is that they are executed out of CORSIM RTE environment and is executed only once in MATLAB. The evaluation results are listed in Table 2 and Table

3. Parameters X_0 and X^* in the Table 3 represent initial values and final solution vector, respectively.

Table 2

Comparison on Optimal Solutions from a Random Initial X_0 and an Optimal Solution X^* as an Initial X_0 for
a New Search

No.	15 min	X_0	SQP x_0^*	SQP $x^{*'}_0$	PS x^*	PS $x^{*'}_0$	GA x^*	GA $x^{*'}_0$	Remark
1	722	782	1126	1126	588	588	863	867	Diversion volume (vph)
2	296	32	240	240	240	240	242	241	Cycle length (Seconds),
3	284	34	190	190	188	188	181	187	Green time of Intersection 1 (Seconds)
4	170	67	44	44	46	46	55	48	
5	166	36	44	44	46	46	55	48	
6	229	16	149	149	140	140	144	144	Green time of Intersection 2 (Seconds)
7	90	31	26	26	30	30	25	25	
8	70	16	149	149	140	140	144	144	
9	125	31	26	26	30	30	25	25	
10	38	16	15	15	18	18	22	21	
11	156	64	38	38	40	40	39	39	
12	40	16	15	15	18	18	22	21	
13	144	64	38	38	40	40	39	39	Green time of Intersection 3 (Seconds)
14	38	11	20	20	21	21	25	25	
15	135	45	42	42	43	43	41	42	
16	41	11	20	20	21	21	25	25	
17	158	45	42	42	43	43	41	42	
18	84	18	36	36	44	44	60	48	
19	196	51	130	130	120	119	104	114	
20	35	18	36	36	44	44	60	48	
21	356	51	130	130	120	119	104	114	Green time of Intersection 4 (Seconds)
22	54	21	30	30	39	39	29	36	
23	140	38	34	34	37	37	34	33	
24	82	21	30	30	39	39	29	36	
25	114	38	34	34	37	37	34	33	
26	68	26	128	128	112	112	103	105	
27	151	41	36	36	40	40	65	55	
28	200	26	128	128	112	112	103	105	
29	80	41	36	36	40	40	65	55	

x_0^* optimal solution given a random initial X_0 . $x^{*'}_0$ optimal solution given an solution as an initial X_0 .

Table 2 shows that all the three algorithms generate a fairly similar cycle length. However, the diversion volumes show big differences. Three algorithms generate three different DMS levels. The green times from all three algorithms provide the necessary time extension for the diversion volume on all intersections. It is hard to say which one yields the best solution at the current evaluation stage, even though the objective function values are known. Because values of the objective functions are the estimated system delay that may include bias or error from estimated delay. The quality of the solution will be further evaluated in Chapter Five by CORSIM MOE outputs.

Because of the increased diversion volume, the groups on the diversion path should get longer green time to accommodate the increased volume. The groups on the diversion path get longer green time from all the three algorithms, which show the three algorithms can generate reasonable signal timing plan. It needs to be pointed out that the SQP and PS generate a unique solution if an optimal solution is used for a new search. It means that they are trapped around a local optimal solution or they capture the unique global optimal solution. In contrast, GA generates a different solution set each time, which shows the GA solution's stochastic property.

Table 3 shows the iterations, number of evaluated fitness functions and the algorithm execution time. It is obvious that the SQP has the fastest speed among the three algorithms (for the given stopping criteria). For a given random initial X_0 , it can find a solution in fifteen seconds. When reasonable initial value of X_0 is given (a solution from other runs), it can find a solution in five seconds. The good initial value of X_0 here is the optimal solution from former executions. The purpose of setting an optimal solution point as the starting point is to test the algorithm's convergence and solution randomness.

Table 3

Comparison on Execution Properties from a Random Initial X_0 and an Optimal Solution X^* as an Initial X_0 for a New Search

Execution Property	SQP			PS			GA		
	Random Initial X_0	Optimal X^* as X_0	%	Random Initial X_0	Optimal X^* as X_0	%	Random Initial X_0	Optimal X^* as X_0	%
Iterations	51	13	25.49%	183	49	26.78%	150	102	68.00%
Function Counts	1567	420	26.80%	4047	1063	26.27%	3001	2041	68.01%
Elapsed time (Seconds)	14.50	4.28	29.54%	86.91	14.87	17.11%	35.39	23.64	66.79%

PS has the longest execution time (87 seconds) for a random initial X_0 . Nevertheless, it has biggest speed improvement for a good initial value of X_0 (the new execution time is only 17.11 percent of the original execution time). For the GA algorithm, a good initial value of X_0 does not save too much time (about 33 percent reduction in execution time). In contrast, SQP gets 70 percent decrease and PS gets 82 percent decrease in execution time given an optimal solution from other executions as an X_0 . The reason for the GA's lower execution time is from the GA's stochastic properties of parents' selection and children generation. As a conclusion, Table 3 shows that the quality of initial X_0 is critical for SQP and PS, and it is important to select a better initial X_0 for a faster solution search. However, problem also exists that the solution may be trapped around the given initial value of X_0 , if a local optimal solution is so close to the initial value that the variance tolerance of variables are easily satisfied for a termination.

The *fmincon* solver in the Optimization Toolbox™ is gradient-based method and it can only deal with real number variables. The proposed model gets all real variables if using diversion volume instead of DMS level. Therefore, SQP is suitable for the proposed model. Because the objective function is continuous and its first and second deviation can be calculated, this method will have more information regarding directions of search and step length. Hence, it could result in a faster search than the ones without using such information from objective and constraints functions. If the function convex property can be proved, it is also possible to know if the solution is global or local optimal from Karush-Kuhn-Tucker (KKT) equations.

The SQP's faster than other algorithms' execution speed is also tested by Hock and Schittkowski (95). A version of SQP algorithm was implemented and tested over general constrained nonlinear optimization problem. SQP algorithm outperformed every other tested method, namely the Penalty method (PE), Multiplier or augmented Lagrangian method (MU), Generalized reduced gradient method (GRG), Lagrangian method of Robinson (LA), in terms of accuracy, efficiency, efficiency related to accuracy and reliability over 119 test problems.

Based on the aforementioned characteristics of SQP algorithm, it is chosen as an essential implementation algorithm for the proposed model, while the Pattern search and Genetic algorithm will also be tested for comparison purpose.

CHAPTER IV

SYSTEM IMPLEMENTATION

“Traffic simulation models use numerical techniques on a digital computer to create a description of how traffic behaves over extended periods of time for a given transportation facility or system”. This is the description of a simulation model stated by HCM 2000 (126). In order to take advantage of the simulation model strengths, this proposed system is developed and evaluated with a microscopic traffic simulation software package - CORSIM. At the same time, the MOE output from CORSIM provides a unique standard for model evaluation for different scenarios, algorithms, or even solutions from different software.

This chapter mainly introduces the system implementation based on CORSIM. The RTE program is coded in C++, which in turn is interconnected with MATLAB environment. The RTE system structure, model structure and data flow are introduced sequentially. CORSIM RTE configuration for call points, RTE execution procedure and some techniques employed in the implementation are also introduced in this chapter.

System Structure

After evaluation of algorithms, three algorithms suitable for the model are coded in MATLAB and are called from RTE. This DMS control system is an integrated system; it runs the C++ coded frame, a developed CORSIM RTE and MATLAB simultaneously. The main program will start CORSIM and execute the Run Time Extension (RTE) to exchange data with MATLAB and update network operation parameters of CORSIM simulation. The RTE passes system state data to the MATLAB optimization package at the end of certain period (for example, signal roll period). Once the optimal solution set is found by the optimization algorithm in MATLAB, the value will be passed back and the CORSIM operation parameters will be updated by RTE right away. CORSIM will keep running with the new system parameters for the next simulation period. When the MATLAB is running, the CORSIM will wait until MATLAB ends. The above steps will be repeated from one analysis period to another period.

Figure 8 shows the program structure of the proposed system. The main program will start the CORSIM simulation over the TRANSYT-7F optimized network. During the simulation, the MATLAB code of model is called repeatedly. The repeated data exchange is conducted between the CORSIM RTE and MATLAB until the user-defined criteria is met. The CORSIM MOE outputs from the TRANSYT-7F and RTE are finally compared for system evaluation.

Figure 9 and Figure 10 shows the system architecture, and data flow. This system starts from data collection from the network, and ends at the DMS message display for each period. Data processing, corridor travel delay optimization and DMS message selection are the critical steps for this system. The data is first collected from the network and then processed for the model's computation. Then the model

will return the solution to reduce corridor traffic delay. Finally, the DMS message (level) is selected and displayed on the DMS board.

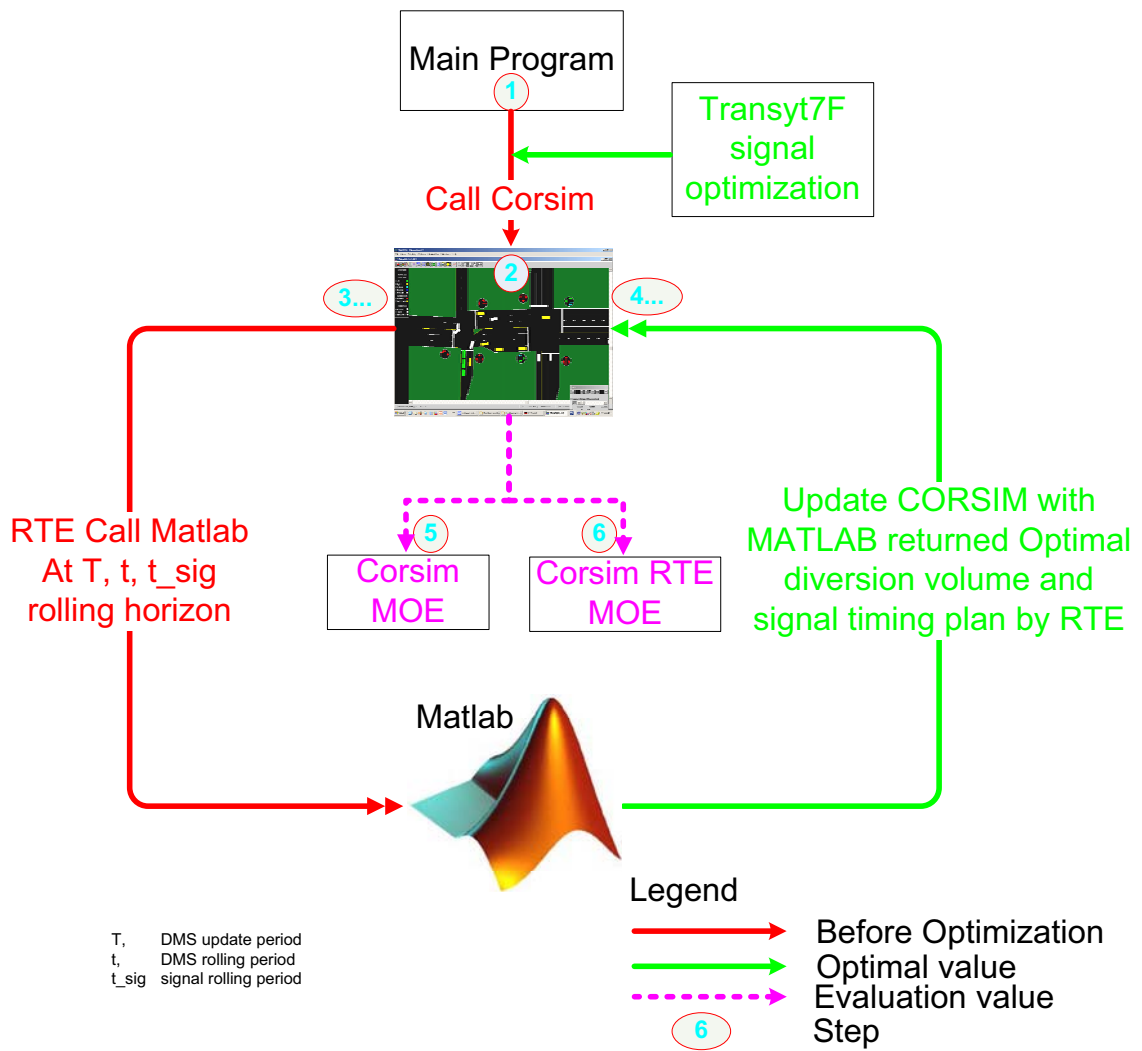


Figure 8 System Structures of Modules

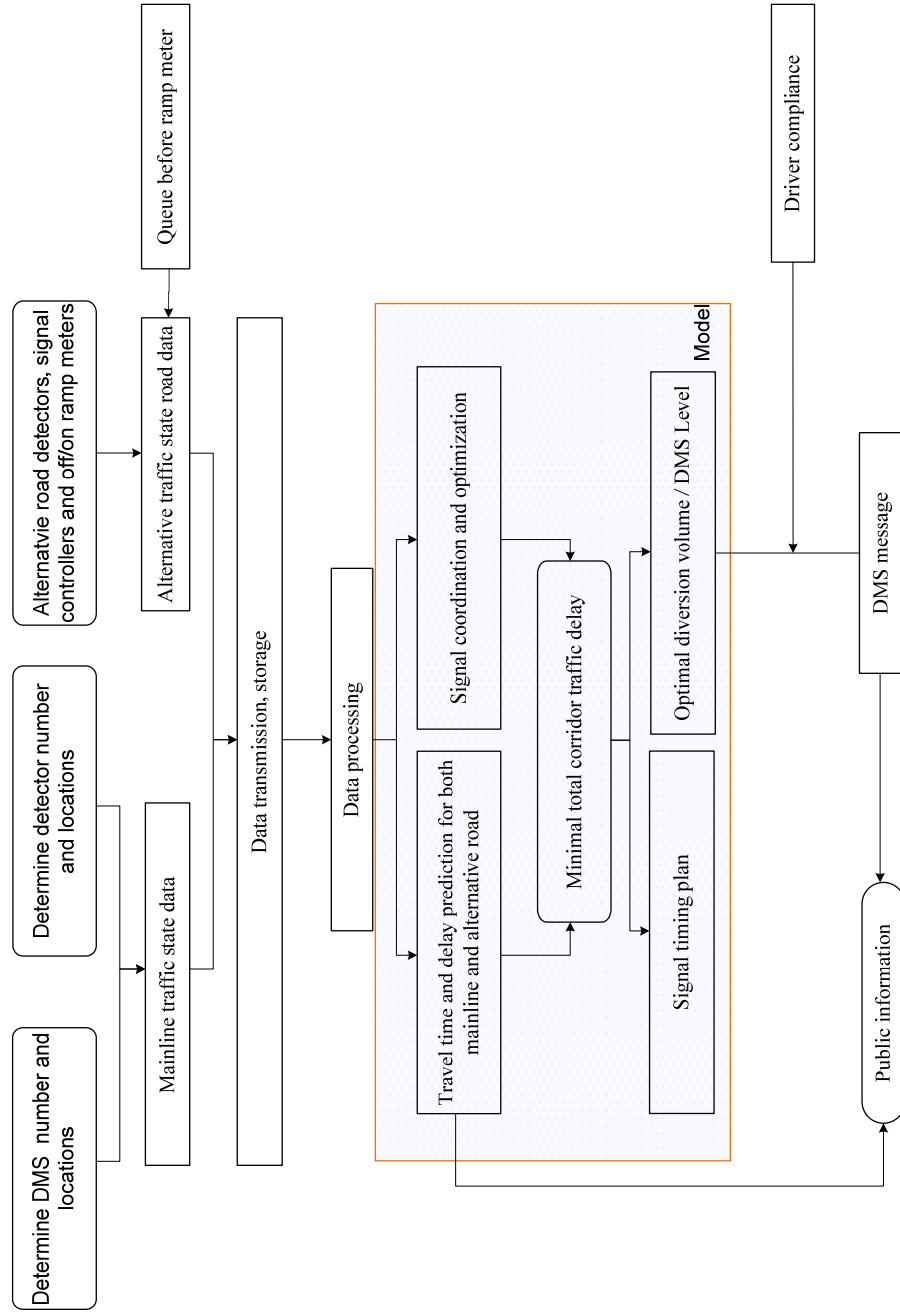


Figure 9 System Structure

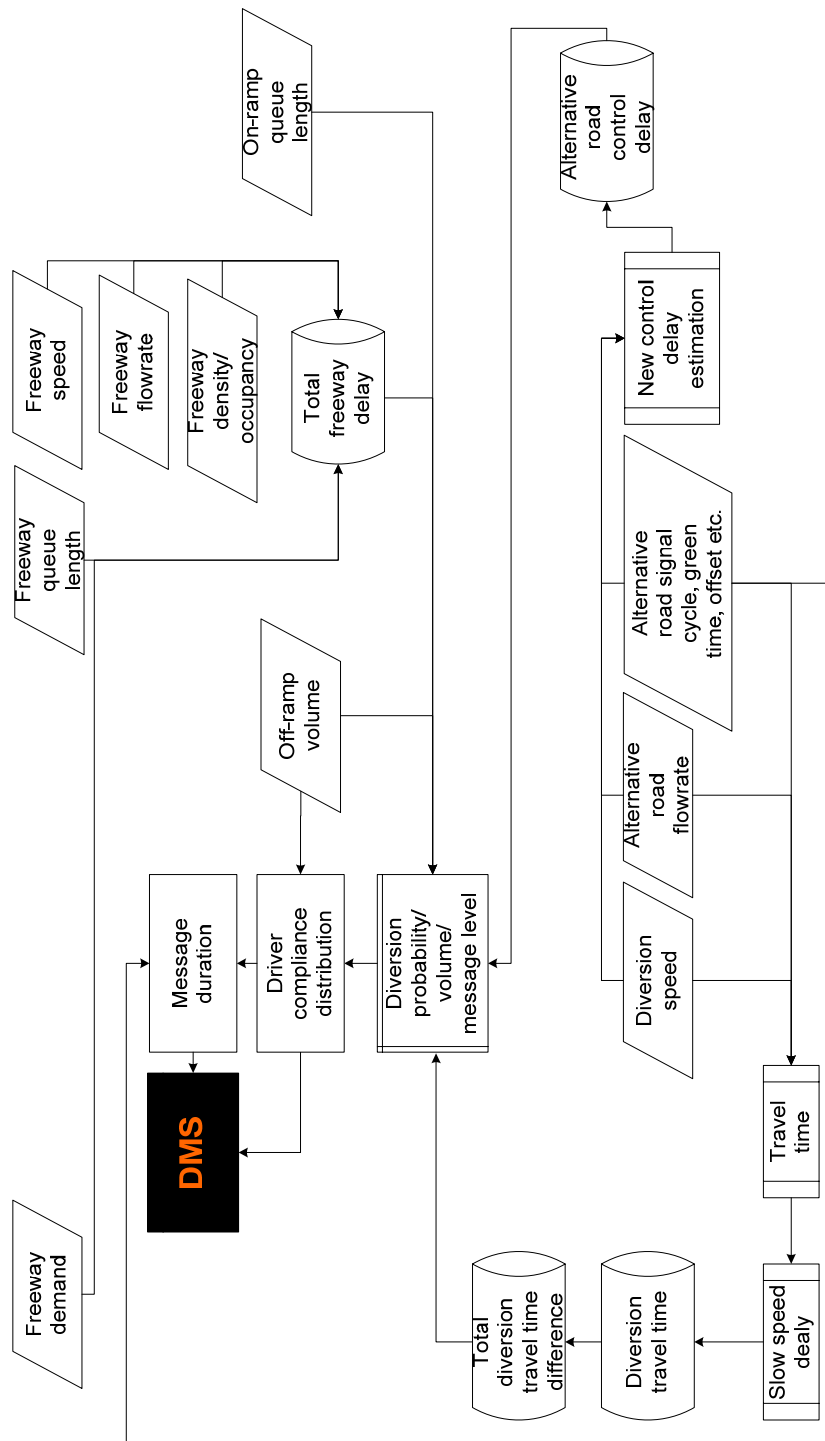


Figure 10 System Data Flow

Techniques Employed in the Implementation

Rolling Horizon

For the intersection signal-timing plan that will be updated periodically, a technique called rolling horizon (53) can help to make the signal control and more efficient updates. One reason for the concept development is that the rolling horizon replaces the future data needs with the current available data. This overcomes long-term prediction inaccuracy and makes the optimization more practical for a real-time system. Current traffic arrival data can be obtained from upstream detectors and utilized for short-term prediction for the next optimization interval.

Rolling horizon updates the signal at a certain interval (roll period) after the optimal timing plan is drawn. It means that an optimal timing plan is implemented only for the roll period and then it will be replaced by the next one. This concept is illustrated in Figure 11. Optimization period is divided into small subintervals. The optimal plan is originally drawn for the whole horizon, and will only be executed for one to multiple subintervals. Once a new optimal timing plan is generated, the existing plan will be replaced with the new one right away.

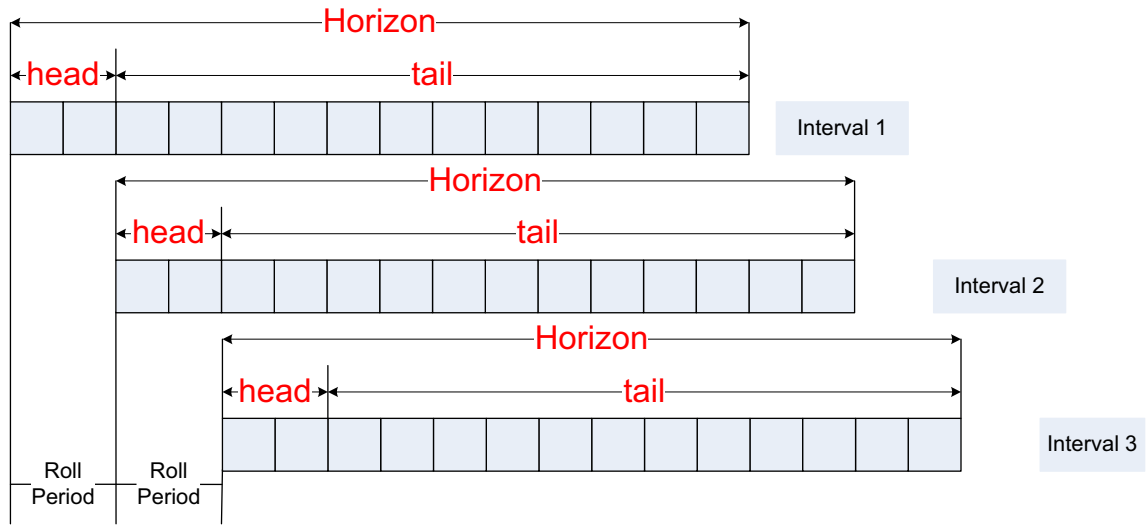


Figure 11 Illustration of Rolling Horizon Concept

In the proposed DMS control system, there are two roll periods. One is for the alternative road signal update and the other one is for the DMS message update. The reason to set up two roll periods is that the traffic signal update period should not be too long and the DMS update period should not be too short. As a result, it is better to set up a separate roll period for each signal and the DMS update. Figure 12 shows the principle of two roll periods system. The roll period of DMS is one level higher than that of the signal roll period. There will be several signal roll periods within a DMS roll period.

The horizon for this proposed DMS system should be longer than the diversion travel time. The signal roll period could be one to two minutes depending on the real-time execution speed of the algorithm (two minutes is found to be the acceptable time from numerical example study). The DMS roll period is recommended to be longer than ten to fifteen minutes on freeway for the high-speed traffic.

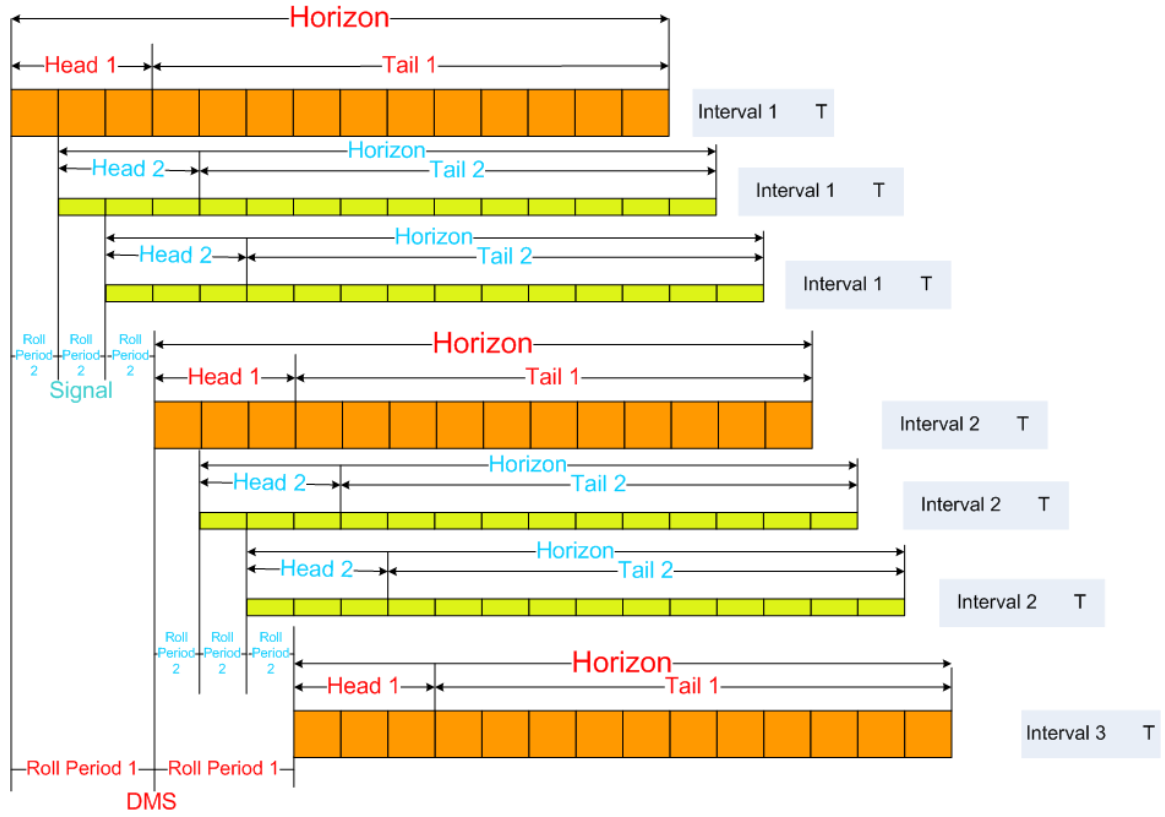


Figure 12 Illustration of DMS and Signal Rolling Periods

Penalty Factor

In order to provide more physically meaningful information beyond the algorithm constraints, penalty factors are considered to guide the model solution search to avoid some obviously infeasible solution areas. When the freeway has a queue or the intersection has the V/C ratio greater than 1.0, a penalty of certain veh-hours will be imposed into the objective function value to avoid that direction. The revised objective function with the penalty factors is shown as

$$\text{Min}(D_{Fa} + D_D + D_I + D_P) = \text{Min}(D_{Fa} + D_D + \sum_{i=1}^n D_i + \sum_{j=1}^G \sum_{p=1}^3 \delta_j D_{pj}) \quad (54)$$

$$\delta_1 = \begin{cases} 1, & \text{If queue on freeway} \\ 0, & \text{Otherwise} \end{cases},$$

$$\delta_2 = \begin{cases} 1, & \text{If } X_p > 1 \\ 0, & \text{Otherwise} \end{cases},$$

$$\delta_3 = \begin{cases} 1, & \text{If any } X > 1 \\ 0, & \text{Otherwise} \end{cases}.$$

Where X is the V/C ratio and X_p is the V/C ratio of the lane groups on the diversion path. D_{pj} is the delay penalty in veh-hours. Since it is possible that both queue on freeway and V/C ratio is greater than one appears simultaneously, the penalty factor value for the above three situation should be refined to give elaborated search guidance. For example, penalty factor will have 1000 veh-hours increase when queue appears on freeway, 500 veh-hours increase when X_p is greater than 1 and 250 veh-hours increase for when X is greater than 1. The penalty factor values are set case by case based on the total delay estimation. The effectiveness of the penalty factor is discussed later in Chapter Five.

Analytical Difference Equations Utilization in SQP Model Solution

SQP algorithm adopted in the *fmincon* solver in MATLAB's optimization toolbox utilizes finite difference values for gradient by default. However, MATLAB also provides option to let users provide the analytical difference equations to calculate gradients. Theoretically speaking, the analytical gradient calculation will be more accurate and efficient than that from the finite difference gradient. Therefore, the analytical gradient calculation is implemented for improving efficiency. In this section, analytical difference equations for all the corridor delay components are deducted.

For convenience, to deduce the analytical difference the objective function is reformatted and illustrated in the following Equations (55) with independent variables.

$$\begin{aligned}
D = & \sum_{i=3}^{NVAR} (d_{1i} + d_{2i})v_i + \\
& \left(\left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fma}} \right] \right) * \\
& (v_1 - x_1) - \frac{\sum_{j=0}^m DSF_{j,j+1}}{v_{Ff}} * (v_1 - x_1) + (T_D - d_{qF}) * x_1 - \\
& \left(\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fma}} \right) * x_1
\end{aligned} \tag{55}$$

Where:

$NVAR$, total number of independent variables,

v_l , total demand on freeway,

x_l , diversion volume.

Other denotations are the same as in Chapter Three.

Total Control Delay

There are two components of control delay: d_l , uniform control delay, and d_2 , incremental delay.

The total control delay is the summation of product of lane group unit control delays and corresponding volumes. The diversion volume will be superimposed onto the lane groups that are on the diversion path.

Therefore, diversion volume variables will be part of the delay equation of lane groups on the diversion path, while the lane groups that are not on the diversion path do not have the diversion volume variable included. As a result, the equations are separated into two groups: on diversion path and NOT on diversion path.

$$d_{1i} = \frac{0.50x_2 \left(1 - \frac{x_i}{x_2}\right)^2}{1 - \left[\min(1, X_i) \frac{x_i}{x_2}\right]} = \begin{cases} 0.5(x_2 - x_i) & X_i \geq 1 \\ \frac{0.5s_i(x_2 - x_i)^2}{(x_2(s_i - v_i - x_1))} & \text{on diversion path} \\ \frac{0.5s_i(x_2 - x_i)^2}{(x_2(s_i - v_i))} & \text{NOT on diversion path} \end{cases} \quad X_i < 1 \quad (56)$$

$i = 3, 4, \dots, NVAR$

$$\frac{\partial d_{1i}}{\partial x_1} = \begin{cases} \begin{cases} 0 & \text{group } i \text{ is on diversion path} \\ 0 & \text{group } i \text{ is NOT on diversion path} \end{cases} & X_i \geq 1 \\ \begin{cases} \frac{0.5s_i(x_2 - x_i)^2}{x_2(s_i - v_i - x_1)^2} & \text{group } i \text{ is on diversion path} \\ 0 & \text{group } i \text{ is NOT on diversion path} \end{cases} & X_i < 1 \end{cases} \quad (57)$$

$i = 3, 4, \dots, NVAR$

$$\frac{\partial d_{1i}}{\partial x_2} = \begin{cases} \begin{cases} 0.5 & \text{group } i \text{ is on diversion path} \\ 0.5 & \text{group } i \text{ is NOT on diversion path} \end{cases} & X_i \geq 1 \\ \begin{cases} \frac{0.5s_i}{(s_i - v_i - x_1)} \left(1 - \left(\frac{x_i}{x_2}\right)^2\right) & \text{group } i \text{ is on diversion path} \\ \frac{0.5s_i}{(s_i - v_i)} \left(1 - \left(\frac{x_i}{x_2}\right)^2\right) & \text{group } i \text{ is NOT on diversion path} \end{cases} & X_i < 1 \end{cases} \quad (58)$$

$i = 3, 4, \dots, NVAR$

$$\frac{\partial d_{1i}}{\partial x_j} = \begin{cases} \begin{cases} -0.5 & \text{group } i \text{ is on diversion path} \\ -0.5 & \text{group } i \text{ is NOT on diversion path} \end{cases} & X_i \geq 1 \\ \begin{cases} \begin{cases} -\frac{s_i(x_2 - x_i)}{x_2(s_i - v_i - x_1)} & i = j \\ 0 & i \neq j \end{cases} & \text{group } i \text{ is on diversion path} \\ \begin{cases} -\frac{s_i(x_2 - x_i)}{x_2(s_i - v_i)} & i = j \\ 0 & i \neq j \end{cases} & \text{group } i \text{ is NOT on diversion path} \end{cases} & X_i < 1 \end{cases} \quad (59)$$

$$i, j = 3, 4, \dots, NVAR$$

$$d_{2i} = 900T \left[(X_i - 1) + \sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}} \right] \quad i = 3, 4, \dots, NVAR \quad (60)$$

$$\frac{\partial d_{2i}}{\partial x_j} = \left(900T + \frac{450T}{\sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}} * \left(2(X_i - 1) + \frac{8kI}{c_i T} \right) * \frac{\partial X_i}{\partial x_j} \right) \quad (61)$$

$$i = 3, 4, \dots, NVAR \quad j = 1, 2, \dots, NVAR$$

$$\frac{\partial X_i}{\partial x_1} = \begin{cases} \frac{\partial \left(\frac{(v_i + x_1)x_2}{s_i x_i} \right)}{\partial x_1} = \frac{x_2}{s_i x_i} & \text{group } i \text{ is on diversion path} \\ \frac{\partial \left(\frac{v_i x_2}{s_i x_i} \right)}{\partial x_1} = 0 & \text{group } i \text{ is NOT on diversion path} \end{cases} \quad (62)$$

$$i = 3, 4, \dots, NVAR$$

$$\frac{\partial X_i}{\partial x_2} = \begin{cases} \frac{\partial \left(\frac{(v_i + x_1)x_2}{s_i x_i} \right)}{\partial x_2} = \frac{(v_i + x_1)}{s_i x_i} & \text{group } i \text{ is on diversion path} \\ \frac{\partial \left(\frac{v_i x_2}{s_i x_i} \right)}{\partial x_2} = \frac{v_i}{s_i x_i} & \text{group } i \text{ is NOT on diversion path} \end{cases} \quad (63)$$

$i = 3, 4, \dots, NVAR$

$$\frac{\partial X_i}{\partial x_j} = \begin{cases} \begin{cases} \frac{\partial \left(\frac{(v_i + x_1)x_2}{s_i x_i} \right)}{\partial x_j} = -\frac{(v_i + x_1)x_2}{s_i x_i^2} & i = j \quad \text{group } i \text{ on diversion path} \\ 0 & i \neq j \end{cases} \\ \begin{cases} \frac{\partial \left(\frac{v_i x_2}{s_i x_i} \right)}{\partial x_j} = -\frac{v_i x_2}{s_i x_i^2} & i = j \quad \text{group } i \text{ is NOT on diversion path} \\ 0 & i \neq j \end{cases} \end{cases} \quad (64)$$

$i, j = 3, 4, \dots, NVAR$

When group i is on diversion path, the total delay of that group from all vehicles is shown in the following equations. Where the D_1 is the total delay of uniform delay and the D_2 is the total delay of incremental delay.

$$D_1 = \sum (d_{1i} * (v_i + x_1)) \quad i \in \text{groups on diversion path} \quad (65)$$

$$D_2 = \sum (d_{2i} * (v_i + x_1)) \quad i \in \text{groups on diversion path} \quad (66)$$

The first partial derivative D_{1i} and D_{2i} of x_1 are different from d_{1i} and d_{2i} since x_1 appears in the equation, and the rest first partial derivatives will remain the same as that from d_1 and d_2 .

$$\frac{\partial D_1}{\partial x_1} = \sum \left(\frac{\partial d_{1i}}{\partial x_1} * (v_i + x_1) + d_{1i} \right) \quad i \in \text{groups on diversion path} \quad (67)$$

$$\frac{\partial D_2}{\partial x_1} = \sum \left(\frac{\partial d_{2i}}{\partial x_1} * (v_i + x_1) + d_{2i} \right) \quad i \in \text{groups on diversion path} \quad (68)$$

$$\frac{\partial D_1}{\partial x_j} = \sum \left(\frac{\partial d_{1i}}{\partial x_j} * v_i \right) \quad i \in \text{groups on diversion path} \quad j = 2, 3, \dots, NVAR \quad (69)$$

$$\frac{\partial D_2}{\partial x_j} = \sum \left(\frac{\partial d_{2i}}{\partial x_j} * v_i \right) \quad i \in \text{groups on diversion path} \quad j = 2, 3, \dots, NVAR \quad (70)$$

When group i is NOT on diversion path, the total delays and their first derivatives of that group from all vehicles are shown in the following equations. The equation difference between the on diversion path and the NOT on diversion path is whether the diversion volume variable is included or not.

$$D_1 = \sum (d_{1i} * v_i) \quad i \in \text{groups NOT on diversion path} \quad (71)$$

$$D_2 = \sum (d_{2i} * v_i) \quad i \in \text{groups NOT on diversion path} \quad (72)$$

$$\frac{\partial D_1}{\partial x_j} = \sum \left(\frac{\partial d_{1i}}{\partial x_j} * v_i \right) \quad i \in \text{groups NOT on diversion path} \quad j = 1, 2, 3, \dots, NVAR \quad (73)$$

$$\frac{\partial D_2}{\partial x_j} = \sum \left(\frac{\partial d_{2i}}{\partial x_j} * v_i \right) \quad i \in \text{groups NOT on diversion path} \quad j = 1, 2, 3, \dots, NVAR \quad (74)$$

Group volumes on diversion path will include the diversion volume. Their first partial derivative equations are different from the rest of the groups (for example, groups numbered 3, 6, 19, and 26 are on diversion path in the numerical example network). The aforementioned deducted equations are reorganized based on cases as the followings.

Partial derivatives of unit uniform delay d_i :

For $i=3$ to $NVAR$

If $X_i \geq 1$

$$\frac{\partial d_{li}}{\partial x_1} = 0$$

$$\frac{\partial d_{li}}{\partial x_2} = 0.5$$

$$\frac{\partial d_{li}}{\partial x_j} = -0.5 \quad i = j \quad i, j = 3, 4, \dots, NVAR$$

$$\frac{\partial d_{li}}{\partial x_j} = 0 \quad i \neq j \quad i, j = 3, 4, \dots, NVAR$$

Else

If it is an on path lane group, (for example, $i = \{3, 6, 19, \text{ and } 26\}$ in the numerical example)

$$\frac{\partial d_{li}}{\partial x_1} = \frac{0.5s_i(x_2 - x_i)^2}{x_2(s_i - v_i - x_1)^2}$$

$$\frac{\partial d_{li}}{\partial x_2} = \frac{0.5s_i}{(s_i - v_i - x_1)} \left(1 - \left(\frac{x_i}{x_2} \right)^2 \right)$$

$$\frac{\partial d_{li}}{\partial x_j} = -\frac{s_i(x_2 - x_i)}{x_2(s_i - v_i - x_1)} \quad i = j \quad i, j = 3, 4, \dots, NVAR$$

$$\frac{\partial d_{li}}{\partial x_j} = 0 \quad i \neq j \quad i, j = 3, 4, \dots, NVAR$$

End if

If it is NOT a on-diversion path lane group ($i! = \{3, 6, 19, 26\}$ in the numerical example)

$$\frac{\partial d_{li}}{\partial x_1} = 0$$

$$\frac{\partial d_{li}}{\partial x_2} = \frac{0.5s_i}{(s_i - v_i)} \left(1 - \left(\frac{x_i}{x_2} \right)^2 \right)$$

$$\frac{\partial d_{li}}{\partial x_j} = -\frac{s_i(x_2 - x_i)}{x_2(s_i - v_i)} \quad i = j \quad i, j = 3, 4, \dots, NVAR$$

$$\frac{\partial d_{li}}{\partial x_j} = 0 \quad i \neq j \quad i, j = 3, 4, \dots, NVAR$$

End if

End if

End for

Partial derivatives of unit incremental delay d_2 :

For =3 to $NVAR$

If it is an on diversion path lane group, for example, $i=\{3, 6, 19, \text{ and } 26\}$

$$\frac{\partial d_{2i}}{\partial x_1} = \left(900T + \frac{450T}{\sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}} * \left(2(X_i - 1) + \frac{8kI}{c_i T} \right) \right) * \frac{x_2}{s_i x_i}$$

$$\frac{\partial d_{2i}}{\partial x_2} = \left(900T + \frac{450T}{\sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}} * \left(2(X_i - 1) + \frac{8kI}{c_i T} \right) \right) * \frac{(v_i + x_1)}{s_i x_i}$$

$$\frac{\partial d_{2i}}{\partial x_j} = \left(900T + \frac{450T}{\sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}} * \left(2(X_i - 1) + \frac{8kI}{c_i T} \right) \right) * \left(-\frac{(v_i + x_1)x_2}{s_i x_i^2} \right)$$

$$i = j \quad i, j = 3, 4, \dots, NVAR$$

$$\frac{\partial d_{2i}}{\partial x_j} = 0 \quad i \neq j \quad i, j = 3, 4, \dots, NVAR$$

End if

If it is a on-path lane group for example, $i!=\{3, 6, 19, 26\}$

$$\frac{\partial d_{2i}}{\partial x_1} = 0$$

$$\frac{\partial d_{2i}}{\partial x_2} = \left(900T + \frac{450T}{\sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}} * \left(2(X_i - 1) + \frac{8kI}{c_i T} \right) \right) * \frac{v_i}{s_i x_i}$$

$$\frac{\partial d_{2i}}{\partial x_j} = \left(900T + \frac{450T}{\sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}} * \left(2(X_i - 1) + \frac{8kI}{c_i T} \right) * \left(-\frac{v_i x_2}{s_i x_i^2} \right) \right)$$

$$i = j \quad i, j = 3, 4, \dots, NVAR$$

$$\frac{\partial d_{2i}}{\partial x_j} \quad i \neq j \quad i, j = 3, 4, \dots, NVAR$$

End if

End for

The above equations are for unit vehicle delays, the total traffic control delay from all vehicles in a lane group is a part of the total delay in the objective function. Therefore, the first derivative of total control delay is eventually needed. However, only first partial derivative of x_l is different from the first partial derivative of unit signal control delay.

$$\frac{\partial(Grad_Tot_d_1)}{\partial x_1} = \sum_{i=3}^{NVAR} \left(\frac{\partial d_{1i}}{\partial x_1} * (v_i + x_1) + \begin{cases} 0.5(x_2 - x_i) & X_i \geq 1 \Rightarrow v_i \geq s_i \\ \frac{0.5s_i(x_2 - x_i)^2}{(x_2(s_i - v_i - x_1))} & X_i < 1 \Rightarrow v_i < s_i \end{cases} \right); \quad (75)$$

$$\frac{\partial(Grad_Tot_d_2)}{\partial x_1} = \sum_{i=3}^{NVAR} \left(\frac{\partial d_{2i}}{\partial x_1} * (v_i + x_1) + 900T \left[(X_i - 1) + \sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}} \right] \right) \quad (76)$$

For j=2 to NVAR

$$\frac{\partial(Grad_Tot_d_1)}{\partial x_j} = \sum_{i=3}^{NVAR} \left(\frac{\partial d_{1i}}{\partial x_j} * (v_i) \right); \quad (77)$$

$$\frac{\partial(Grad_Tot_d_2)}{\partial x_j} = \sum_{i=3}^{NVAR} \left(\frac{\partial d_{2i}}{\partial x_j} * (v_i) \right). \quad (78)$$

End for

Total Freeway Travel Delay

Total freeway travel delay is a function of diversion volume only; thus, the first derivative of x_1

gets non-zero value, and the rest first derivatives of x_2 to x_{NVAR} are all zeros.

$$D_{Fa} = \left(\left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fma}} \right] \right. \\ \left. * (v_1 - x_1) - \frac{\sum_{j=0}^m DSF_{j,j+1}}{v_{Ff}} * (v_1 - x_1) \right) \quad (79)$$

$$\frac{\partial D_{Fa}}{\partial x_1} = - \left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fma}} \right] \\ + \frac{\sum_{j=0}^m DSF_{j,j+1}}{v_{Ff}} \quad (80)$$

For $j=2$ to $NVAR$

$$\frac{\partial D_{Fa}}{\partial x_j} = 0 \quad (81)$$

End for

Total and Average Freeway Queue Delay

In this section, D denotes the total Queue delay and d represents average queue delay. F_a represents the arrival flow and F_c represents the freeway capacity. T is the studied period. For the freeway queue delay, there are seven cases (case A to case F) to be considered under different initial queue (Q_0), freeway demand (F_d) and capacity (F_c) combinations.

If $F_a < F_c$

If initial Queue ≤ 0 (No initial queue Q_0 , no queue at the end of period T)

Case A:

$$D = \frac{F_a^2}{F_c(F_c - F_a)} = \frac{(v_1 - x_1)^2}{F_c^2 - F_c v_1 + F_c x_1} \quad (82)$$

$$\begin{aligned} \frac{\partial D}{\partial x_1} &= \frac{-2(v_1 - x_1)(F_c^2 - F_c v_1 + F_c x_1) - F_c(v_1 - x_1)^2}{(F_c^2 - F_c v_1 + F_c x_1)^2} \\ &= \frac{-2F_c(v_1 - x_1) + (v_1 - x_1)^2}{F_c(F_c - v_1 + x_1)^2} \end{aligned} \quad (83)$$

$$d = \frac{F_a}{TF_c(F_c - F_a)} = \frac{(v_1 - x_1)}{T(F_c^2 - F_c v_1 + F_c x_1)} \quad (84)$$

$$\frac{\partial d}{\partial x_1} = -\frac{F_c}{T(F_c^2 - F_c v_1 + F_c x_1)^2} \quad (85)$$

Else

If (initial queue + F_a) $> F_c$ (initial queue, end queue exist, end queue $<$ initial queue, and $F_a - F_c < 0$)

Case B:

$$D = \frac{(Q_0 + T(F_a - F_c))^2}{2F_c T} = \frac{(Q_0 + T(v_1 - x_1 - F_c))^2}{2F_c T} \quad (86)$$

$$\frac{\partial D}{\partial x_1} = -\left(\frac{Q_0 + T(v_1 - x_1 - F_c)}{F_c} \right) \quad (87)$$

$$d = \frac{Q_0 + (F_a - F_c)T}{F_c T} = \frac{Q_0 + V_1 T - F_c T - x_1 T}{F_c T} \quad (88)$$

$$\frac{\partial d}{\partial x_1} = -\frac{1}{F_c} \quad (89)$$

Else (initial queue, no queue at the end of period T)

Case C:

$$D = \frac{Q_0^2}{2F_c T} \quad (90)$$

$$\frac{\partial D}{\partial x_1} = 0 \quad (91)$$

$$d = \frac{Q_0}{2F_c T} \quad (92)$$

$$\frac{\partial d}{\partial x_1} = 0 \quad (93)$$

End if

End if

Else

If initial queue ≤ 0 (no initial queue of period T)

If $F_a - F_c \leq F_c$

Case D:

$$D = 0.5(T(F_a - F_c))^2 = 0.5T^2(v_1 - x_1 - F_c)^2 \quad (94)$$

$$\frac{\partial D}{\partial x_1} = -T^2(v_1 - x_1 - F_c) \quad (95)$$

$$d = 0.5T(F_a - F_c) = 0.5T(v_1 - x_1 - F_c) \quad (96)$$

$$\frac{\partial d}{\partial x_1} = -0.5T \quad (97)$$

Else $F_a - F_c > F_c$

Case E:

$$D = \frac{T^2(F_a - F_c)^2}{2F_c T} = \frac{T(v_1 - x_1 - F_c)^2}{2F_c} \quad (98)$$

$$\frac{\partial D}{\partial x_1} = -\frac{T}{F_c}(v_1 - x_1 - F_c) \quad (99)$$

$$d = \frac{(T(F_a - F_c))^2}{2F_c T^2(F_a - F_c)} = \frac{(F_a - F_c)}{2F_c} = \frac{(v_1 - x_1 - F_c)}{2F_c} \quad (100)$$

$$\frac{\partial d}{\partial x_1} = -\frac{1}{2F_c} \quad (101)$$

End if

Else (initial queue exists)

Case F

Delay part 1:

$$D = \frac{T^2(F_a - F_c)^2}{2} = \frac{T^2(v_1 - x_1 - F_c)^2}{2} \quad (102)$$

$$\frac{\partial D}{\partial x_1} = -T^2(F_a - F_c) = -T^2(v_1 - x_1 - F_c) \quad (103)$$

$$d = \frac{T^2(F_a - F_c)^2}{2(Q_0 + T(F_a - F_c))} = \frac{T^2(v_1 - x_1 - F_c)^2}{2(Q_0 + T(v_1 - x_1 - F_c))} \quad (104)$$

$$\begin{aligned} \frac{\partial d}{\partial x_1} &= \frac{-2T^2(v_1 - x_1 - F_c)(Q_0 + T(v_1 - x_1 - F_c)) + T^3(v_1 - x_1 - F_c)^2}{2((Q_0 + T(v_1 - x_1 - F_c)))^2} \\ &= -\frac{2T^2 Q_0(v_1 - x_1 - F_c) + T^3(v_1 - x_1 - F_c)^2}{2(Q_0 + T(v_1 - x_1 - F_c))^2} \end{aligned} \quad (105)$$

Delay part 2

$$D = \frac{(Q_0 + T(F_a - F_c))^2}{2F_c T} = \frac{(Q_0 + T(v_1 - x_1 - F_c))^2}{2F_c T} \quad (106)$$

$$\frac{\partial D}{\partial x_1} = -\frac{Q_0 + T(F_a - F_c)}{F_c} = -\frac{Q_0 + T(v_1 - x_1 - F_c)}{F_c} \quad (107)$$

$$d = \frac{(Q_0 + T(F_a - F_c))^2}{2F_c T(Q_0 + T(F_a - F_c))} = \frac{Q_0 + T(v_1 - x_1 - F_c)}{2F_c T} \quad (108)$$

$$\frac{\partial d}{\partial x_1} = -\frac{1}{2F_c} \quad (109)$$

Total delay of case F:

$$\frac{\partial D}{\partial x_1} = -\left(T^2(v_1 - x_1 - F_c) + \frac{Q_0 + T(v_1 - x_1 - F_c)}{F_c} \right) \quad (110)$$

$$\frac{\partial d}{\partial x_1} = -\left(\frac{T^2 Q_0 (v_1 - x_1 - F_c)}{2((Q_0 + T(v_1 - x_1 - F_c)))^2} + \frac{1}{2F_c} \right) \quad (111)$$

End if

End if

First derivatives of x_2 to x_{nvar} of the freeway delay are zeros.

For $pd=2$ to $NVAR$

$$\frac{\partial D}{\partial x_j} = 0 \quad j = 2, 3, \dots, NVAR \quad (112)$$

$$\frac{\partial d}{\partial x_j} = 0 \quad j = 2, 3, \dots, NVAR \quad (113)$$

End

Detour Delay

Average queue delay of freeway is included in this part while the total queue delay is included in

total freeway delay part.

$$D_D = [T_D - (T_{Fa} + d_{qF})] * x_1 \quad (114)$$

$$\frac{\partial D_D}{\partial x_1} = T_D - T_{Fa} - d_{qF} - \frac{\partial d_{qF}}{\partial x_1} x_1 \quad (115)$$

$$\frac{\partial D_D}{\partial x_1} = T_D - d_{qF} - \frac{\partial d_{qF}}{\partial x_1} x_1 - \left[\frac{DSF_{0,1} + \frac{DSF_{1,2}}{2}}{v_{F1a}} + \sum_{j=2}^{m-1} \left[\frac{\frac{DSF_{j-1,j}}{2} + \frac{DSF_{j,j+1}}{2}}{v_{Fja}} \right] + \frac{DSF_{m,m+1} + \frac{DSF_{m-1,m}}{2}}{v_{Fma}} \right] \quad (116)$$

$$\frac{\partial D_D}{\partial x_j} = 0 \quad j = 2, 3 \dots NVAR \quad (117)$$

Pseudo Code of System Implementation

Pseudo codes of the whole system implementation are listed in Appendix A. The following sections mainly discuss about CORSIM RTE configuration for the system implementation, the RTE coding in Visual Studio 2005, and the model coding in MATLAB.

CORSIM Call Points for RTE

After the C++ RTE codes are compiled, a Dynamic Link Library (DLL) is generated. Functions inside the DLL are ready for CORSIM call at certain internal time points along the CORSIM execution time line.

The internal call points in the CORSIM that CORSIM will call the RTE functions are described briefly in the following paragraphs. There is a call point configuration example from case study in the Chapter Five.

There are fifteen call points revealed by CORSIM Run-Time Extension (RTE) Developer's Guide (123). The relationships among these call points are organized and illustrated in Figure 13. For the proposed system, about six out of the fifteen call points are actually called during this dissertation study.

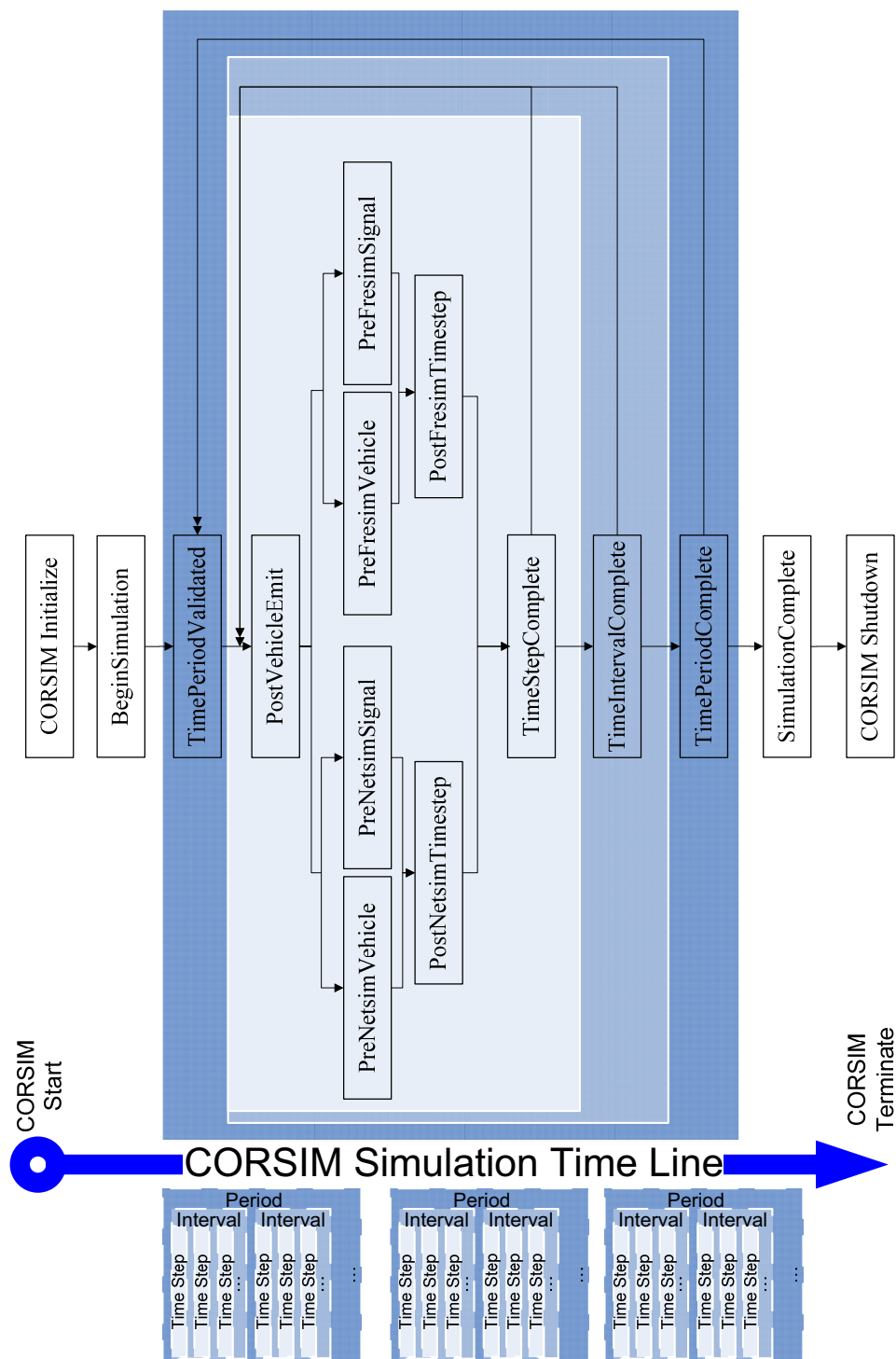


Figure 13 Structure Illustration of CORSIM Call Points

When CORSIM starts, the first task is to read its input (TRF) file (Figure 13). Then, CORSIM will initialize the system (network fill). When the initialization is done, the network usually reaches the state of equilibrium that the difference between vehicle in and vehicle out of the network should be within a tolerable range.

CORSIM simulation follows the time step, time interval and time period hierarchy. CORSIM will validate each time period data before the start of simulation for each period. After CORSIM gets into the simulation stage, it follows a time step by time step simulation procedure. At each time step, several call points are available. Because FRESIM and NETSIM are two distinct sub-networks in CORSIM, different treatments for the vehicle and signal in those two sub-networks were developed. As a result, two sets of call points before vehicle and signal are set for the two sub-networks. The two sets of call points for FRESIM and NETSIM are both called after the first task of a step - emitting vehicles into the network.

After the simulation is completed and before the CORSIM program exits, two more call points are provided. The first one is called at the end of the simulation and the other one is called just before the shutdown of CORSIM. It needs to be pointed out that during the called function execution time, CORSIM simulation is paused and waiting for the return from the function execution.

The six call points were utilized for this proposed system are “Initialize,” “PreNETSIMSignal,” “PreFRESIMVehicle,” “BeginSimulation,” “TimeStepComplete,” and “Exit”. At “Initialize” call point, user configure file for the studied case is read in and the network geometric data will be output by RTE later. At each time step, there is an RTE function monitoring the time line to check if there is an NETSIM signal needs to be updated or new generated FRESIM vehicles needs to be assigned onto the diversion path.

Every vehicle will face a choice of diversion following the diversion probability from a latest message on a DMS board. If a vehicle chooses the diversion route, an RTE path assign function is called at the “PreFRESIMVehicle” to change the vehicle’s path parameter. When an RTE exits, there is one more call point “Exit” from CORSIM by default. At “Exit” call point, the RTE function *write_summary* is called to output the entire system execution summary.

RTE Coding in Visual Studio 2005

As illustrated in Figure 8, a mainframe program is developed to start the TSIS and CORSIM automatically. Once CORSIM starts, the user can activate the RTE and run simulation manually during the proof-of-concept running. CORSIM provides command line execution and script call that enable users to start CORSIM and RTE automatically (this will be the future improvement of this system to reduce whole system execution time). During CORSIM simulation, the proposed optimization model is called by CORSIM RTE and solved in MATLAB repeatedly following the user configuration in the user configure file.

The RTE is structured with several functions to exchange data with MATLAB, to call MATLAB, to read in solutions from a MATLAB data file, to assign diversion path to certain vehicles and to change the intersection signal-timing plan. In Figure 14, critical steps and related activities are shown in the flowchart. At “Initialize” stage of CORSIM, function *read_file* is called to read in user configuration file. Then, the CORSIM network geometric data and other time independent data is output by function *network*. The data output by *network* function are general geometric and simulation configuration of a network.

After the initialization stage, a function dealt with freeway detector initialization data is called just before the simulation starts. The reason to start this function and keep the detector data of freeway initialization period is that the freeway detector data is cumulated from the initialization period rather than from the start time of simulation. Freeway detector volume and speed of each optimization period need the initial data for extracting period data from cumulative data.

Once the simulation starts, several functions will be called at each time step (1-second by default). At the “TimeStepComplete,” RTE function will check if it is time for a DMS update period, DMS rolling period or signal rolling period. If it is one of the three time points, MATLAB is called to search for a solution. Before the MATLAB is called, the time dependent network status data such volume, speed and so on are dumped into a data file for passing to MATLAB.

When RTE finds the MATLAB is in the Windows process list, it means MATLAB is active and the CORSIM simulation is then forced to wait until MATLAB is completed. After MATLAB finishes the optimization, a data file is generated for CORSIM RTE to get the solution. A function *read_file_periodic* is in charge of reading in the MATLAB solution. CORSIM will resume running after process of reading in data.

At the same time, RTE also checks and executes the signal update and path assignment to all the new generated FRESIM vehicles before the next time step starts. Because the signal change only happens at the end of signal roll period, the new timing plan is streamed into a specific stream data structure along a time line. Only when the time clock reaches a signal changing time point, will the signal update actually start. Otherwise, RTE only checks and matches the time point on the signal update time line.

Functions developed in RTE and their brief descriptions are listed in the following paragraphs.

read_file: To read in the RTE running configurations, special nodes, links and such information for diversion simulation. The user can control the RTE running and algorithm used in the MATLAB optimization by configuring the input file that will be read in by this function. An example of the configure file is included in the Appendix B.

read_file_periodic: To read in the MATLAB returned solution after each MATLAB running.

write_file_periodic: To write data for MATLAB running before each MATLAB call. An example of the configure file is included in the Appendix C.

write_summary: After simulation is done, all the time dependent execution data is output by this function.

network: To output all the network-wide geometric data, such as node, link, and freeway detector geometric data and such time independent data.

dt_initial_vol: To take initial freeway detector values. Because freeway detector values are cumulated from initial period of CORSIM, values at clock time zero are necessary for later periodic computation.

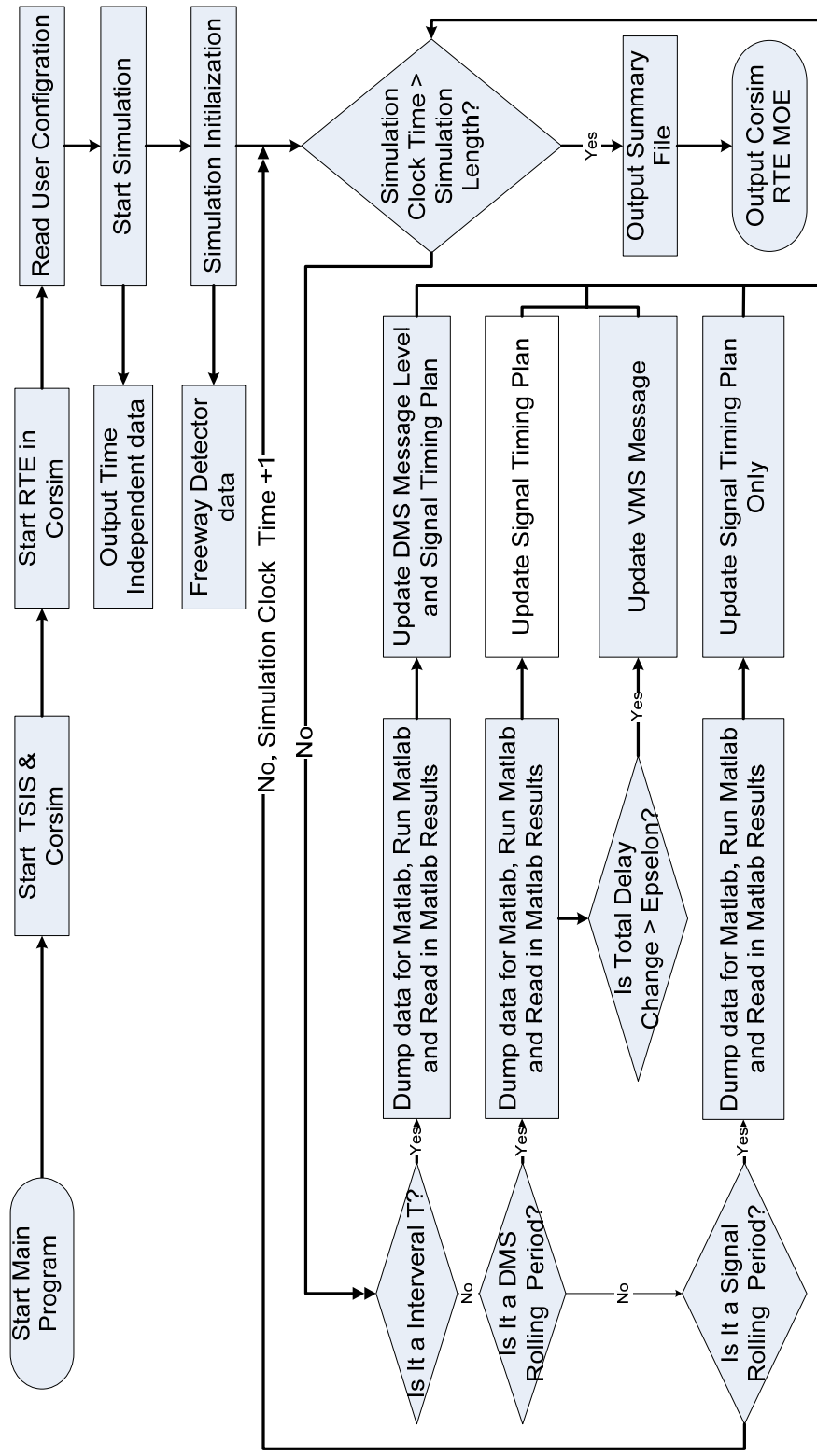


Figure 14 System Implementation Structure

call_MATLAB: To start MATLAB and execute optimization modules.

signal_change_at_offset: Because the offset for intersection on alternative road, the signal plans of different intersections should not be all changed at the same time. Since the offset cannot be greater than 256 from the CORSIM signal time variable limitation, it is necessary to start writing the less than 256 seconds offset time into CORSIM variable at certain integer times of 256 seconds (depending on the real offset value) by RTE.

veh_path_assign: In order to make sure that the diverted vehicles will get back onto freeway, the diverted vehicles must be assigned a path to follow, which will lead the diversion traffic back onto freeway in CORSIM.

veh_diversion_choice: Each vehicle has the MATLAB returned diversion probability (determined by DMS message level) to choose diversion before the diversion node. This function enumerates all the FRESIM vehicles to find those who are originally not a path following vehicles and will not take the off-ramp, then, generates an extreme value random number for path assignment decision of a vehicle.

Model Coding in MATLAB

In the MATLAB model coding, there are three different functions regarding different RTE call types. The first function is the main function of the whole model. It will read in the data from CORSIM RTE first. Then, it will estimate minimal cycle length based on given volume and saturation flow of all approaches. The estimated cycle length serves as lower bound of cycle length in model constraints.

Because the MATLAB solvers need object function for the optimal solution search, a sub-function of model objective function that completes all the model calculations is defined inside the MATLAB main function range.

Model objective function calculations mainly include freeway travel delay, signal control delay and the diversion travel time difference; among which, the freeway queue delay calculation is the most complicated one. It needs to estimate the queue length that cannot be fully monitored by the freeway detector station. There are different methods to calculate stochastic and standing queue delay depending on queue status. Since the model coded in MATLAB will be started over at each call, all the necessary system history data is kept in RTE data structure and will be passed to MATLAB by disk data file.

The second function is only part of the first function. It only deals with the signal optimization that is one variable less than the first function - the diversion volume variable, x_I . The reason for creating this second function is that the candidate algorithms cannot optimize the mixed integer and real variable types. As a result, the DMS level will not be the direct result from MATLAB module. Alternatively, the diversion volume as a real number is returned in the solution. The diversion volume is rounded to the closest predefined DMS level following the predefined mapping table (Table 1). This approximation causes change in the final diversion volume. Therefore, a new signal only optimization based on the new DMS level caused diversion volume is necessary to be conducted again. The second function just returns the new signal-timing plan to the first function and the first function will finish the data output (including the DMS level).

The third function is similar to the second function in that it only deals with signal optimization.

The difference is that the third function can run separately while the second function can only run within the first function. The reason is that only signal is optimized in RTE at the point of signal roll period and the second function is a dependent function of the first one. Therefore, the third function is modified based on the second function and it can communicate with RTE directly.

CHAPTER V

RESULT ANALYSIS AND SYSTEM EVALUATION

The proposed system evaluation is based on simulation case studies. CORSIM measures of effectiveness (MOE) from an example network, which is a small corridor network with one freeway and four intersections on one alternative road. This was used to evaluate the model, algorithms and system effectiveness.

The CORSIM MOEs adopted for evaluation are total delay of the network, and the network-wide average speed. Total delay and average speed of the network were compared between the base case and the after DMS control case. The estimated delay from objective function is not employed directly for evaluation because the CORSIM MOEs are unique and comparable under different scenarios.

Cases and Scenarios

A test network is designed to conduct the case studies. Evaluations of model, algorithms and system performances are all based on this network. For each intersection, CORSIM can only deal with twelve signal time intervals, which are equal to four phases (for pre-timed signals, it is up to 8 phases for actuated signals). Therefore, a maximum of four phases is designed in these case studies. At the intersection node 39 (intersection of off-ramp and arterial road), only two phases are set since it is a T

intersection. All the rest three intersections on alternative road have four phases. The following Figure 17 to Figure 20 demonstrate the studied intersections' geometry data, configurations and lane groups with independent variable numbers marked. At the same time, the equality constraints of green time in a same phase are also shown in those figures.

Six scenarios under different traffic conditions are designed in this evaluation study. Traffic conditions are classified into three levels: heavy, moderate and light congestion. Traffic demands over one direction of a four-lane freeway (two lanes in one direction) are set to 3500, 2900, and 2500 vph. The demand to capacity ratios are 1.7, 1.3 and 1.1, respectively (freeway capacity in CORSIM is assumed to be 2300 vph for queue delay estimation). The six scenarios are from two strategies based on the above three traffic conditions. The first strategy is to code the approximated diversion percentage from a pre-run model solution of the network and the second is to keep the original turning percentage of diversion node. The reason to put the diversion percentage into the diversion node is that the TRANSYT-7F cannot generate diversion volume in its solution, which will make the base case result less comparable with the proposed model. After merging the diversion percentage into the diversion node, it makes the CORSIM MOE from model solution and TRANSYT-7F solution base on a similar network state and thus is more comparable.

Because none of the three algorithms can guarantee global optimal solutions, it is possible that the solution varies for a same scenario under different CORSIM runs. The major reason is that CORSIM uses stochastic simulation model. If random seeds change, simulation will be different as well. Therefore, multiple runs are necessary for observing the statistical algorithm characteristics. The moderate scenario

with twenty percent diversion volume is chosen to run more than thirty-two times to observe the statistical characteristics of the three algorithms.

Besides observing model performance, algorithms introduced in Chapter Three also need to be evaluated under different circumstances. Therefore, six scenarios were run separately by SQP, GA, and PS algorithms. After system simulation and RTE calling for MATLAB model solution, CORSIM MOEs are utilized to compare the base network performance with the improved network performance. The simulation execution time is also compared between CORSIM and RTE model execution for model and algorithms' efficiency.

Example Network

The example network is shown in Figure 15 and Figure 16. Figure 15 illustrates the real network with traffic detector locations. Figure 16 illustrates the link and node codes in the CORSIM. Only the impacted freeway direction is coded. The oppose direction is not included because it is not affected by the work zone and it will not affect the work zone traffic as well.

The studied network is a corridor with one freeway and one parallel arterial road that has four intersections with pre-time signal control. The studied freeway segment length between diversion node and merging node is 7.37 miles while the entire length of the freeway segment is 14.36 miles long. Total length of alternative route is 11.89 miles (from diversion node to merging node). The free flow speed for freeway and alternative road are 65 and 30 mph respectively.

Freeway capacity decrease is caused by a 0.37-mile long full right lane closure work zone on the freeway. The work zone discharge rate is measured from CORSIM simulation to be around 2300 vph. Ten percent of trucks enter the network from freeway entry node and no trucks enter the NETSIM entry nodes. Capacities of the freeway and alternative roads are observed from CORSIM simulation of the example network.

There are ten detectors (stations) on freeway segment and ramps. Because there is only one lane on the off-ramp and on-ramp, there is one detector in the detector station of ramps. Otherwise, there is one detector on each lane of the freeway. In this example network, two detectors are in one freeway detector station. For the NETSIM links in CORSIM, the traffic status data can be obtained from the traffic signal detectors or from the NETSIM link property. Therefore, the alternative road detectors are not set up when CORSIM simulation serves as the data source and evaluation platform. However, if no other resources for the alternative road traffic state data are available in the real world, it is necessary to set up detectors to collect data. Data obtained from NETSIM link properties are similar or the same as the ones from traffic detectors.

The geometric configuration and lane groups of the four studied intersections are shown in the following Figure 17 to Figure 20. Numbers with circles are node number in the CORSIM network and arrows show the studied traffic lane groups. The x with number shows the independent variable number of the related lane group.

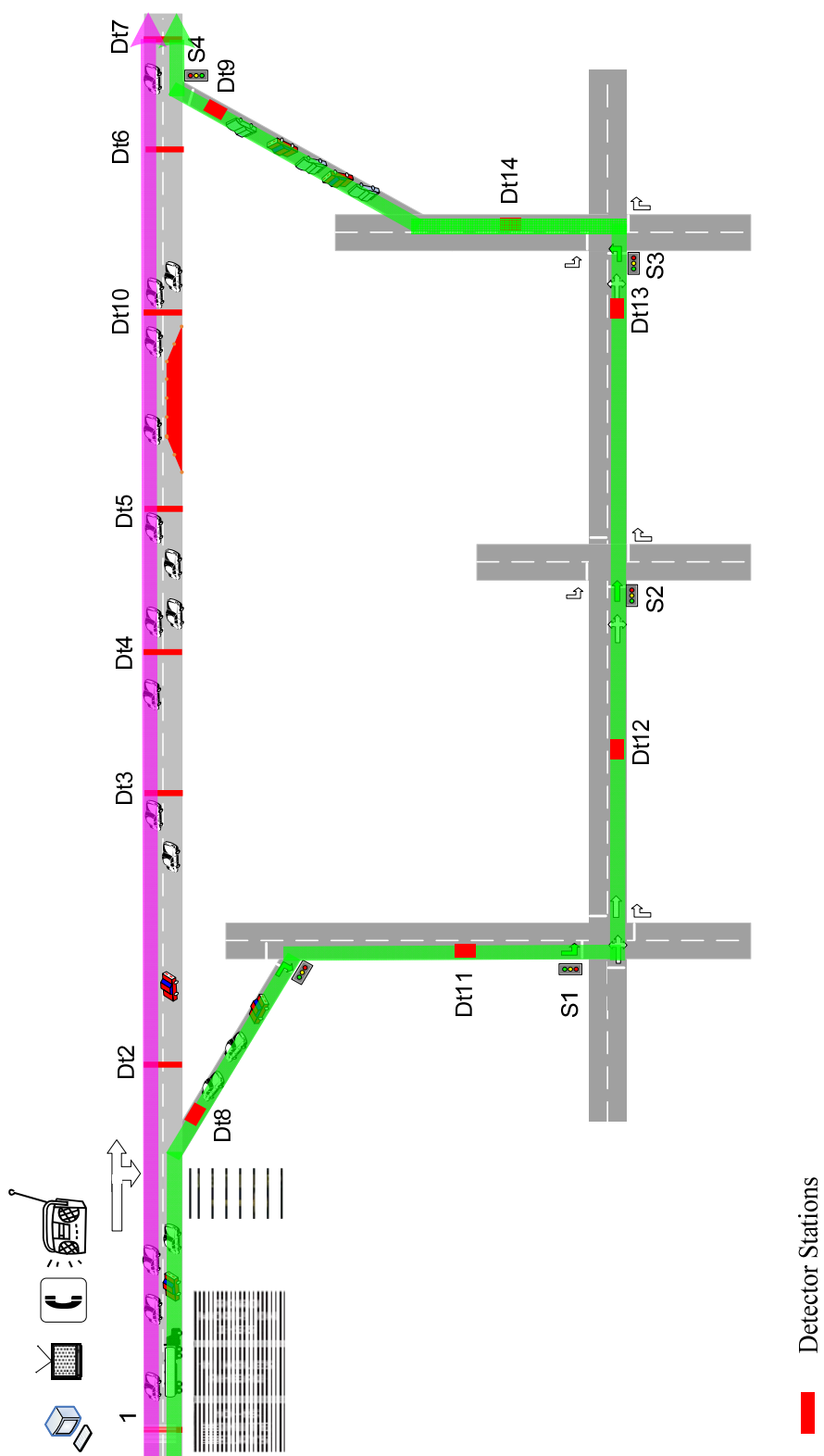


Figure 15 Example Network

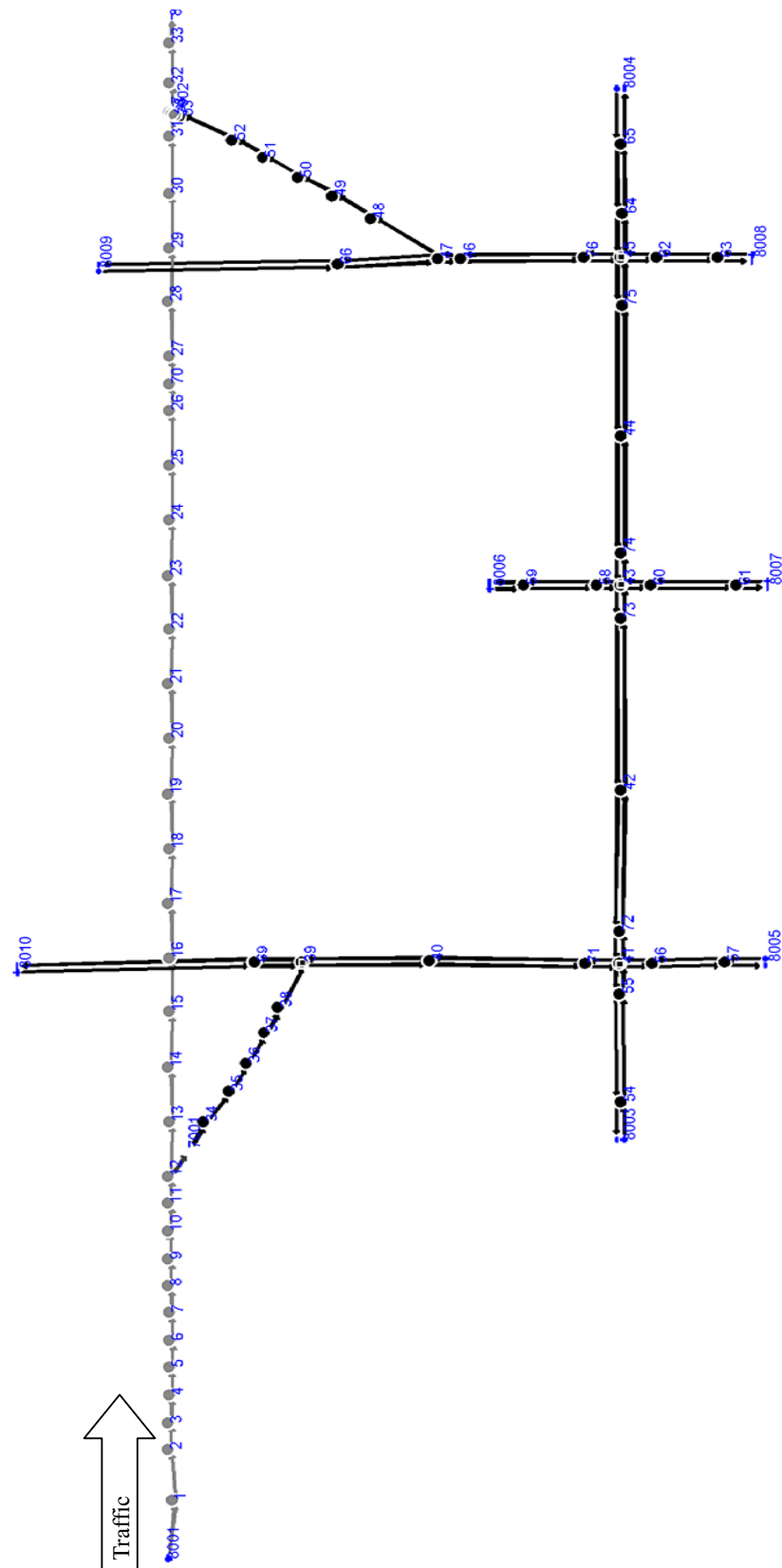


Figure 16 Example Network Code in CORSIM

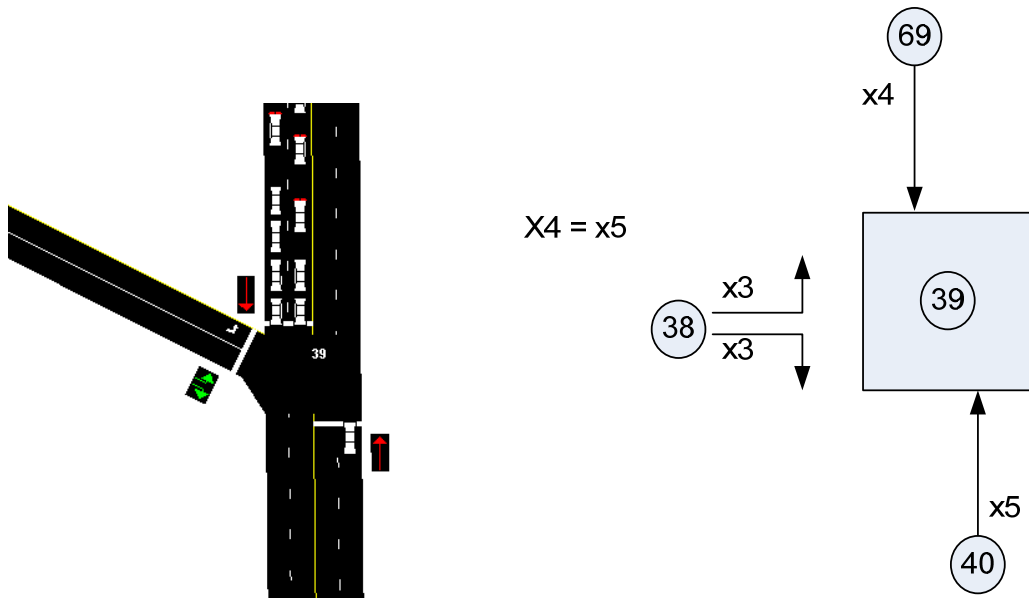


Figure 17 Geometric Information and Variable Number of Intersection Node 39

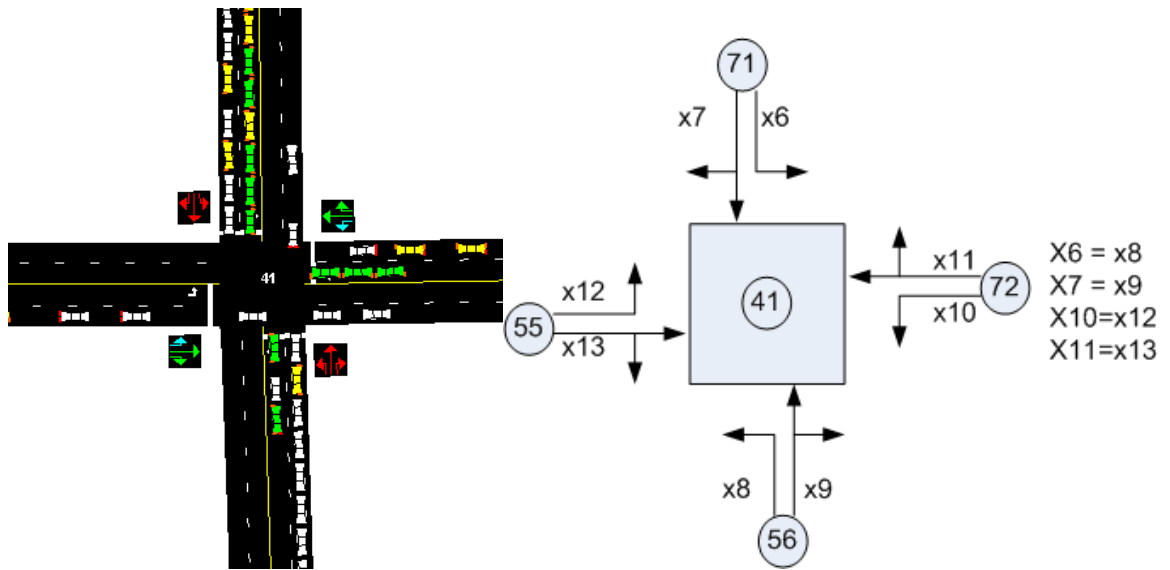


Figure 18 Geometric Information and Variable Number of Intersection Node 41

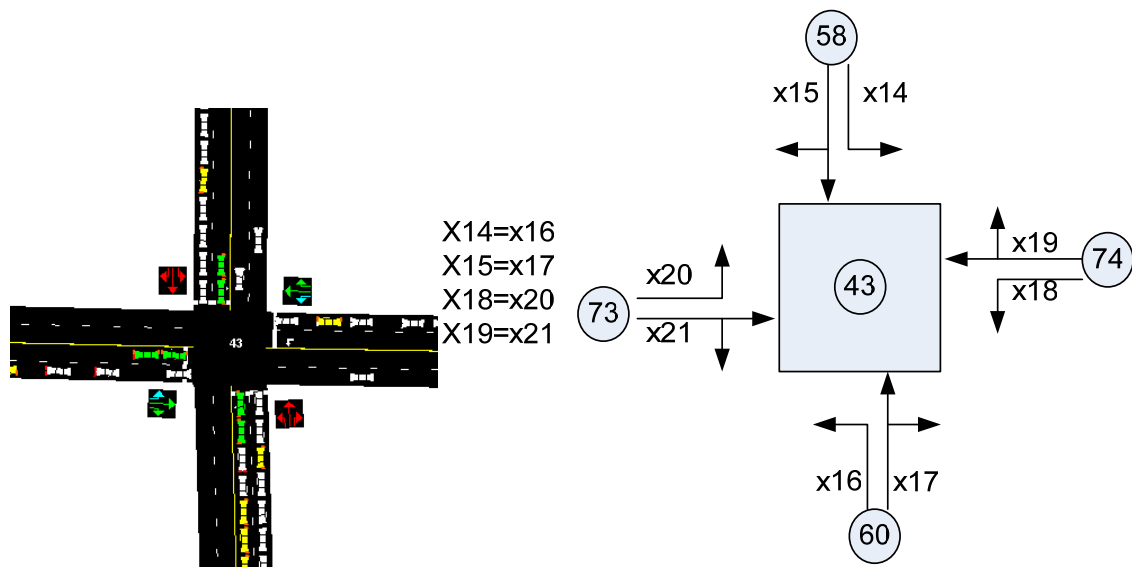


Figure 19 Geometric Information and Variable Number of Intersection Node 43

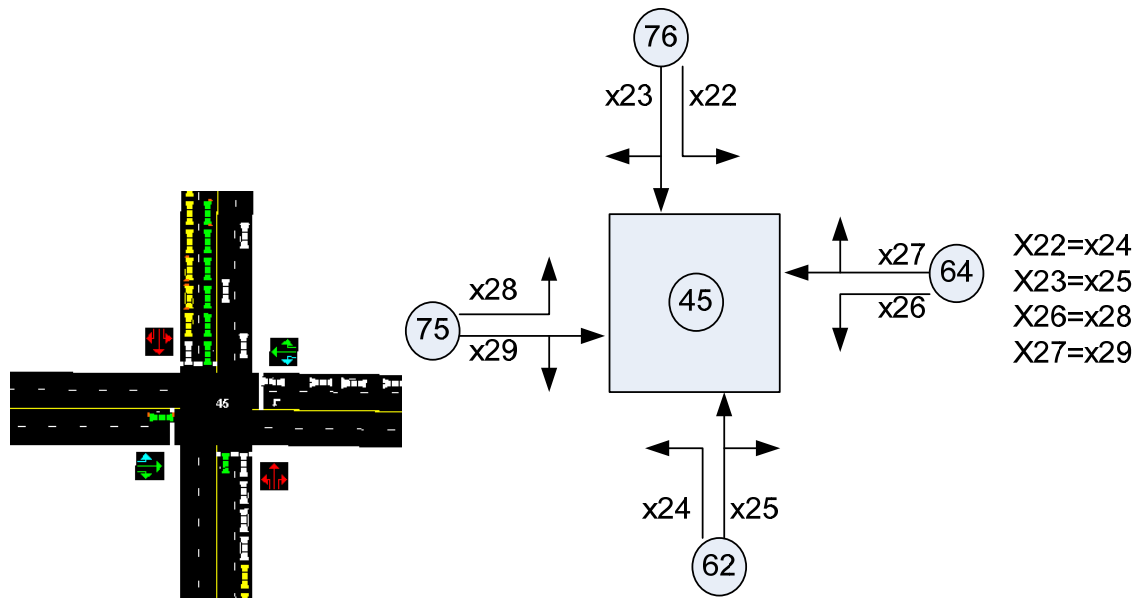


Figure 20 Geometric Information and Variable Number of Intersection Node 45

Node 39 is a T intersection with a two-phase pre-time signal plan. The rest three intersections 41, 43 and 45 are standard four-leg intersection with pre-time four-phase signal plan. The signal plan drawing of these two types of intersection from TRANSYT-7F is illustrated in Figure 21 and Figure 22. The cycle length, phase and time splits as well as the saturation flow rate for each lane group are shown in the figures.

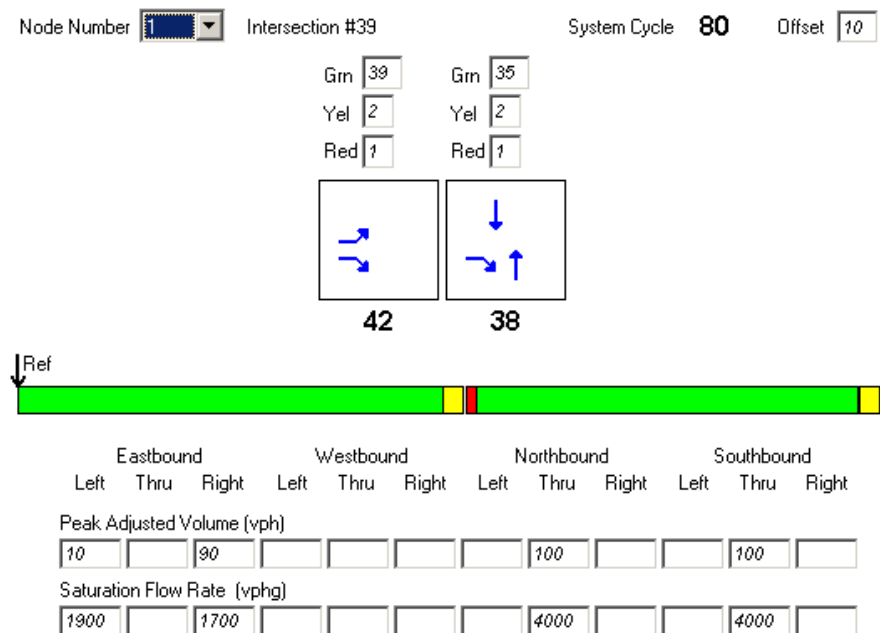


Figure 21 T Intersection, Two-Phase Signal Plan at Node 39

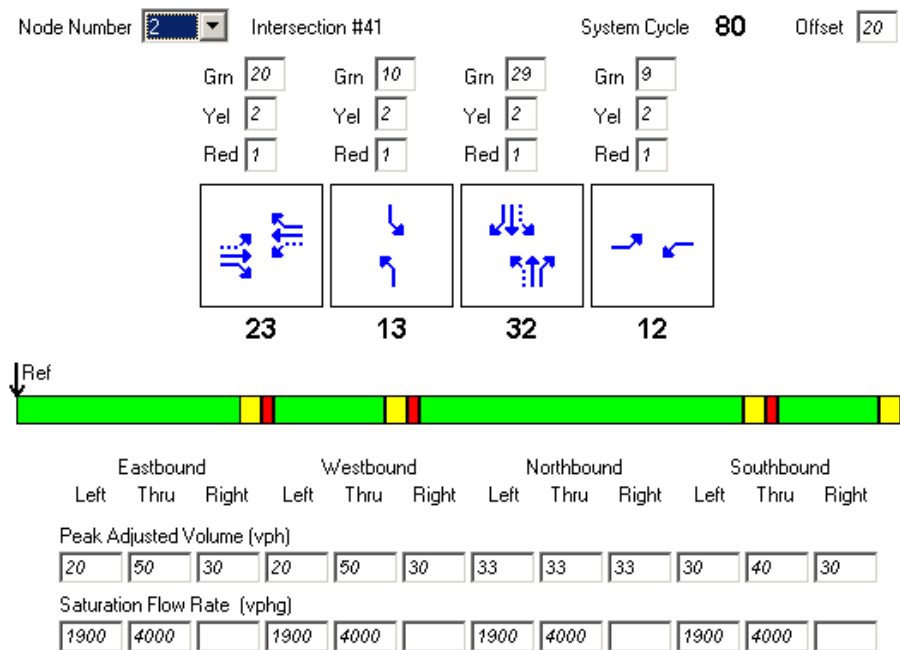


Figure 22 Four-Leg Intersection, Four-Phase Signal Plan at Node 41, 43, and 45

CORSIM Running Control

For all the aforementioned six scenarios, three-hour simulation is conducted. The simulation is separated into three one-hour periods. Time interval length is set to 120 seconds for all the periods. All configurations for the three periods are the same. Other running control parameters are set up as the CORSIM default values. Because of the stochastic nature of simulation, twenty CORSIM runs for the base scenarios and ten RTE optimization runs for all three algorithms are completed for comparison purpose. For observing statistical algorithms' properties, more than thirty-two runs were conducted for the three algorithms under moderate traffic volume, with a pre-run turning percentage at the diversion node.

Base case turning percentage at the diversion node is set to zero for the convenience of model analysis. Nevertheless, the base turning percentage can be overlapped with the proposed turning percentage because the model optimization only takes the volume and speed as input data; the model does not consider the turning percentage directly. After the new diversion, percentage is returned by the model, each vehicle whose destination is not the off-ramp link will face a choice to divert. The diversion will follow an extreme distribution with a mean value associated with the returned ideal DMS message level. Thus, the base turning percentage will not affect the proposed DMS diversion level. On the other hand, the base turning movement will not be affected by the DMS level either.

Variables

Variables of the proposed system are classified into two categories, network state variables, and independent decision variables. Network state variables are used to describe the traffic conditions of the network at a specific time. Most of the time, volume and speed are the ones required by the model. Independent decision variables are the ones getting return value from the optimization model. Independent variables mainly include diversion rate (associated with DMS message level), alternative road signal cycle length and green times for all the lane groups of alternative road intersections.

The diversion volume, which is returned from the optimization model directly, is then converted to DMS level following the predefined level-mapping table about DMS level and expected turning rate. Therefore, the diversion volume discussed here is not an arbitrary number; on the contrary, it is only a

certain predefined portion of the demand. This method guarantees that the DMS message level is within the predefined range.

Network State Variables

Volume and speed of the studied network are the traffic state variables to describe a network state. For the example network, volumes include freeway volume and alternative road volumes from all approaches. The freeway has only one entrance; therefore, there is only one demand volume of the system to be considered for a diversion split. However, there are eight detector stations on the freeway, which means there are eight speeds and volumes for freeway segment at a time. Volume data from each detector station are needed for speed forecast from the speed-flow curve.

The alternative road has four intersections with fifteen legs. Each leg except the ones on the T intersection is split into two lane groups, left turn traffic and through with right turn traffic. For the T intersection, not all the legs have two lane groups. As a result, twenty-seven lane groups need volume values for each optimization call.

Independent Decision Variables

The independent decision variables are the ones got return values from MATLAB optimization model. In the example network, there are twenty-seven green times corresponded to the same number of volumes provided from the state variables. The other two decision variables are the diversion rate (DMS message level) and cycle length for all intersections on alternative road.

The independent variables and their explanations for the studied example network are listed as followings. At the same time, they are also shown in Figure 17 to Figure 20.

x(1), Diversion message level, it is converted from diversion volume following predefined mapping table.

x(2), Common cycle length of alternative road intersections.

x(3), Green time of intersection 39, approach 1, group 1, off-ramp.

x(4), Green time of intersection 39, approach 2, group 1, NORTH BOUND, South Approach.

x(5), Green time of intersection 39, approach 3, group 1, SOUTH BOUND, North Approach.

x(6), Green time of intersection 41, approach 1, group 1, SOUTH BOUND, North Approach.

x(7), Green time of intersection 41, approach 1, group 2, SOUTH BOUND, North Approach.

x(8), Green time of intersection 41, approach 2, group 1, NORTH BOUND South Approach.

x(9), Green time of intersection 41, approach 2, group 2, NORTH BOUND South Approach.

x(10), Green time of intersection 41, approach 3, group 1, EAST BOUND West Approach.

x(11), Green time of intersection 41, approach 3, group 2, EAST BOUND West Approach.

x(12), Green time of intersection 41, approach 3, group 1, WEST BOUND East Approach.

x(13), Green time of intersection 41, approach 4, group 2, WEST BOUND East Approach.

x(14), Green time of intersection 43, approach 1, group 1, SOUTH BOUND North Approach.

x(15), Green time of intersection 43, approach 1, group 2, SOUTH BOUND North Approach.

x(16), Green time of intersection 43, approach 2, group 1, NORTH BOUND South Approach.

x(17), Green time of intersection 43, approach 2, group 2, NORTH BOUND South Approach.

- x(18), Green time of intersection 43, approach 3, group 1, EAST BOUND West Approach.
- x(19), Green time of intersection 43, approach 3, group 2, EAST BOUND West Approach.
- x(20), Green time of intersection 43, approach 4, group 1, WEST BOUND East Approach.
- x(21), Green time of intersection 43, approach 4, group 2, WEST BOUND East Approach.
- x(22), Green time of intersection 45, approach 1, group 1, SOUTH BOUND North Approach.
- x(23), Green time of intersection 45, approach 1, group 2, SOUTH BOUND North Approach.
- x(24), Green time of intersection 45, approach 2, group 1, NORTH BOUND South Approach.
- x(25), Green time of intersection 45, approach 2, group 2, NORTH BOUND South Approach.
- x(26), Green time of intersection 45, approach 3, group 1, EAST BOUND West Approach.
- x(27), Green time of intersection 45, approach 3, group 2, EAST BOUND West Approach.
- x(28), Green time of intersection 45, approach 4, group 1, WEST BOUND East Approach.
- x(29), Green time of intersection 45, approach 4, group 2, WEST BOUND East Approach.

Constraints

The constraints matrix of signal variables is listed in Table 4. The constraint of diversion volume is given by boundaries; therefore, it is not set up in Table 4. The constraints for the proposed model are only linear equality constraints and boundaries. The equality constraints are given in the form of $Ax = b$. The last column of the Table 4 is for b and the rest columns are for A . Basically, there are two types of constraints for the model. The first type is signal cycle length and time splits constraint. The summation of all the time splits of each phase in a cycle, namely green time, yellow time and all red time should be equal

to the cycle length. The second type of constraint is for the green times in a phase. For a given phase, it may involve more than one lane groups. Because the green time variables are associated with lane groups, it is necessary to constrain the group green time in a same phase to have the same length. This green time equality constraints are also demonstrated in Figure 17 to Figure 20.

The constraints in the form of lower and upper boundaries are given in a separate vector. The lower bound of the cycle length is computed from HCM2000 equations. Because the maximum CORSIM cycle length is 256 seconds, the upper bound of cycle length is set to 256 seconds. The minimum green time is set to five seconds and the maximum green time is set to 232 seconds (when all other phases are set to minimal green time for a four-phase intersection). All red and yellow time is set to constant for all intersections.

Model Constraints Implementation

128

Case Implementations in RTE

For the RTE implementation, data is exchanged between RTE and MATLAB model and hard disk data files in the numerical example is used for this implementation. Appendixes 2 to 5 are example files for related input and output data from the studied cases. Appendix B is the user RTE configuration file that provides critical user configure data such as intersection node numbers, path links, algorithm selection and parameters such as roll periods length and so on. For those case studies, computation period length is set to 3600 seconds, the DMS roll period is set to 1800 seconds and the signal roll period is set to 900 seconds.

Ideally speaking, the roll period length for the signal update is preferably less than two minutes. However, because the upper bound of the cycle length is set to 232 seconds, the roll period should be equal to or longer than this time even though the algorithm execution time can satisfy the two minutes roll period requirements. On the other hand, the roll period's lengths will affect the total number of MATLAB calls. If the MATLAB is called too frequently, the whole RTE execution time may be too long from all the combinations of multiple runs, scenarios and algorithms for evaluation. Based on the above reasons, the signal roll period length is taken to be 900 seconds in the numerical examples (user can define the roll period in the configuration file). However, there should be no problems in the real-world implementation if the signal roll period is set to 120 seconds if signal cycle length is less than 120 seconds.

Appendix C and Appendix D are the data exchange files for CORSIM RTE and MATLAB model. RTE provides volume, speed and such network state data to MATLAB and MATLAB returns solution of DMS message level, cycle length and green times of all lane groups.

Appendix E is an example of CORSIM RTE running log file. In the log file, CORSIM running progress information is recorded. For example, when the MATLAB is called, what the roll period type is and how long the optimal model ran, can all be found from the log file. The overall execution time for a scenario is output to log file by CORSIM at the end of simulation as well.

Functions Called at Call Points

The following Figure 23 provides an example of the call points RTE function configuration for the case study. Five call points (excluding the default “Exit” point) are used. At each call point, a function from RTE DLL is assigned. At the lower left corner of the figure, there is a list of available functions from the RTE DLL file. Users can pick and assign a function to a call point.

Figure 24 and Figure 25 show an example of the SQP execution plots. The upper plot is the object function value at each step. The lower plot is the step size of x (distance from one x to another one). Figure 24 is for the full optimization with diversion volume (totally 29 variables) and Figure 25 is for the signal optimization that has the diversion volume as a known value (totally 28 variables).

Figure 26 and Figure 27 show an example of the Pattern Search execution plots. The upper left plot is the objective function value at each step. The lower left plot is the x step size; the upper right plot is the mesh size (expands and shrinks during the searching) and the lower right plot is the value of independent variables of the last iteration.

Figure 28 and Figure 29 show an example of the GA execution plots. The upper plot is the fitness value, and the lower plot is the average distance of x , which indicates the size of searched range. The objective function value serves as the fitness value for this GA model solution.

The SQP and PS plots show that the optimization for signal only (28 independent variables excluding diversion volume) has a faster speed than the full independent variables optimization (all 29 independent variables). However, plots of GA do not show that the signal optimization has a faster speed tendency than optimization of all independent variables.

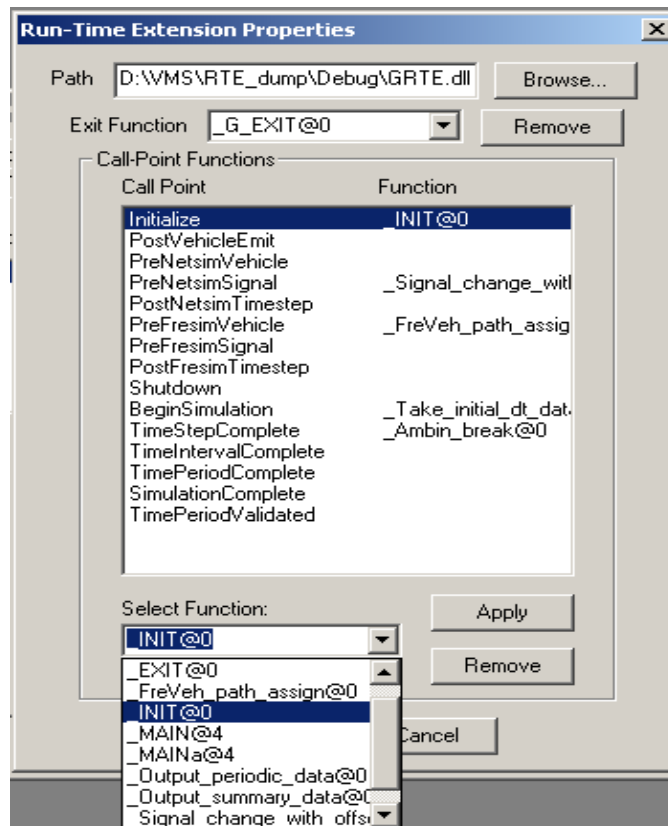


Figure 23 CORSIM RTE Configurations for Case Study

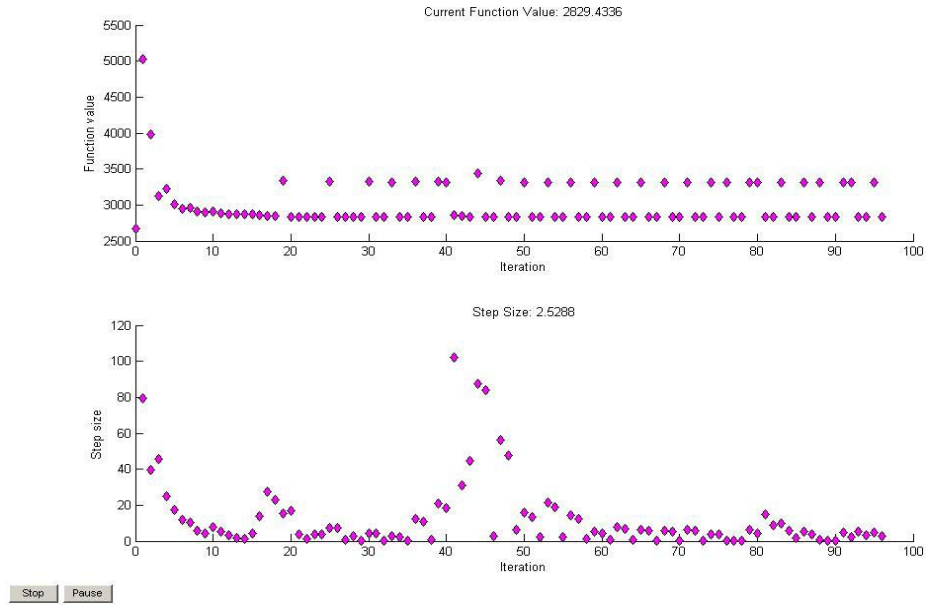


Figure 24 Illustration of SQP Optimization Progress of Signal Parameters and Diversion Volume

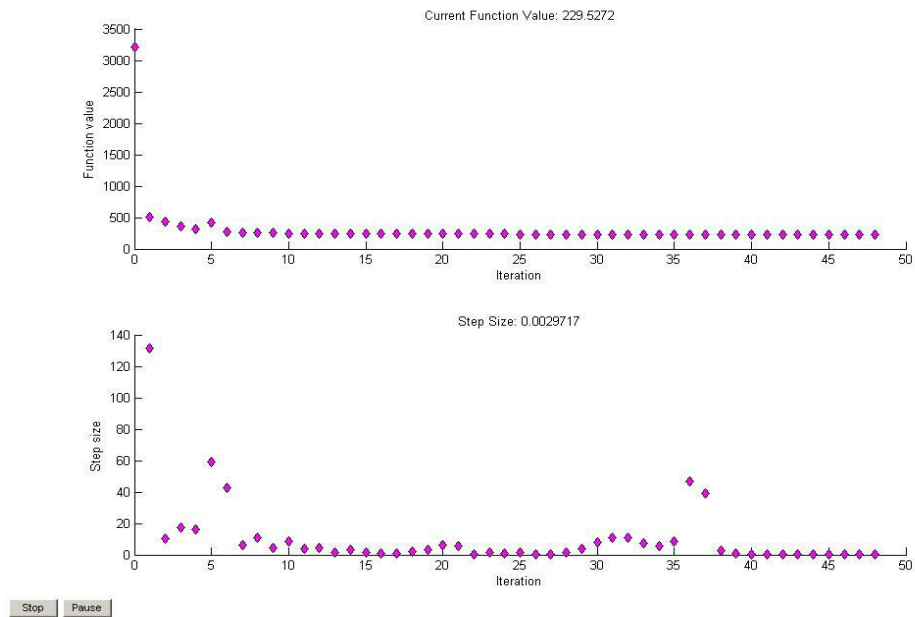


Figure 25 Illustration of SQP Optimization Progress of Signal Parameters

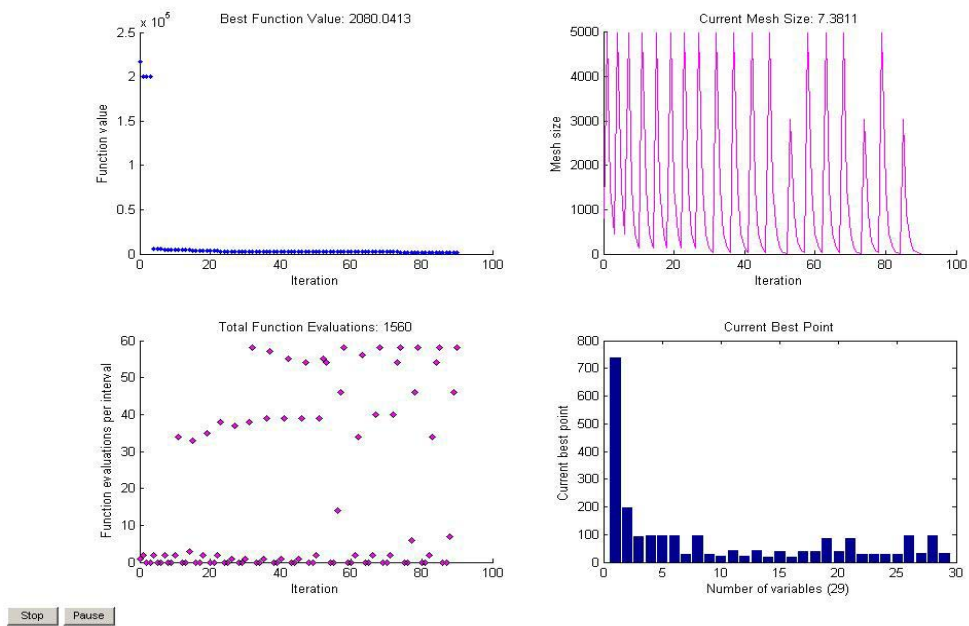


Figure 26 Illustration of Pattern Search Optimization Progress of Signal Parameters and Diversion Volume

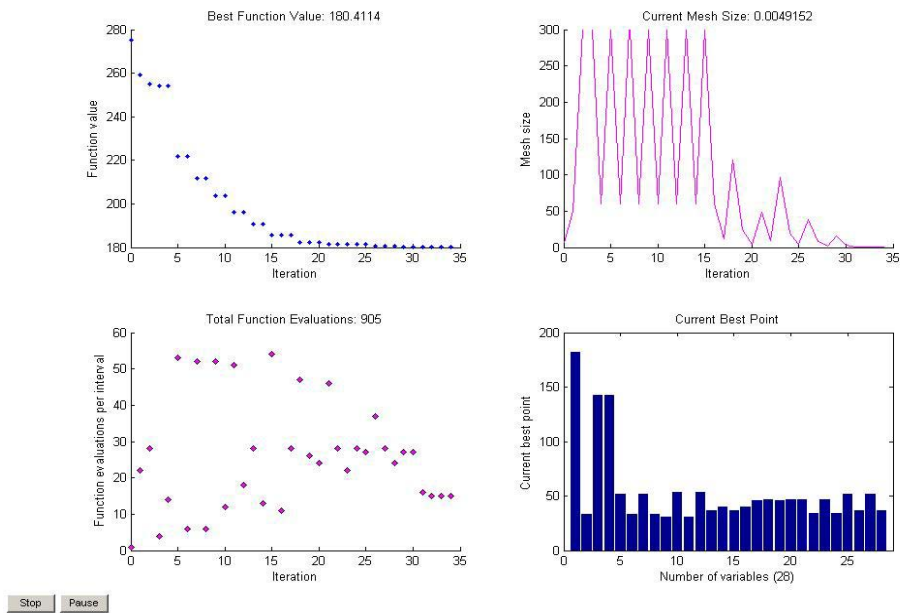


Figure 27 Illustration of Pattern Search Optimization Progress of Signal Parameters

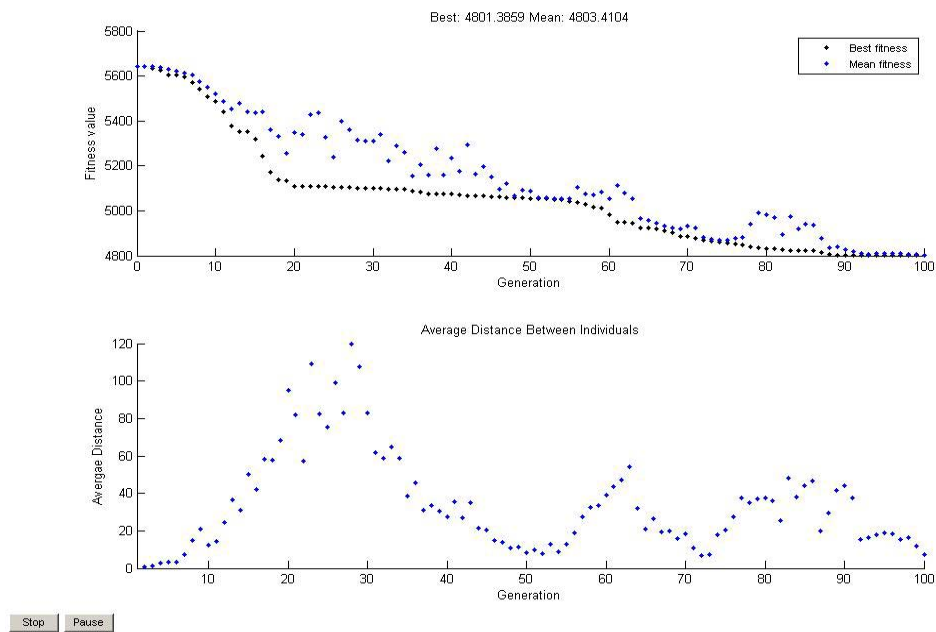


Figure 28 Illustration of GA Optimization Progress of Signal Parameters and Diversion Volume

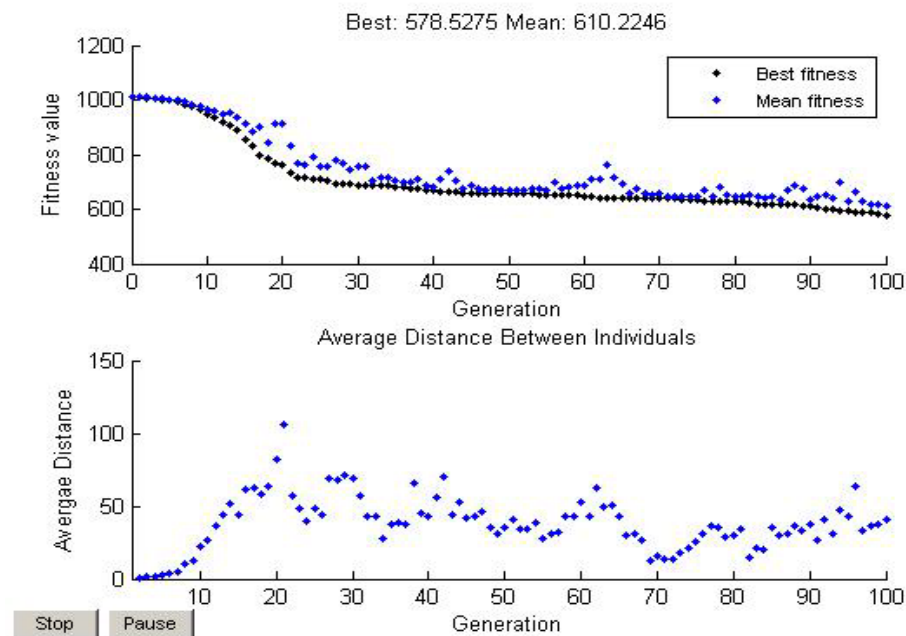


Figure 29 Illustration of GA Optimization Progress of Signal Parameters

Model Validation

Model validation is conducted by comparing the model solution with the TRANSYT-7F optimized solution. CORSIM serves as a unique solution evaluation platform for different models under these kinds of circumstances. During the evaluations, TRANSYT-7F is a first run over a given example network without optimal signal parameters (signal splits, cycle length and offset). Then a set of optimized signal parameters is generated by TRANSYT-7F with CORSIM Processing Method, which uses CORSIM as its algorithm evaluation tool and will code the solution in to CORSIM TRF file directly. TRANSYT-7F then codes the optimized parameters back into the CORSIM TRF file, which is used as a base case to run the proposed model for another set of solutions. A new CORSIM simulation will run with the developed RTE and MATLAB optimization module. Since the proposed model is an on-line optimization model, the RTE updates the CORSIM simulation by changing the related memory variables after the solution is generated. The TRANSYT-7F does not include the diversion volume in its optimal solution; therefore, CORSIM MOEs from its solution cannot be compared directly with those from the proposed model. As an alternative, two scenarios are set up depending on the turning percentage of the diversion node. The first scenario is to compare two solutions without thinking about the diversion volume difference, and the second scenario is to set an average turning percentage from model solutions for the TRANSYT-7F optimization. Thus, the TRANSYT-7F's solutions consider the increased alternative diversion volume, and the solutions are more comparable under the second scenario.

CORSIM Processing Method in TRANSYT-7F requires CORSIM version 5.1 or above. CORSIM Input TRF files are imported into TRANSYT-7F for direct CORSIM simulation processing. A

TRANSYT-7F's TIN file is generated to store user configurations. CORSIM Processing Method is accomplished by executing the GA from TRANSYT-7F, and running the CORSIM simulation engine in a "batch mode". In other words, TRANSYT-7F will generate a serial of CORSIM TRF files that are coded with parameters from GA solutions. After running each batch, parent's selection is finished by evaluating the Performance Index (PI) and disutility index (DI, Figure 30). Network wide delay time is chosen as PI in TRANSYT-7F in order to compare with the proposed model. Other options are set as similar as the CORSIM simulation configuration for model solutions.

The 'Run Instructions' dialog box is shown with the following settings:

- Simulation Method:** ☒ Step-Wise, ☐ Link-Wise
- Simulation Type:** ☒ Optimization, ☐ Estimation
- Genetic Algorithm Settings:** ☒ Genetic Algorithm, ☐ Hill-Climb
- Phasing Optimization Settings:** ☐ Narrow Output, ☒ Wide Output
- Cycle Lengths:** Lower Cycle Length (sec) = 80, Original Cycle Length (sec) = 80, Upper Cycle Length (sec) = 140, Cycle Increment (sec) = 5
- Uncoordinated Network:** ☐ Uncoordinated Network
- Delay Time:** PI
- Network Wide:** DI
- Simulation Period (min):** 15
- Random # Seed Variations:** 1
- Initialization Period (sec):** 600
- Volume Multiplier (%):** 100
- Output Options:** ☒ Cycle Length, ☐ Phasing Sequence, ☒ Splits, ☒ Offsets
- Initial Timing Flags:** ☐ Initial Timing Flag 1, ☐ Initial Timing Flag 2, ☐ Ignore Actuated Flags
- Buttons:** Run Title, Cancel, OK

Figure 30 Demonstration of TRANSYT-7F Optimization Configuration

In the center of the TRANSYT-7F optimization configuration window (Figure 30), there is a button “Genetic Algorithm Settings” for GA parameters configuration (Figure 31). Value of parameters and options are set as close as the ones used in model’s GA solution.

Once TRANSYT-7F finds the optimal solution , a punch file in TRF format is saved and it can serve as the base case for the proposed model solution.

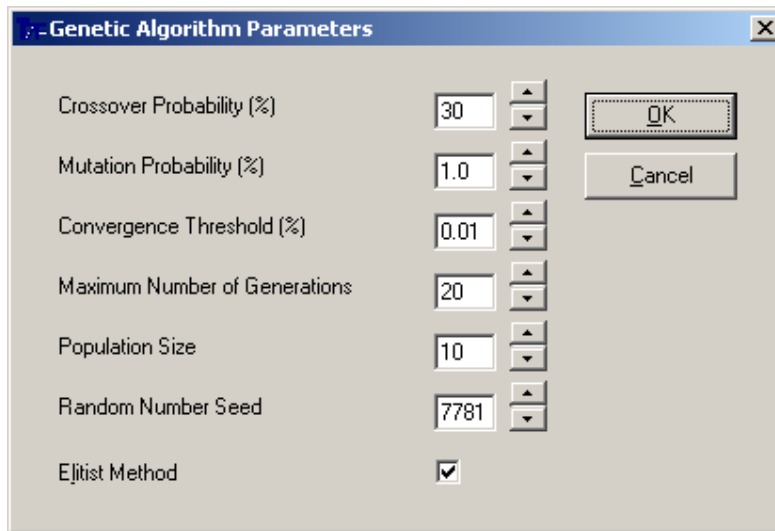


Figure 31 TRANSYT-7F Genetic Algorithm Parameters

Result Analysis

There are totally three one-hour simulation periods for each of the studied scenarios. The first period serves as a data collection period and no optimization solution is generated from this period. Started from the second period, the optimal solution from MATLAB is updated into the CORSIM simulation before CORSIM resumes simulation from waiting for MATLAB’s termination. However, the optimized

signal plan will be updated at certain time offsets. The fourth intersection is 8.68 miles from the diversion node. Signal offset under the free flow speed of 30 mph is 1042 seconds, which is the minimal offset of the fourth intersection for all the scenarios if the first intersection serves as the pivot intersection. Because of the existence of such a big value, the lag of signal updates will make the second simulation period run partly under the original timing plan (intersections far from the diversion node) and partly under the optimized timing plan (intersections close to the diversion node). The MOEs from this second period in CORSIM cannot fully represent the performance of the optimal solution. The third period gets all timing plans updated with the optimal solution. Only the MOE from the third period and thereafter can represent the performance of optimal solution. Conclusion for algorithms of this dissertation will mainly rely on the third simulation period.

CORSIM MOE Outputs

Two network MOEs are selected to evaluate the network improvements from the proposed model: network-wide total travel delay and network-wide average speed. The before and after case studies are all completed in CORSIM with the same running parameters. CORSIM MOEs can be found in the output file from a single run at the section end of each period and the MOE output data file from multiple runs (user specified).

Network-wide total delay and average speed are analyzed under a single run and multiple runs during the evaluation. For the statistical analysis of algorithm property, fifty runs for one out of the six scenarios are conducted. Execution speed of algorithms is also an important property to evaluate algorithms.

Therefore, algorithm execution time for overall RTE execution and even a single model execution along the progress is recorded and analyzed later in this section.

During the scenario analysis stage, single CORSIM runs are first conducted to get the average speed and total delay. Then, multiple CORSIM runs are conducted with consideration of the simulation's randomness status. Twenty runs are set for the base cases, which are merely CORSIM runs and have nothing to do with RTE then. Ten runs are conducted for RTE optimization cases. The reason that RTE runs are less than CORSIM runs is that the RTE needs longer execution time (about twice of CORSIM running time). To save time, the RTE gets only ten runs to observe the property of multiple CORSIM runs. Finally, there are fifty runs for all three algorithms to observe the statistical property over the same scenario.

Analysis of Single CORSIM Run

All the three algorithms are conducted in a single CORSIM and run over the six scenarios. The total delay and average speed of network from CORSIM are listed in the Table 5 to Table 7 for SQP, PS and GA algorithm respectively.

The SQP solutions generate a reduction of 8.97 percent to 14.09 percent in total delay and increase of 2.62 percent to 10.76 percent in average speed among the six scenarios. The highest total delay decrease is from a moderate congestion scenario and the lowest total delay decrease is from a heavy congestion situation (Table 5).

The PS solutions show an average reduction of 7.24 percent total delay decrease for moderate congestion without a pre-determined diversion percentage. Meanwhile, the PS solutions generate an average increase of 3.81 percent in average speed. However, the PS fails to reduce total delay and increase the average speed for the light congestion with base diversion percentage (Table 6).

The GA gets worse than PS on the algorithm performance failures. GA failed three times in total delay decrease and average speed increase. The average reduction in total delay of GA is 5.82 percent and the average speed increase is 3.08 percent (Table 7).

Since the SQP does not fail in any scenarios that are the same as the ones PS and GA dealt with. It might be from the PS and GA algorithm parameters, initial variable values or from combinations of both. In the statistical analysis section, this will be discussed to find out more about PS and GA failures.

In Table 8, total delay from three algorithms is analyzed. For each scenario, the minimum total delay among the three algorithms is found first. Delays from other two algorithms are then compared with the minimal total delay to verify the solution quality. Among the three algorithms, the delay from SQP's solution varies from 100.00 percent to 100.21 percent if compared with the minimal delay from all the three algorithms. In other words, the SQP solution either results in the minimal delay among the three algorithms or generates a delay value that is very close to the minimal delay (0.21 percent higher than the minimal delay).

The delay from PS's solution varies from 100.00 percent to 143.19 percent if compared with the minimal delay of all the three algorithms, while the delay from of GA's solution varies from 104.82 percent to 121.76 percent. Even though the PS algorithm gets a bigger difference range than the GA does, the

median of the PS algorithm is 104.78 percent, which means only a few scenarios get poor solutions. Table 8 shows that PS failed in both light congestion scenarios while all other delays were within 5 percent of the minimal delay of all algorithms. The GA gets a median of 110.70 percent, which means that more than half of the delays from GA solution are 10.7 percent bigger than the minimal delay among all algorithms.

The GA algorithm does not have even one minimal solution among the three algorithms. All the minimal delay are from either SQP or PS. The SQP has five out of six minimal delay solutions among the three algorithms. However, it is still too early to say GA is not as good as SQP or PS algorithm. More evaluations about algorithm solution quality will be conducted in the multiple runs and statistic analysis section.

Solution Analysis of Multiple CORSIM Runs

Because the SQP algorithm has a better performance from the basic evaluation and the single CORSIM run evaluation, it is selected to run multiple times for all six scenarios and to compare with the CORSIM multiple runs of base scenarios. The rest of the two algorithms only run multiple times for two out of the six scenarios to observe the performance under multiple runs. Table 9 and Table 10 show the total delay and average speed analysis of SQP. The multiple runs generate an average total delay decrease range from 8.97 percent to 27.64 percent and a mean value of average speed increase range from 2.55 percent to 23.12 percent for the six scenarios. The average total delay decrease for all six scenarios is 15.03 percent, which is a slightly lower than the single run's 15.25 percent decrease in the last section. At the same time, the mean value of the average speed increased slightly from 7.38 percent to 8.53 percent. As the

tendency showed in single run, the multiple runs also show that the total delay increase and the average speed decrease tendency.

Moderate congestion without base turning percentage scenario is run multiple times for PS and GA algorithms and the same scenario with 20 percent base turning percentage is run multiple times for GA. Table 11 and Table 12 show the result comparison. Multiple-run of PS generates a lower total delay decrease (7.24 percent) than that from signal run (13.77 percent), and a lower average speed increase (3.81 percent) than that from single run 6.96 percent. The multiple-run of GA gets similar results and the multiple-run generates smaller reduction in total delay and increase of average speed than that from a single run. It is 5.82 percent against 9.30 percent, and 3.08 percent against 4.80 percent for the total delay and average speed, respectively.

Statistic Solution Analysis of Multiple CORSIM Runs

The multiple runs indicate that the simulation randomness status and randomness from local optimal solution property make the solution vary from one run to another. More runs are conducted in order to observe and analyze the statistical properties of these algorithms. Fifty CORSIM RTE runs are simulated for each of the algorithms and the statistical characteristics are shown in the Figure 32. The *x-coordinates* are the number of runs, the major *y-coordinates* at the left are total delay in veh-hours, and the minor *y-coordinates* at the right are average speed in mph. The lines of total delay and the lines of average speed have an obvious mirror effect along the horizontal direction, which is consistent with the physical relationship between total delay and average speed.

Table 13 shows the means and standard deviations of the multiple runs for the three algorithms.

Figure 33 and Figure 34 visualize the tendency of total delay and average speed. Obviously, SQP generates the best solutions with minimal value and moderate standard deviation; the PS ranks the second and the GA gets the highest delay among the three algorithms. The mean delay values from three algorithms and the mean speed values from SQP vs. PS, SQP vs. GA are statistically different with a 95 percent confidence. Whereas, the mean speed value from PS and GA are not different at a 95 percent confidence interval. The tests are made from t-test (two unequal variances sample) pair by pair among the three algorithms. So far, it is safe to say that the best solution is from SQP. This conclusion confirms the results from the basic algorithm evaluation, the single CORSIM RTE runs and multiple RTE runs.

Table 5
CORSIM MOE Comparison between before and after SQP Optimization

SQP	Scenarios	V/C	Divs %	CORSIM Period	Total Delay (Veh-Hours)			Average Speed (mph)		
					CORSIM	Model	Change %	CORSIM	RTE	Change %
TRANSYT-7F optimized signal without diversion volume	Heavy congestion	1.7	0	2	2965.68	2849.56	-3.92%	22.43	23.24	3.61%
				3	6261.95	5762.18	-7.98%	18.13	19.91	9.82%
	Moderate congestion	1.3	0	2	1559.83	1451.90	-6.92%	28.82	29.44	2.15%
				3	3280.80	2699.74	-17.71%	25.00	27.02	8.08%
	Light congestion	1.1	0	2	808.94	779.87	-3.59%	33.99	33.93	-0.18%
				3	1534.07	1235.97	-19.43%	31.98	33.24	3.94%
TRANSYT-7F optimized signal with diversion volume	Heavy congestion	1.7	30	2	2902.73	2769.18	-4.60%	22.71	23.62	4.01%
				3	6185.33	5681.20	-8.15%	18.28	20.08	9.85%
	Moderate congestion	1.3	20	2	1592.48	1505.78	-5.44%	28.65	29.08	1.50%
				3	3280.29	2778.90	-15.28%	25.02	26.73	6.83%
	Light congestion	1.1	20	2	771.62	754.65	-2.20%	34.37	34.34	-0.09%
				3	1483.82	1143.33	-22.95%	32.28	34.13	5.73%

Table 6
CORSIM MOE Comparison between before and after Pattern Search Optimization

Pattern Search	Scenarios	V/C	Divs %	TP	Total delay Total Delay (Veh-Hours)			Average Speed (mph)		
					CORSIM	RTE	Change	CORSIM	RTE	Change
TRANSYT-7F optimized signal without diversion volume	Heavy congestion	1.7	0	2	2965.68	2829.18	-4.60%	22.43	23.34	4.06%
				3	6261.95	5749.97	-8.18%	18.13	19.96	10.09%
	Moderate congestion	1.3	0	2	1559.83	1450.41	-7.01%	28.82	29.48	2.29%
				3	3280.80	2828.91	-13.77%	25.00	26.74	6.96%
	Light congestion	1.1	0	2	808.94	771.33	-4.65%	33.99	33.85	-0.41%
				3	1534.07	1488.54	-2.97%	31.98	31.67	-0.97%
TRANSYT-7F optimized signal with diversion volume	Heavy congestion	1.7	30	2	2902.73	2756.00	-5.05%	22.71	23.66	4.18%
				3	6185.33	5754.90	-6.96%	18.28	19.92	8.97%
	Moderate congestion	1.3	20	2	1592.48	1501.45	-5.72%	28.65	29.07	1.47%
				3	3280.29	2883.64	-12.09%	25.02	26.54	6.08%
	Light congestion	1.1	20	2	771.62	742.04	-3.83%	34.37	34.10	-0.79%
				3	1483.82	1637.11	10.33%	32.28	30.45	-5.67%

Table 7
CORSIM MOE Comparison between before and after GA Optimization

GA	Scenarios	V/C	Divs %	TP	Total Delay (Veh-Hours)			Average Speed (mph)		
					CORSIM	RTE	Change	CORSIM	RTE	Change
TRANSYT-7F optimized signal without diversion volume	Heavy congestion	1.7	0	2	2965.68	2853.18	-3.79%	22.43	23.14	3.17%
				3	6261.95	6032.34	-3.67%	18.13	19.12	5.46%
	Moderate congestion	1.3	0	2	1559.83	1540.83	-1.22%	28.82	29.02	0.69%
				3	3280.80	2975.60	-9.30%	25.00	26.20	4.80%
	Light congestion	1.1	0	2	808.94	817.82	1.10%	33.99	33.84	-0.44%
				3	1534.07	1441.44	-6.04%	31.98	32.46	1.50%
TRANSYT-7F optimized signal with diversion volume	Heavy congestion	1.7	30	2	2902.73	2902.57	-0.01%	22.71	22.83	0.53%
				3	6185.33	5955.08	-3.72%	18.28	19.05	4.21%
	Moderate congestion	1.3	20	2	1592.48	1659.68	4.22%	28.65	28.20	-1.57%
				3	3280.29	3383.55	3.15%	25.02	24.70	-1.28%
	Light congestion	1.1	20	2	771.62	768.29	-0.43%	34.37	34.32	-0.15%
				3	1483.82	1271.10	-14.34%	32.28	33.56	3.97%

Table 8

CORSIM Total Delay Comparison among Three Algorithms – Single Run

Categories	Scenarios	V/C	Divs %	TP	Total Delay from CORSIM RTE (Veh-Hour)			MIN Delay among Three Algorithms (Veh-Hour)	Percentage of the Minimal Delay among Three Algorithms		
					SQP	PS	GA		SQP	PS	GA
TRANSYT-7F optimized signal without diversion volume	Heavy congestion	1.7	0	2	2849.56	2829.18	2853.18	2829.18	100.72%	100.00%	100.85%
				3	5762.18	5749.97	6032.34	5749.97	100.21%	100.00%	104.91%
	Moderate congestion	1.3	0	2	1451.90	1450.41	1540.83	1450.41	100.10%	100.00%	106.23%
				3	2699.74	2828.91	2975.60	2699.74	100.00%	104.78%	110.22%
	Light congestion	1.1	0	2	779.87	771.33	817.82	771.33	101.11%	100.00%	106.03%
				3	1235.97	1488.54	1441.44	1235.97	100.00%	120.43%	116.62%
TRANSYT-7F optimized signal with diversion volume	Heavy congestion	1.7	30	2	2769.18	2756.00	2902.57	2756.00	100.48%	100.00%	105.32%
				3	5681.20	5754.90	5955.08	5681.20	100.00%	101.30%	104.82%
	Moderate congestion	1.3	20	2	1505.78	1501.45	1659.68	1501.45	100.29%	100.00%	110.54%
				3	2778.90	2883.64	3383.55	2778.90	100.00%	103.77%	121.76%
	Light congestion	1.1	20	2	754.65	742.04	768.29	742.04	101.70%	100.00%	103.54%
				3	1143.33	1637.11	1271.10	1143.33	100.00%	143.19%	111.18%

Table 9

SQP Total Delay Comparison between Single Run and Multiple Runs

SQP	Scenarios	V/C	Divs %	CORSIM Period	Total delay One Run (Veh-Hours)				Total delay Multirun (Veh-Hours)				Standard Deviation	
					CORSIM	RTE	Change %		CORSIM	RTE	Change %		CORSIM	RTE
TRANSYT-7F optimized signal without diversion volume	Heavy congestion	1.7	0	2	2965.68	2849.56	-3.92%		2893.22	2728.06	-5.71%		51.77	42.74
				3	6261.95	5762.18	-7.98%		6204.07	4489.28	-27.64%		133.06	64.72
	Moderate congestion	1.3	0	2	1559.83	1451.90	-6.92%		1583.25	1488.97	-5.95%		29.95	37.16
				3	3280.80	2699.74	-17.71%		3294.42	2855.47	-13.32%		73.57	65.06
	Light congestion	1.1	0	2	808.94	779.87	-3.59%		771.49	770.32	-0.15%		100.31	52.94
				3	1534.07	1235.97	-19.43%		1440.42	1252.57	-13.04%		183.12	81.64
TRANSYT-7F optimized signal with diversion volume	Heavy congestion	1.7	30	2	2902.73	2769.18	-4.60%		2877.86	2700.36	-6.17%		51.45	40.55
				3	6185.33	5681.20	-8.15%		6185.44	5630.79	-8.97%		/	59.22
	Moderate congestion	1.3	20	2	1592.48	1505.78	-5.44%		1575.05	1473.57	-6.44%		58.92	41.95
				3	3280.29	2778.90	-15.28%		3288.20	2824.84	-14.09%		90.42	75.01
	Light congestion	1.1	20	2	771.62	754.65	-2.20%		773.53	776.64	0.40%		79.18	57.16
				3	1483.82	1143.33	-22.95%		1446.84	1257.40	-13.09%		125.56	102.22

Table 10
SQP Average Speed Comparison between Single Run and Multiple Runs

SQP	Scenarios	V/C	Divs %	CORSIM Period	Average Speed (mph)			Average Speed Multirun (mph)			Standard Deviation	
					CORSIM	RTE	Change %	CORSIM	RTE	Change %	CORSIM	RTE
TRANSYT-7F optimized signal without diversion volume	Heavy congestion	1.7	0	2	22.43	23.24	3.61%	22.71	23.78	4.70%	0.24	0.20
				3	18.13	19.91	9.82%	18.22	22.43	23.12%	0.30	0.14
	Moderate congestion	1.3	0	2	28.82	29.44	2.15%	28.68	29.19	1.76%	0.21	0.27
				3	25.00	27.02	8.08%	24.96	26.44	5.92%	0.27	0.24
	Light congestion	1.1	0	2	33.99	33.93	-0.18%	34.39	34.18	-0.61%	1.03	0.56
				3	31.98	33.24	3.94%	32.57	33.39	2.55%	1.19	0.51
TRANSYT-7F optimized signal with diversion volume	Heavy congestion	1.7	30	2	22.71	23.62	4.01%	22.78	23.91	4.93%	0.23	0.19
				3	18.28	20.08	9.85%	18.24	20.21	10.76%	/	0.13
	Moderate congestion	1.3	20	2	28.65	29.08	1.50%	28.74	29.29	1.90%	0.42	0.30
				3	25.02	26.73	6.83%	24.99	26.54	6.21%	0.33	0.28
	Light congestion	1.1	20	2	34.37	34.34	-0.09%	34.37	34.12	-0.74%	0.81	0.61
				3	32.28	34.13	5.73%	32.52	33.37	2.62%	0.84	0.65

Table 11
Comparison of PS between Single Run and Multiple Runs, Moderate Congestion

PS	Scenarios	V/C	Divs %	TP	Total Delay (Veh-Hours)				Average Speed (mph)			
					CORSIM Multirun	RTE Multirun	Change Multirun	Change Single-run	CORSIM Multirun	RTE Multirun	Change Multirun	Change Single-run
TRANSYT-7F optimized signal, no diversion	Moderate congestion	1.3	0	2	1559.83	1509.13	-3.25%	-7.01%	28.82	29.04	0.76%	2.29%
				3	3280.80	3043.33	-7.24%	-13.77%	25.00	25.95	3.81%	6.96%

Table 12
Comparison of GA between Single Run and Multiple Runs, Moderate Congestion

GA	Scenarios	V/C	Divs %	TP	Total delay (Veh-Hours)				Average speed (mph)			
					CORSIM Multirun	RTE Multirun	Change Multirun	Change Single-run	CORSIM Multirun	RTE Multirun	Change Multirun	Change Single-run
TRANSYT-7F optimized signal, no diversion	Moderate congestion	1.3	0	2	1559.83	1573.72	0.89%	-1.22%	28.82	28.74	-0.28%	0.69%
				3	3280.80	3089.82	-5.82%	-9.30%	25.00	25.77	3.08%	4.80%

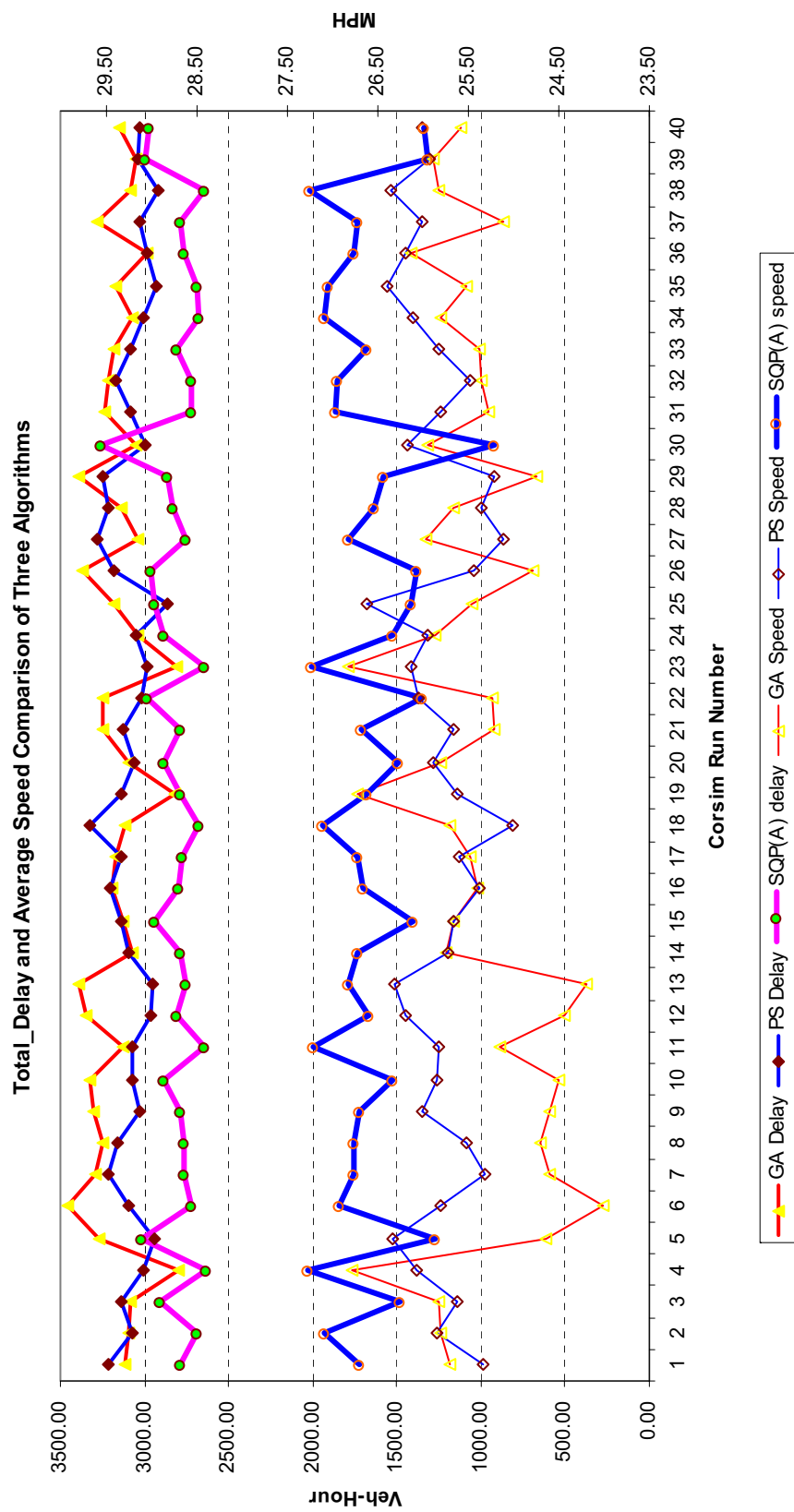


Figure 32 Statistical MOE Comparisons among the Three Algorithms

Table 13

Statistical CORSIM MOE Comparison among the Three Algorithms

Moderate 20	Mean	Stdev
SQP Delay (Veh-Hours)	2806.2	122.92
SQP Speed (mph)	26.6	0.44
PS Delay (Veh-Hours)	3076.17	105.62
PS Speed (mph)	25.84	0.38
GA Delay (Veh-Hours)	3159.62	151.75
GA Speed (mph)	25.44	0.64

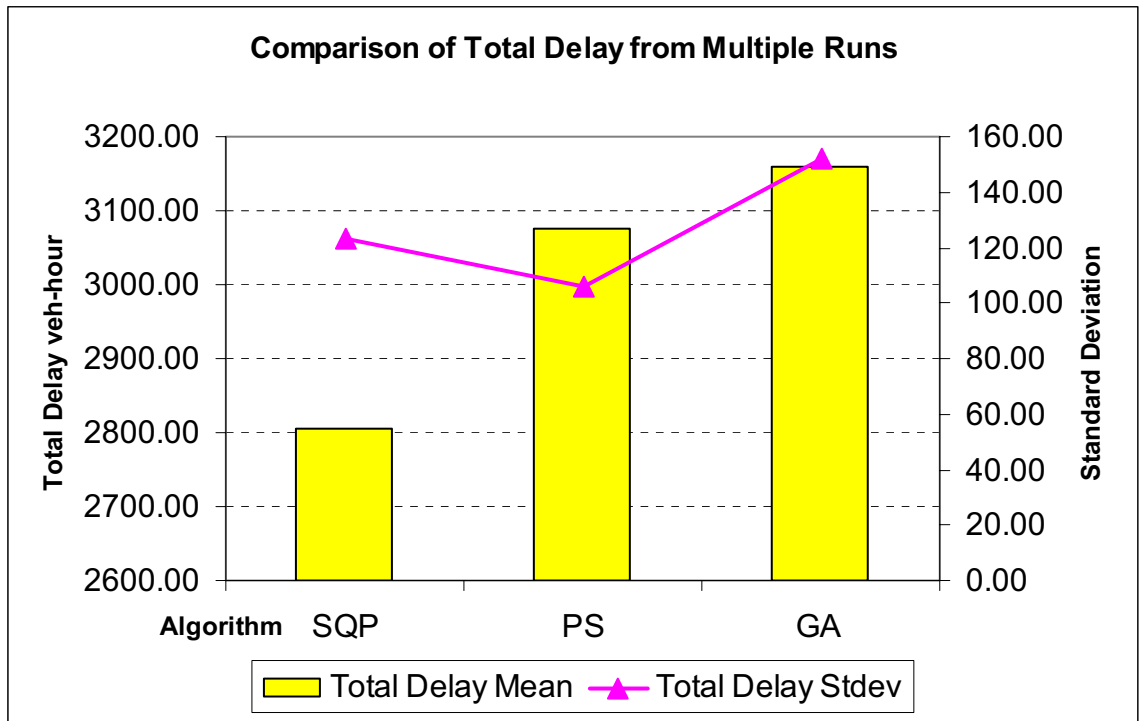


Figure 33 Statistical Total Delay Comparisons among the Three Algorithms

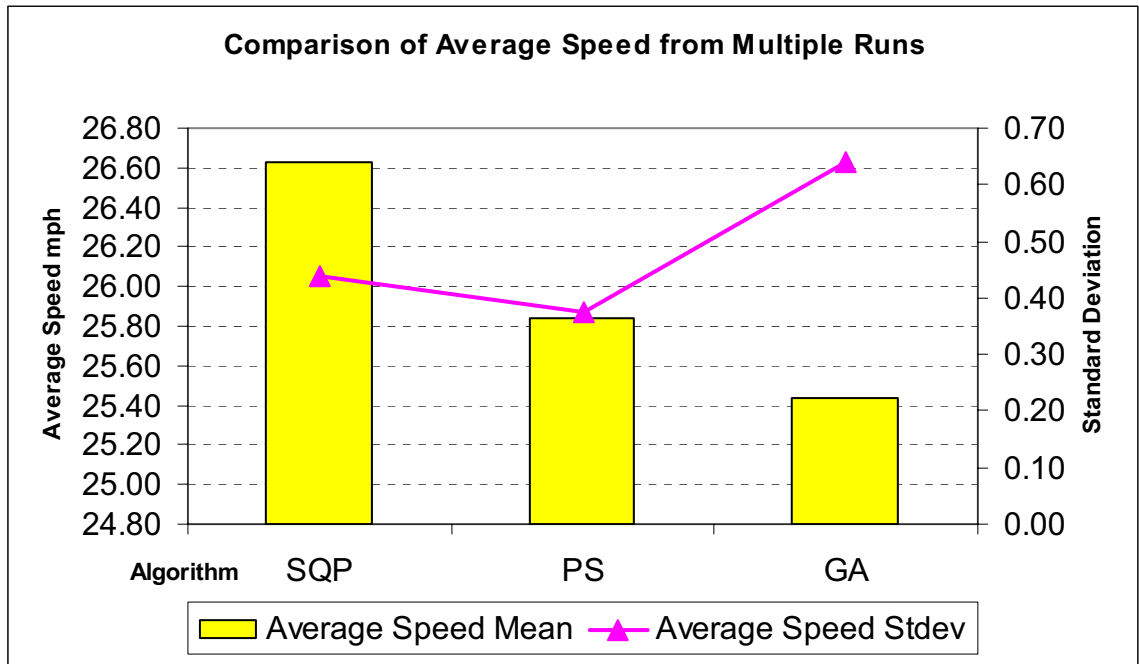


Figure 34 Statistical Average Speed Comparisons among the Three Algorithms

Execution Time

The three candidate algorithms are all implemented in case studies. Their execution times from different scenarios are important indicators for further system implementation in the real world.

Execution time from single runs of both CORSIM and CORAIM RTE are collected and listed in the Table 15. The SQP algorithms have the shortest average execution time (260 seconds for eight MATLAB optimization calls). Because PS and GA will have different execution time given different combination of algorithm options, the execution times for PS and GA are hard to compare if options are not comparable. For example, the pattern size tolerance and polling method will affect the speed of PS while the crossover and mutation probability will affect the GA execution time. Searching for the best option of

these two algorithms deserves to be a new optimization problem. Since the SQP already showed the dominant execution speed advantage, it is not necessary to work on this option study in this research, though it deserves study in the future.

The execution time of each model run is also observed and the results are shown in Table 14. Each type of model run (signal optimization only, DMS level and signal optimization and system optimization) also proves that the SQP has the fastest execution speed.

Table 14

Running Time Comparison on Different Optimization Execution Type

Algorithm	T (Seconds)	t (Seconds)	t_sig (Seconds)
SQP	39.3	30.3	20.1
PS	39.7	36.8	34.2
GA	53.9	49.8	25.1

T: Calculation period, all variables optimization; t_sig, signal roll period, signal parameters optimization only; t, DMS roll period, all variables optimization as well as new solution delay decrease check for the DMS message change.

The CORSIM SQP RTE execution time (including optimization) from multiple runs varies more than the mere CORSIM simulation time. Table 16, Table 17 and Figure 35 show the execution time from multiple runs of CORSIM, SQP RTE and their comparison.

Roughly, the execution time of CORSIM with RTE simulation is more than double of the simulation time of CORSIM only (Table 16 and Table 17). The time increase is from RTE and MATLAB execution. On the other hand, the variance of execution time of CORSIM with RTE is bigger than that of

CORSIM only. The reason for this is that the RTE and MATLAB algorithm execution time varies much more than CORSIM simulation only. Therefore, when CORSIM is run with RTE, the variance of execution time is dominated by the RTE's characteristics.

In Figure 35, the CORSIM standard deviation shows that the CORSIM execution time is fairly stable, even its execution time increases with the traffic load. However, the RTE execution time is not obviously affected by the traffic situation. The moderate scenarios' standard deviation of SQP RTE is the smallest among these scenarios, which indicates that the SQP solution randomness is smaller for the moderate traffic conditions. The SQP solution for moderate traffic has the greatest reduction in delay. The statistics from the fifty runs prove that the SQP has the minimal execution time and minimal standard deviation among the three algorithms (Table 19). Once again, the SQP is proved the best one among the three candidate algorithms by execution time.

Table 15

Execution Time Analysis of the Three Algorithms*

Alg	Scenarios	T%	t	t sig	t	t sig	T	t sig	T	t sig	T	t sig	T	MATLAB Total	Computation	RTE	CORSIM	RTE+CORSIM
SQP	Heavy congestion	0	50	20	29	20	26	20	26	20	26	20	0	211	15.6	226.6	400.4	627.0
	Moderate congestion	0	59	20	29	20	29	20	29	20	26	20	0	223	47.4	270.4	254.1	524.5
	Light congestion	0	47	20	41	20	38	20	32	20	32	20	0	238	53.0	291.0	179.9	470.9
	Heavy congestion	30	53	20	23	20	23	20	26	20	26	20	0	205	37.5	242.5	394.9	637.3
	Moderate congestion	20	35	20	32	20	29	20	29	20	29	20	0	205	49.7	254.7	252.0	506.6
	Light congestion	20	50	23	35	20	32	20	35	20	35	20	0	235	41.8	276.8	178.7	455.5
PS	Average		49	21	32	20	30	20	29	20	29	20	0	220	41	260	277	537
	Heavy congestion	0	67	35	38	38	35	38	35	38	35	39	0	325	32.6	357.6	400.4	757.9
	Moderate congestion	0	47	26	38	38	26	32	41	38	0	0	0	286	45.9	331.9	254.1	586.0
	Light congestion	0	35	23	35	38	35	38	41	29	0	0	0	274	59.3	333.3	179.9	513.2
	Heavy congestion	30	41	38	35	38	32	38	35	29	0	0	0	286	41.9	327.9	394.9	722.7
	Moderate congestion	20	50	26	38	38	38	23	32	38	0	0	0	283	56.3	339.3	252.0	591.2
GA	Light congestion	20	35	26	35	38	35	38	38	38	0	0	0	283	75.7	358.7	178.7	537.4
	Average		46	29	37	38	34	35	37	35	0	0	0	290	52	341	277	618
	Heavy congestion	0	62	23	47	23	47	23	47	23	47	23	0	295	39.6	334.59	400.4	734.9
	Moderate congestion	0	72	23	47	23	47	23	47	23	47	23	0	305	56.5	361.53	254.1	615.6
	Light congestion	0	62	23	47	23	47	23	47	23	47	23	0	295	57.9	352.86	179.9	532.7
	Heavy congestion	30	47	26	58	36	60	36	63	41	0	0	0	367	45.8	412.77	394.9	807.6
	Moderate congestion	20	62	23	50	23	47	23	50	23	0	0	0	301	74.3	375.3	252.0	627.3
	Light congestion	20	47	23	47	26	47	23	47	23	47	23	0	283	51.9	334.9	178.7	513.6
	Average		59	24	49	26	49	25	50	26	0	0	0	308	54	362	277	639

* In seconds. T: Calculation period, t_sig, signal roll period, t DMS roll period.

Table 16

Multiple SQP RTE Running Time

RTE Running Time	Heavy Congestion, NO Base Turning % (Seconds)	Heavy Congestion, 30% Initial Turning (Seconds)	Moderate Congestion, NO Base Turning % (Seconds)	Moderate Congestion, Initial Turning (Seconds)	Light Congestion, NO Base Turning % (Seconds)	Light Congestion, 20% Initial Turning (Seconds)
run 1	380.33	417.84	344.78	363.72	342.96	321.58
run 2	392.76	416.05	382.7	401.78	374.62	339.76
run 3	412.44	435.87	398.34	366.02	427.69	359.62
run 4	409.91	401.91	406.97	411.27	366.56	355.42
run 5	395.86	471.07	369.95	404.36	381.36	364.31
run 6	408.69	444.68	392.12	384.83	397.86	350.18
run 7	397.36	529.68	398.84	400.86	405.22	381.94
run 8	400.5	513.87	385.71	388.5	394.44	377.62
run 9	558.27	407.18	381.78	391.25	388.46	346.37
run 10	393.12	409.56	369.95	390.43	408.07	385.24
Average	414.92	444.77	383.11	390.30	388.72	358.20
Stdev	51.27	45.70	18.12	15.67	23.85	20.01

Table 17
Multiple Running Time of CORSIM

CORSIM	Heavy Congestion, NO Base	Heavy Congestion, 30%	Moderate Congestion, NO	Moderate Congestion, 20%	Light Congestion, NO	Light Congestion, 20%
run 1	214.35	215.5	116.44	116.75	69.89	71.8
run 2	216.05	210.41	117.44	112.45	68.84	73.07
run 3	210.89	213.23	117.94	113.54	70.59	70.4
run 4	209.27	204.42	120.53	119.13	72.68	71.11
run 5	211.41	211.87	127.76	121.6	71.86	70.56
run 6	212.51	209.00	118.73	123.52	74.1	69.42
run 7	219.05	208.17	111.81	115.33	72.46	70.91
run 8	218.97	208.49	115.16	114.72	71.2	67.7
run 9	210.43	206.66	114.47	116.43	69.88	72.61
run 10	214.56	205.66	115.33	115.36	70.05	70.49
run 11	208.17	213.97	113.45	115.31	73.28	70.05
run 12	212.42	217.63	115.08	119.47	73.59	73.25
run 13	228.27	210.78	112.13	113.5	70.05	71.17
run 14	212.3	210.83	117.85	118.23	72.91	71.79
run 15	210.84	208.43	115.76	113.72	72.11	73.19
run 16	208.35	213.42	115	116.91	72.35	69.49
run 17	209.42	211.68	115.75	117.12	70.18	71.44
run 18	210.04	207.92	113.28	122.21	73.67	71.23
run 19	208.77	212.75	114	119.89	67.1	71.37
run 20	212.36	225.49	116.02	120.12	72.95	73.42
Average	212.92	211.32	116.20	117.27	71.49	71.22
Stdev	4.80	4.71	3.49	3.16	1.85	1.46

Table 18

Changes from Diversion Volume to Diversion Level Approximation

Scenarios	Algorithms	Demand (vph)	Diversion Volume (vph)	Diversion %	Diversion Level	Cycle Length (Seconds)	Total RTE Model (Veh-Hour)	Freeway Delay (Veh-Hour)	Detour Delay (Veh-Hour)	Control Delay (Veh-Hour)	Total Delay Change %
Moderate 00% diversion	SQP	Diversion Volume	2896	34%		125	4755.24	3096.72	912.52	746.00	
		Level Approximation	2896	30%	4	90	4836.57	3197.35	922.92	716.29	1.71%
	GA	Diversion Volume	2896	39%		111	4658.34	4231.08	-281.76	709.02	
		Level Approximation	2896	40%	5	51	5008.79	4195.82	-288.82	1101.79	7.52%
	PS	Diversion Volume	2904	15%		140	4727.79	4822.25	-389.89	295.43	
		Level Approximation	2904	20%	3	163	3578.10	3149.24	-38.70	467.56	-24.32%
Moderate 20% diversion	SQP	Diversion Volume	2896	34%		125	4765.14	3109.47	909.27	746.40	
		Level Approximation	2896	30%	4	150	4828.42	3217.69	920.88	689.84	1.33%
	GA	Diversion Volume	2888	1%		48	12446.41	12395.35	-31.16	82.22	
		Level Approximation	2888	1%	1	53	12427.33	12393.01	-39.88	74.20	-0.15%
	PS	Diversion Volume	2896	16%		240	2785.14	2378.73	-23.33	429.74	
		Level Approximation	2896	20%	3	144	3573.69	3151.55	-19.46	441.61	28.31%

Table 19

Statistical Comparison of Execution Time among the Three Algorithms

Scenario moderate with 20% diversion	Mean (Seconds)	Stdev (Seconds)
SQP 50 run time	375.11	15.38
PS 50 run time	719.23	50.45
GA 50 run time	480.91	23.15

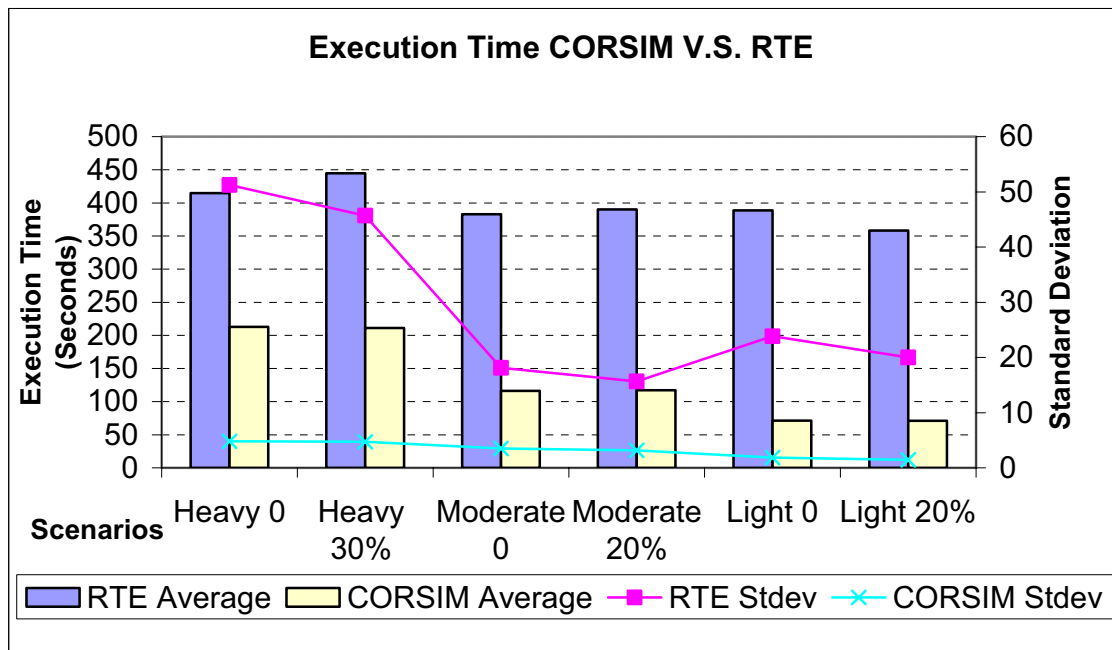


Figure 35 Average Execution Time Comparisons between CORSIM and SQP RTE

Diversion Volume to DMS Level Approximation Effects

Because the algorithms can only deal with real continuous numbers, the model cannot output the DMS level directly. As an alternative implementation method, the model first returns the diversion volume and then runs the approximation function to find a DMS level closest to a given diversion probability. This approximation results in a different diversion volume and the original solution for signal is not applicable anymore. Running the optimization model again is necessary to find a new signal plan given the new diversion volume caused by the DMS message level with all the same intersection volumes. These two different optimizations result in different CORSIM MOE outputs.

The total delays from these two optimizations are compared in the Table 18 For the two evaluated scenarios, SQP yields a slight increase of total delay for both of the scenarios (1.71 percent and 1.33 percent), while the PS and GA generate one increase and one decrease of total delay. One scenario from PS gets a reduction of 24.32 percent in total delay and the other one gets an increase of 28.31 percent in total delay, which indicates that the PS algorithm has higher randomness for these two scenarios.

Application of Penalty Factor

As stated in Chapter Three, the purpose of penalty factor is to guide the algorithm to avoid the infeasible solution area to save search time. However, the penalty factor does not work very effectively in the case studies; it does not save too much execution time. For example, in one of the pattern search scenarios, it only saves a function value evaluations from 2430 to 2411 and execution time from 50.51 to 48.96 seconds. It provides approximately 0.78 percent time saving in function value evaluation and 3.1

percent saving in execution time. Since the saving is not significant, this penalty factor method is not recommended for the final model implementation. It is only treated as an attempt to improve execution speed. Figure 36 and Figure 37 show the execution progresses of PS with penalty factors and without penalty factors. Figure 36 shows that there is an obvious function value drop around iteration 110. The drop span is the total penalty factor value (for example, 1000 veh-hours for standing queue on freeway in the case study, 500 veh-hours for lane group's V/C ratio >1 on diversion path). The jump indicates that Pattern Search finds solutions to avoid assigning more volume to congested areas, for example, freeway with queue, and alternative road on-diversion-path lane groups with $V/C >1$. As a contrast, in the Figure 37, there is not a function value drop without penalty factor employed.

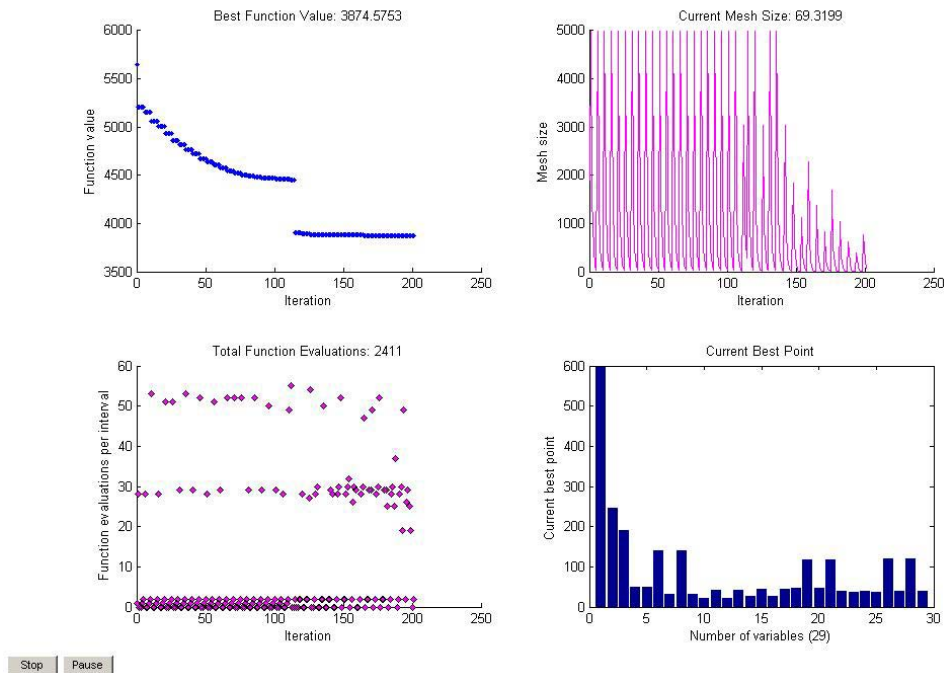


Figure 36 Illustration of Pattern Search Optimization Progress with Penalty

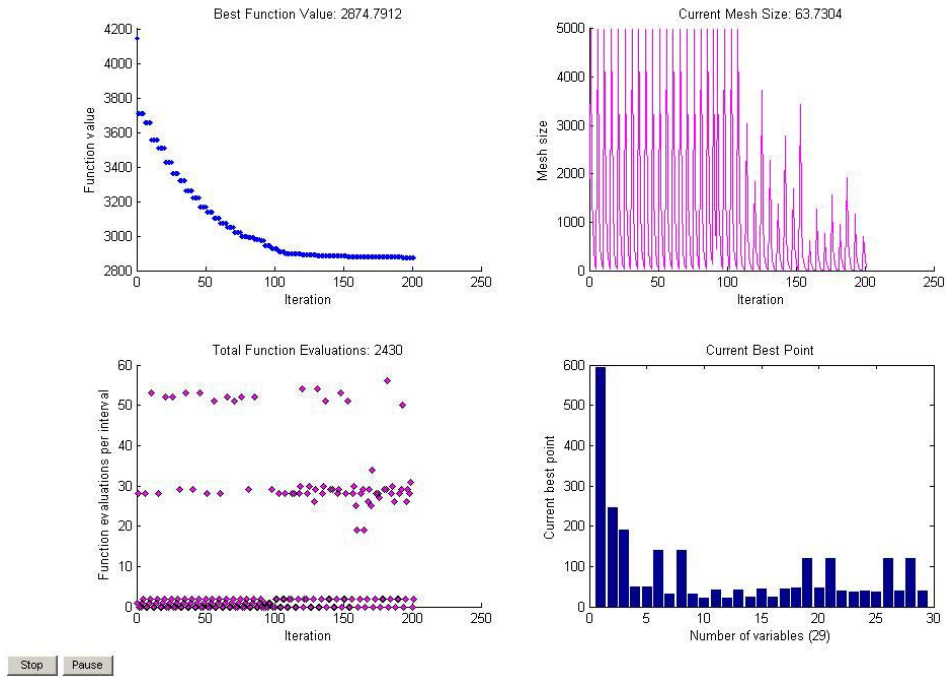


Figure 37 Illustration of Pattern Search Optimization Progress without Penalty Factor

Application of Analytical Difference in SQP

Theoretically speaking, the analytical gradient method will not significantly improve the quality of solution because the finite difference precision in *fmincon* (SQP) is 10^{-6} . However, it does improve the algorithm execution speed and shorten the execution time. The numerical results show that average function value evaluations and execution time in analytical difference method changes dramatically – only 6.69 percent and 18.13 percent of those in numerical differential method (finite difference, Table 20).

Table 20

Comparison between Finite Difference and Analytical difference Gradient in MATLAB SQP Execution Properties

Traffic Condition Cases		Case A (Sec)	Case B (Sec)	Case C (Sec)	Case D (Sec)	Case E (Sec)	Case F (Sec)	Average
Finite difference	Iterations	60	96	83	58	79	52	99.21%
Analytical difference		62	95	85	54	77	52	
Analytical/Numerical		103.33%	98.96%	102.41%	93.10%	97.47%	100.00%	
Finite difference	Function value evaluated	1835	2919	2526	1774	2407	1590	6.69%
Analytical difference		131	193	172	112	161	105	
Analytical/Numerical		7.14%	6.61%	6.81%	6.31%	6.69%	6.60%	
Finite difference	Execution time (Seconds)	53.89	73.29	62.91	44.11	59.05	39.32	18.13%
Analytical difference		9.54	13.56	11.38	8.21	10.43	7.15	
Analytical/Numerical		17.70%	18.50%	18.09%	18.61%	17.66%	18.18%	

However, the RTE execution time of a three-hour simulation does not show time-saving as much as the single MATLAB model runs. The finite difference of SQP RTE runs 375 seconds with a standard deviation of 15 seconds, while the analytical difference of SQP RTE has 321 seconds execution time with a

standard deviation of 12 seconds from forty CORSIM runs. It has only 14.4 percent saving on execution time. The execution time data implies that the model execution time does not take the major part of the RTE execution time. On the other hand, CORSIM simulation takes the major execution time and the MATLAB program start time ranks the second in the execution time. At the same time, disk data exchange also takes a certain portion of execution time. All these time consumptions beyond model execution imply that improvements can be achieved either from MATLAB program start time improvement (such as MATLAB background engine technique) or from other optimization libraries that can be called directly by RTE.

Summary

This chapter introduces the numerical example result analysis and system evaluation. Example network setup, cases and scenarios design, CORSIM running control, variables, constraints, RTE configurations for case implementations are introduced first. Then, model validation is described. Finally, result analysis about CORSIM MOE outputs, system execution time is described.

In addition, some techniques such as approximation of diversion volume to DMS Level, application of penalty factor in objective function and application of analytical difference in SQP algorithm employed during the numerical implementation are also introduced.

The SQP algorithm is proven the best one among the three candidate algorithms: SQP, GA, and PS for the proposed model. Model solution from SQP algorithm shows a maximum reduction of 14.09

percent in delay if compared with TRANSYT-7F's solution. Therefore, the model is effective in reducing corridor delay.

CHAPTER VI

CONCLUSIONS AND FUTURE RESEARCH

Conclusions

This dissertation proposed a DMS diversion control system. DMS and alternative road traffic signals are the two types of control elements in this system. The objective of this model is to reduce the total corridor delays, which includes freeway travel delays, arterial signal control delays and the travel time difference between mainline and alternative road. The objective function is a nonlinear function. Therefore, only algorithms that can solve nonlinear problems were considered.

Nonlinear constraints do not exist for this model. Constraints are mainly from traffic signal split regulations and limitations. For example, green times should be the same for the lane groups in a same phase, and total green times should be less than the cycle length and so on. Lower bounds and upper bounds of independent variables (for example, diversion volume and signal timings) are set up because the independent variables have physical meanings that need acceptable values in the real world.

Three algorithms, SQP, Pattern Search and Genetic Algorithm, from MATLAB Optimization Toolbox and Genetic Algorithm and Direct Search Toolbox were initially selected to solve the model. After evaluation of solution quality and execution speed from the numerical examples, the SQP algorithm is recommended for system implementation.

CORSIM simulation is employed to provide data for model and solution evaluation. A CORSIM RTE and a mainframe program are developed in C++ to exchange the traffic network state data and model solution between CORSIM and MATLAB. At the same time, the RTE also serves to update CORSIM controls with the model solution.

Dataset for model solution can be either collected from detectors deployed along the study area or extracted from traffic simulation. In this dissertation, the CORSIM simulation is employed as a data source for the proposed model. Totally, ten detector stations are set up for the numerical example freeway traffic data collection. For the alternative road, the NETSIM link property data is exported instead of collecting from detectors for the numerical examples. All the data is extracted by RTE and written into a data file for exchanging with MATLAB.

Model validation is conducted by comparing CORSIM MOE outputs between the model and the TRANSYT-7F solutions. TRANSYT-7F CORSIM Processing Method is adopted for generating the base case signal timing plan the proposed model optimization. TRANSYT-7F codes its optimized signal timing plan back into the CORSIM file, which will be run to get CORSIM MOEs for later comparison. Since the proposed model solution over CORSIM simulation is an on-line optimization, the simulation is updated with the newest solution right after it was found, it is not necessary to rerun the CORSIM simulation for MOEs.

In order to make the model evaluation more comparable and reliable, CORSIM MOEs, average network speed and network total delay are selected to compare the before-and-after case performances. The network in this dissertation specifically refers to a freeway corridor area where the DMS and arterial signal

can influence. The arterial road signal-timing plan in base case is optimized by TRANSYT-7F signal optimization software. Multiple simulation runs were conducted to overcome the random nature in CORSIM and the local optimal solution property of the evaluated algorithms.

Results showed that the proposed model could improve the overall corridor capacity utilization. The degree of improvement depends on the algorithm solution qualities. The best algorithm among the three tested algorithms is the SQP algorithm that has the fastest execution speed, best solution quality and best solution uniformity (after diversion volume to integer DMS level approximation, still gives fairly close solution to the original one). Unlike the PS and GA, SQP does not have that many options necessarily to be set up for a good solution, which gives the SQP solution higher certainty. In contrast with Pattern Search and Genetic Algorithm, SQP can have a even faster execution speed given analytical difference gradient.

SQP solutions can reduce the average total delay under studied scenarios from 8.97 percent to 14.09 percent and increase network average speed up to 10.76 percent. For the heavy and low congestion situations, the total delay decreases were less than the moderate congestion situation. This indicates that the proposed model and its solution algorithm works better under moderate congestion situation.

Execution time of the MATLAB model module is less than two minutes, which is acceptable for a real-world implementation of a minimal two-minute traffic signal roll period. The developed RTE can accommodate different network configurations, for example, different total number of intersections, and different diversion paths and so forth. Because the algorithm is coded in MATLAB instead of RTE itself, new algorithm can be introduced for the model solution without any changes in the RTE that needs compiling after code change. This will give the system good expandability.

Because the SQP algorithm cannot guarantee global optimal solution, good quality of initial independent variable values is critical for final solution quality and system performance.

Future Research

Because of restraints from assumptions and research scope, some areas are not covered in this dissertation. In order to verify the performances of the proposed system under more complicated situations, some further studies are considered to improve the system performance and make it more reliable and flexible for the real-world implementation. The followings are the possible studies in the coming future:

1. Since the starting of full MATLAB program takes longer than the MATLAB background engine does. Running MATLAB engine instead of full MATLAB program, compiling MATLAB codes into executable codes and merging with RTE codes, or employing other available SQP library will save model execution time. In addition, data exchange between CORSIM RTE and MATLAB in computer memory will save hard disk data exchange time and thus will improve execution efficiency.
2. If the algorithm is embedded in the RTE, it will greatly save the data exchange time and improve the system's performance eventually. Without MATLAB, the CORSIM RTE implementation will be more applicable in the real world from easier system configuration and higher execution speed.
3. Because of the complexity of driver compliance with the DMS message and the difficultness

of field data collection, data from other studies are referred to. Driver compliance is assumed to follow Extreme Value distribution in this dissertation. In the future, more field compliance data toward uniform message (level) definition might be available and can be more valuable for this research.

4. DMS messages and level classification need to be studied in depth. Results from study should provide useful information to answer questions such as how many levels should be set up, and how much diversion volume should be expected from a message and so on.
5. On-ramp meter is not included in the system because the priority area of this study is set before the bottleneck area and the off-ramp. Since ramp metering is an effective and frequently used control element in freeway corridor control, including of on ramp meter might possibly improve the system performance.
6. Signal offset is not included in the objective function; therefore, it cannot be optimized directly. An improvement is to merge the signal-offset variable into the control delay model and eventually to overcome the control delay computation deficiency of the current model.
7. Model improvements for more complicated cases: 1). More than one exit for a single alternative road; 2). More than one alternative road with only one exit; 3). More than one alternative road with more than one exit. The third one is the most complicated case, which is similar to dynamic assignment over a network with a focus on DMS control objective.

REFERENCES

1. U. S. Department of Transportation and Federal Highway Administration, “Facts & Statistics”, http://ops.fhwa.dot.gov/wz/resources/facts_stats.htm, Accessed in Mar. 2008.
2. U. S. Department of Energy, “Temporary Losses of Highway Capacity and Impacts on Performance”, Oak Ridge National Laboratory, ORNL/TM-2002/3. May 2002.
3. U. S. Department of Transportation and Federal Highway Administration, G. Ullman, “Characteristics of Today’s Work Zones”, Transportation Research Board Annual Meeting, Washington, D.C., January 2004.
4. B. Cronin, S. Mortensen, and D. Thompson, “Integrated Corridor Management”, Public Roads, Vol. 71 No. 5 March/April 2008.
5. Research and Innovative Technology Administration and U.S. Department of Transportation, “ITS Benefits Database, Automated Work Zone Information System (AWIS), California”, <http://www.benefitcost.its.dot.gov/ITS/benecost.nsf/bylink/BOTM-July2006>, accessed in Mar. 2008.
6. S. Peeta, J. L. Ramos, and R. Pasupathy, “Content of Dynamic Message Signs and On-Line Driver Behavior”, Transportation Research Record 1725, 2000, P102-108.
7. B. G. Benson, “Motorist Attitudes about Content of Variable-Message Signs”, Transportation Research Record 1550, Washington, D.C., 1996, P48-57.
8. A. Polydoropoulou, M. Ben-Akiva, A. Khattak, and G. Lauprete, “Modeling Revealed and Stated En-Route Travel Response to Advanced Traveler Information Systems”, Transportation Research Record 1537, 1996.
9. V. Lachhwani and A. J. Horowitz, “Criteria for Portable ATIS in Work Zones”, <http://www.ctre.iastate.edu/smartwz>, University of Wisconsin – Milwaukee, accessed in Mar 2008.
10. The Scientex Corp., “ADAPTIR System”, <http://www.scientexcorp.com/ad-main.htm>, accessed in Mar 2008.

11. P. McCoy, "Adaptir", Univ of Nebraska, Lincoln, 2000,
<http://www.matc.unl.edu/research/MwSWZDI/>, accessed in Mar. 2008.
12. Quixote, "the Future of Work Zone Safety Is the IntelliZone System",
<http://www.quixtrans.com/intellizone.htm>, accessed in Mar. 2008.
13. A. J. Horowitz and T. Notbohm, "Evaluation of IntelliZone: A System for Providing Speed Advisories to Drivers Entering Work Zones", University of Wisconsin, Milwaukee, 2003.
14. T. Notbohm, A. Drakopoulos, and A. J. Horowitz, "Smart Work Zone Deployment Initiative Summer 2001 Travel Time Prediction System (TIPS)", Wisconsin Department of Transportation, 2001.
15. H. T. Zwahlen and A. Russ, "Evaluation of the Accuracy of a Real-Time Travel Time Prediction System in a Freeway Construction Work Zone", Transportation Research Board Annual Meeting, Washington DC 2001.
16. PDP Associates, "Traffic Information & Prediction System TIPS",
http://www.pdpassociates.com/tips_introduction.htm, accessed in Mar 2008.
17. P. Pant, "A Portable Real-Time Traffic Control System for Highway Work Zones", university of Cincinnati, FHWA/oh-2000/011, 2000.
18. G. Pesti, "Brown Real-Time CMS Control, and Iteris Wireless Detection", Texas Transportation Institute, 2002.
19. ASTI Transportation Systems, Inc., "Computerized Highway Information Processing System",
<http://www.asti-trans.com/software.htm>, accessed in Mar. 2008.
20. L. Chu, "Evaluation of Automatic Work Zone Information Systems",
<http://www.its.uci.edu/~lchu/workzone.html>, accessed in Mar 2008.
21. L. Chu et al., "Evaluation of Effectiveness of Automated Work zone Information Systems", Transportation Research Board Annual Meeting, Washington, D.C., January 2005.
22. L. H. Tudor, A. Meadors, R. Plant, "Deployment of Smart Work Zone Technology in Arkansas", Transportation Research Board Annual Meeting, Washington, D.C., January 2002.
23. E. Lee, and C. Kim, "Automated Work Zone Information System AWIS on Urban Freeway Rehabilitation: California Implementation", Transportation Research Board Annual Meeting, Washington, D.C., January 2006.
24. G. Pesti, P. T. McCoy, M. D. Meisinger, and V. Kannan , "Work Zone Speed Advisory System", Texas Transportation Institute, 2002.

25. Z. Guo and K. K. Srinivasan, "Investigating the Effect of User Behavior Factors and Transportation Control Measures on Day-to-day Network Evolution and Trip-time Reliability", Transportation Research Record 1926.
26. L. H. Tudor, A. Meadors, and R. Plant, "Deployment of Smart Work Zone Technology in Arkansas", Transportation Research Board Annual Meeting, Washington, D.C., January 2002.
27. C. Diakaki, M. Papageorgiou, and T. Mclean, "Application and Evaluation of the Integrated Traffic-Responsive Urban Corridor Control Strategy IN-TUC in Glasgow", Transportation Research Board Annual Meeting, Washington, D.C., January 1999.
28. Texas Department of Transportation, "Travel Time Messaging on Dynamic Message Signs - Houston, TX", http://ops.fhwa.dot.gov/publications/travel_time_study/houston/houston_ttm.htm, accessed in Mar. 2008.
29. R. Bushman and C. Berthelot "Estimating the Benefits of Deploying Intelligent Transportation Systems in Work Zones", Transportation Research Board Annual Meeting, Washington, D.C., January 2004
30. M. Vasudevan, K. Wunderlich, J. Larkin, and A. Toppen, "Comparison of Mobility Impacts on Urban Commuting Between Broadcast Advisories and Advanced Traveler Information Services", Transportation Research Board Annual Meeting, Washington, D.C., January 2005.
31. S. Tignor et. al, "Innovative Traffic Control Technology and Practice in Europe", U.S. DOT, FHWA-PL-99-021, August 1999.
32. F. Logi and S. G. Ritchie, "Development and Evaluation of a Knowledge-Based System for Traffic Congestion Management and Control", Transportation Research Part C, Vol 9, No. 6, P443-459.
33. B. Ran, B. Barrett, and E. Johnson, "Evaluation of Dynamic Message Signs in Wisconsin: Driver Survey", WISCONSIN DOT, 0092-45-17, October 2004, <http://www.dot.wisconsin.gov/library/research/docs/finalreports/45-17variablemessagesigns.pdf>, accessed in Mar 2008.
34. L. R. Dumke and T. E. Doyle, "Intelligent Transportation Systems in Work Zones: Leveraging the Internet and Wireless Communications", paper presented at the 11th Annual ITS America Meeting. Miami, Florida, 2001.
35. H. S. Mahmassani and R. Jayakrishnan, "System Performance and User Response under real-time information in a congested traffic corridor", Transportation Research Part A. Vol.25, No.5, 1991, P293-307.
36. Y. Yim and J. Ygnace, "Link Flow Evaluation Using Loop Detector Data: Traveler Response to Variable-Message Signs", Transportation Research Record 1550, 1996, P58-64.

37. A. J. Khattak, "Comparative Analysis of Spatial Knowledge and En Route Diversion Behavior in Chicago and San Francisco: Implications for ATIS". Transportation Research Record 1621, 1998, P27-35.
38. F. Kim, S.G. Barfield, W., and L. Ng, "Statistical Analysis of Commuters' Route, Mode, and Departure Time Flexibility", Transportation Research Part C, Vol. 2, 1994, P35-47.
39. K. Chatterjee, N.B. Hounsell, P.E. Firmin, and P.W. Bonsall, "Driver Response to Dynamic Message Sign Information in London". Transportation Research Part C, Vol. 10, 2002, P149-169.
40. A. J. Horowitz, I. Weisser, and T. Notbohm, "Diversion from Rural Work Zone Owing to Traffic-Responsive Dynamic Message Signage System", Transportation Research Board Annual Meeting, Washington, D.C., January 2003
41. H. Zwahlen and A. Russ, "Evaluation of the Motoring Public's Acceptance of a Real-Time Travel Time Prediction System in a Freeway Construction Work Zone", Transportation Research Board Annual Meeting, Washington, D.C., January 2002.
42. A. J. Miller, "A Computer Control System for Traffic Networks", Proceedings of the second International Symposium on Theory of Road Traffic Flow, Paris, 1965.
43. A. J. Miller, "A Computer Control System for Traffic Networks", Proceedings of the Second International Symposium of the Theory of Traffic Flow, London, 1963.
44. J. D. Zijverden and H. Kwakernaak, "A New Approach to Traffic-Actuated Computer Control of Intersections, t1 Proceedings of the 4th International Symposium on Theory of Traffic Flow, Bonn, Germany, 1969.
45. K. Bang and L. Nilsson, "Optimal Control of Isolated Traffic Signals Control in Transportation Systems", Proceedings of the IFAC/IFIP/IFORS Third Imitational Symposium, Columbus. Ohio, August 9-13, 1976.
46. K. Bang and L. Nilsson, "Optimal Control of Isolated Traffic Signals", Proceeding of Australian Road Research Board, Vol. 8, 1976.
47. R. B. Grafton and G. F. Newell, "Optimal Policies for the control of an under saturated Intersection", Proceedings of the third International Symposium on Theory of Traffic Flow, New York, 1967.
48. O. L. Robertson and R. D. Bretherton, "Optimum Control of an Intersection for any Known Sequence of Vehicle Arrivals, IS Traffic Control and Transportation Systems", Proceedings of the 2nd IFAC/IFIP/IFORS Symposium, Monte Carlo, September 1974.

49. I. Day, "SCOOT - Split, Cycle & Offset Optimization Technique", Transportation Research Board, Adaptive Traffic Signal Control Workshop, Pacific Grove, CA, July, 1998.
50. P. B. Hunt, Robertson, and R.D. Bretherton, "The SCOOT On-line Traffic Signal Optimization Technique", Traffic Engineering and Control, April 1982.
51. N.R. Gross, "SCATS Adaptive Traffic System" Adaptive Traffic Control Workshop, Pacific Grove, CA, July 1998.
52. P. Mirchandani and L. Head, "The RHODES Prototype: A Technical Review", Gardner Systems, University of Arizona, Transportation Research Board A3A18 mid Year Meeting and Adaptive Traffic Signal Control Workshop, Pacific Grove, CA, July, 1998 .
53. N. H. Gartner, "OPAC: A Demand-Responsive Strategy for Traffic Signal Control", Transportation Research Record. 906, 1983.
54. Y. Luo and G. Chang, "Exploring the Robust Signal Optimization Method for Pre-timed Control at Isolated Intersections", Transportation Research Board Annual Meeting, Washington, D.C., January 2002.
55. J. Lee, E. Chung, B. Abdulhai, and A. Shalaby, "Real-Time Optimization for Adaptive Traffic Signal Control Using Genetic Algorithms", Transportation Research Board Annual Meeting, Washington, D.C., January 2004.
56. B. Park and I. Yun, "Evaluation of Stochastic Optimization Methods of Traffic Signal Control Settings for Coordinated Actuated Signal Systems", Transportation Research Board Annual Meeting, Washington, D.C., January 2006.
57. D. Hale, "Traffic Network Study Tool – TRANSYT-7F, United States Version", Mctrans Center in the University of Florida, Gainesville, January 2006.
58. D. Hale, "Traffic Network Study Tool – TRANSYT-7F, United States Version", Mctrans Center in the University of Florida, Gainesville, January 2006.
59. http://www.mti.umd.edu/people/mahmassani/research/Research_descriptions_HSM_website_10%2004.pdf, accessed in Mar 2008.
60. Y. Zhang, "Optimal Traffic Control for a Freeway Corridor under Incident Conditions", dissertation, Virginia Tech, October 1996.
61. A. D. May, "Traffic Flow Fundamentals", Prentice Hall, New Jersey, 1990.
62. Y. Jiang, "Estimation of Traffic Delays and Vehicle Queues at Freeway Work Zones", Transportation Research Board, Washington, D.C., 2001

63. M. V. Aerde, "Single regime speed-flow-density relationship for congested and uncongested highways", Transportation Research Board Annual Meeting, Washington, D.C., 1995.
64. S. Houde, E. Safirova, and W. Harrington, "Washington START Transportation Model, Description and Documentation", Washington, RFF DP 07-43, 2007.
65. S. Erlingsson, A. M. Jonsdottir, and T. Thorsteinsson, "Traffic Stream Modeling of Road Facilities", Transport Research Arena Europe 2006,
[http://vgwww.vegagerdin.is/Vefur2.nsf/Files/2006_TRA_2/\\$file/Van%20Aerde%20Traffic-Stream-SE-AMJ-THTH-2006.pdf](http://vgwww.vegagerdin.is/Vefur2.nsf/Files/2006_TRA_2/$file/Van%20Aerde%20Traffic-Stream-SE-AMJ-THTH-2006.pdf), accessed in Apr. 2008.
66. S. H. Hamdar, S. M. Eisenman, and H. S. Mahmassani, "Evaluation of Operational Strategies for Integrated Corridor Management", Transportation Research Board Annual Meeting, Washington, D.C., 2006.
67. The MathWorks Inc, "Optimization Toolbox™ 4 User's Guide", MathWorks, 2008
68. The MathWorks Inc, "Genetic Algorithm and Direct Search Toolbox™ 2 User's Guide", MathWorks, 2008.
69. FHWA, "CORSIM 6.0 User Manual", FHWA, 2006.
70. C. G. Broyden, "The Convergence of a Class of Double-Rank Minimization Algorithms", Journal Inst. Math. Applic, Vol. 6, P76-90, 1970.
71. W. C. Davidon, "Variable Metric Method for Minimization", A.E.C. Research and Development Report, ANL-5990, 1959.
72. R. Fletcher, "A New Approach to Variable Metric Algorithms", Computer Journal, Vol. 13, P317-322, 1970.
73. R. Fletcher, "Practical Methods of Optimization", Vol. 1, Unconstrained Optimization, John Wiley and Sons, 1980.
74. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization", Computer Journal, Vol. 6, P163-168, 1963.
75. D. Goldfarb, "A Family of Variable Metric Updates Derived by Variational Means", Mathematics of Computing, Vol. 24, P23-26, 1970.
76. D. F. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization", Mathematics of Computing, Vol. 24, P647-656, 1970.

77. M. Momma and K. P. Bennett, “A Pattern Search Method for Model Selection of Support Vector Regression”, <http://www.siam.org/meetings/sdm02/proceedings/sdm02-16.pdf>, accessed in Oct. 2008.
78. D. E. Goldberg, “Genetic Algorithms in Search, Optimization & Machine Learning”, Addison-Wesley, 1989.
79. A. R. Conn, N. I. M. Gould, and Ph. L. Toint. “A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds”, *SIAM Journal on Numerical Analysis*, Volume 28, No. 2, P545–572, 1991.
80. A. R. Conn, N. I. M. Gould, and Ph. L. Toint. “A Globally Convergent Augmented Lagrangian Barrier Algorithm for Optimization with General Inequality Constraints and Simple Bounds”, *Mathematics of Computation*, Volume 66, No. 217, P261–288, 1997.
81. V. Torczon, “On the Convergence of Pattern Search Algorithms”, *SIAM Journal on Optimization*, Volume 7, No. 1, P1–25, 1997.
82. R. M. Lewis and V. Torczon, “Pattern Search Algorithms for Bound Constrained Minimization”, *SIAM Journal on Optimization*, Volume 9, No. 4, P1082–1099, 1999.
83. R. M. Lewis and V. Torczon, “Pattern Search Methods for Linearly Constrained Minimization”, *SIAM Journal on Optimization*, Volume 10, No. 3, P917–941, 2000.
84. C. Audet and J. E. Dennis Jr., “Analysis of Generalized Pattern Searches”, *SIAM Journal on Optimization*, Volume 13, No. 3, P889–903, 2003.
85. R. M. Lewis and V. Torczon, “A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds”, *SIAM Journal on Optimization*, Volume 12, No. 4, P1075–1089, 2002.
86. R. H. Byrd, J. C. Gilbert, and J. Nocedal, “A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming”, *Mathematical Programming*, Vol 89, No. 1, P149–185, 2000.
87. T. F. Coleman and Y. Li, “An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds”, *SIAM Journal on Optimization*, Vol. 6, P418–445, 1996.
88. T. F. Coleman and Y. Li, “On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds”, *Mathematical Programming*, Vol. 67, No. 2, P189–224, 1994.

89. P. E. Gill, W. Murray, and M.H. Wright, "Practical Optimization", London, Academic Press, 1981.
90. S. P. Han, "A Globally Convergent Method for Nonlinear Programming", Vol. 22, Journal of Optimization Theory and Applications, P297, 1977.
91. M. J. D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", Numerical Analysis, Vol. 630, 1978.
92. M. J. D. Powell, "The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations", Nonlinear Programming 3, Academic Press, 1978.
93. R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps", Mathematical Programming, Vol 107, No. 3, P391–408, 2006.
94. R. Fletcher, "Practical Methods of Optimization", John Wiley and Sons, 1987.
95. W. Hock, and K. Schittkowski, "A Comparative Performance Evaluation of 27 Nonlinear Programming Codes", Computing, Vol. 30, P335, 1983.
96. M. J. D. Powell, "Variable Metric Methods for Constrained Optimization", Mathematical Programming: The State of the Art, P288-311, 1983.
97. K. Schittkowski, "NLQPL: A FORTRAN-Subroutine Solving Constrained Nonlinear Programming Problems", Annals of Operations Research, Vol. 5, P485-500, 1985.
98. V. Aerde and H. Rakha. "Multivariate calibration of single regime speed-flowdensity relationships", Proceedings of the vehicle navigation and information systems (VNIS) conference, Seattle, Washington, 1995.
99. P. T. Martin and L. M. Hockaday, "SCOOT - an update", ITE Journal 65. 1995, P44-48.
100. P.B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "SCOOT - a traffic responsive method of coordinating signals", Laboratory Report 1014, Transport and Road Research Lab, Crowthorne, Berkshire, U.K. 1981.
101. B.G Hansen, P. T. Martin, and H. J. Perrin, "SCOOT real-time adaptive control in a CORSIM simulation environment", Transportation Research Record 1727, 2000, P27-31.
102. P. R. Lowrie. "SCATS: The Sydney Co-ordinated Adaptive Traffic System", IEE International conference on road traffic signaling, 1982, London, U.K.
103. K. Hu, "LADOT's adaptive traffic control system", Workshop on Adaptive Signal Control Systems, Transportation Research Board Annual Meeting, Washington D. C., Jan. 2000.

104. C. Bielefeldt and F. Busch, "MOTION - A new on-line traffic signal network control system", the 7th IEEE Conference on Road Traffic Monitoring and Control, 1994.
105. L. Khoudour, J. B. Lesort, and J. L. Farges, "PRODYN - Three years of trials in the ZELT Experimental Zone", Recherche - Transports - Securite, English Issue, No. 6. Special Traffic Management, 1991, P89-98.
106. V. Mauro and D. Taranto, "UTOPIA", the 6th IFAC/IFIP/IFORS Symposium on Control and Communication in Transportation, 1990, Paris, France.
107. R. A. Reiss, "Development of Traffic Logic for Optimizing Traffic Flow in an Intercity Corridor", DOT-FH-II-8738, U. S. Department of Transportation, 1978, Washington D. C.
108. R. A. Reiss, "Algorithm development for corridor traffic control In Traffic", Transportation, and Urban Planning, Vol.2, 1981, George Goodwin, London.
109. N. H. Gartner and R. A. Reiss, "Congestion Control in Freeway Corridors: The IMIS System in Flow Control of Congested Networks", Volume F38, 1987.
110. R. A. Reiss, N. H. Gartner, and S. L. Cohen, "Dynamic Control and Traffic Performance in A Freeway Corridor: A Simulation Study", Transportation Research Part A, Vol.25A, 1991, P267-276.
111. B. Han and R. A. Reiss, "Coordinating ramp meter operation with upstream intersection traffic signal", Transportation Research Board Annual Meeting, 1994, Washington D. C.
112. C. Jacob and B. Abdulhai, "Automated Adaptive Traffic Corridor Control using Reinforcement Learning: Approach and Case Studies", Transportation Research Board Annual Meeting, January 2006 Washington D. C.
113. M. Cremer and S. Schoof, "On control strategies for urban traffic corridors", Proceedings of IFAC Control, Computers, Communications in Transportation, 1989, Paris, France.
114. J. Ma, Y. Nie, and H. M. Zhang, "Solving the Integrated Corridor Control Problem Using Simultaneous Perturbation Stochastic Approximation", Transportation Research Board Annual Meeting, 2007, Washington D. C.
115. G. L. Chang, P. K. Ho, and C. H. Wei, "A dynamic system-optimal control model for commuting traffic corridors", Transportation Research Part C, Vol. 1 1993 , P1-12.
116. N.H. Gartner, "OPAC: A demand responsive strategy for traffic signal control", Transportation Research Record 906, 1983, Washington, D.C.

117. K.L. Head, P. B. Mirchandani, and D. Shepherd, "A hierarchical framework for real-time traffic control", Transportation Research Record 1360, 1992, Washington, D.C. P82-88.
118. P. T. Martin, J. Perrin, B. R. Chilukuri, C. Jhaveri, and Y. Feng, "Adaptive Signal Control II", UT-03.28, University of Utah, 2003.
119. R.A.Vincent and C. P. Young, "Self-Optimizing Traffic Signal Control Using Microprocessors - the TRRL 'MOVA' Strategy for Isolated Intersections", Traffic Engineering and Control 27, No. 7-8. Jul-Aug 1986. P385-387.
120. FHWA, ACS-Lite http://ops.fhwa.dot.gov/acs_lite/index.htm, accessed in Oct. 2008.
121. MitreTek System, "QuickZone Delay Estimation Program Version 0.99, User Guide", Mar. 2001.
122. ITT Industries, Inc., "CORSIM Data Dictionary Version 6.0", April 2005
123. ITT Industries, Inc., "CORSIM Run-Time Extension (RTE) Developer's Guide Version 6.0", November 2005.
124. The Scientex Corporation, ADAPTIR System, <http://www.scientexcorp.com/adaptir.htm>, accessed in Mar. 2008.
125. Quixote Transportation Technologies, Inc., IntelliZone Integrated System, <http://www.qttinc.com/pages/intellizone.html>, accessed in Mar. 2008.
126. Transportation Research Board, Highway Capacity Manual (HCM) 2000, Washington DC. 2002.

APPENDIX A.

PSEUDO CODE OF SYSTEM IMPLEMENTATION


```

// DMS_MAIN.cpp : DMS system main frame
int _tmain(int argc, _TCHAR* argv[])
{
    start_CORSIM(); //RTE will be actived in CORSIM, read/write data
    DMS_RTE; //manual operation
}
//RTE configuration and functions usage
DMS_RTE()
{
    At initialize stage:
        call read_file(); //read in user input file
        call network(); //output network data
    At BeginSimulatin stage:
        call dt_initial_vol(); //keep initial stage detector data
    for (clock < simulation time)
    {
        At EACH Timestepcomplete:
            call time_step_call(); //check to see if it needs to start T, t or t_sig MATLAB run
        At EACH timestep PreNETSIMSignal;
            call signal_update(); //update signal if there is interval data on timeline for this step
        At EACH timestep FRESIMVehicle:
            call path_assign(); //randomly assign path for new generated FRESIM vehicles
    }
    At Exit stage:
        call write_summary(); //output running result and steps' data
}
//read in the user input control file
int read_file()
{
    read in.open ("DMS_in.txt");
//network
    read in total_intersections; //intersections to be optimized on the alternative road
    read in num_interX_property; //number of properties of an intersection
    for all intersection_nodes
        for all intersection properties
            read in all intersection_properties; //Legs, travel time from diversion node
        }
    }
    read in diversion_node#; //diversion node number in CORSIM .trf file, user nodes number
    read in after_diversion_node#; //the nodes number right after diversion node, user nodes

```

```

number
    read in    merge_node#; //merge node number after work zone, user nodes number
    read in    path_nodes1#; //total number of nodes on freeway path, from 8XXX to 8XXX, full path
    read in    path_nodes2#; //total number of nodes on alternative road path, from 8XXX to 8XXX,
including 7XXX nodes.
    read in main_path_nodes;//read in all the nodes on main path to an array
    read in diversion_path_nodes;//read in all the nodes on diversion path to an array
    generate_path(diversion_path); //only the diversion path is for car following, main path is for link
usage
    read in    total_number_of_independent_variables; //29 for example, div_vol, cycle, and each
group green time.
    read in    mainline_vol_station_number,mainline_capacity;//mainline demand detector station
number, and bottle neck capacity
    read in off_vol_station_number, offramp_capacity; //offramp detector number, and its capacity
for total number of independent variables -2
    {
        read in x.order; //order of variable
        read in x.upnode;//link upnode, where the group locates
        read in x.dnnode;//link upnode, where the group locates
        read in x.group; //group number
        read in x.interval;//signal interval number of this group
        read in x.saturation_rate;//saturation rate for this group
    }
//running control
    read in algorithm_idx;//algorithm that will be used 1.fmincon, 2. PS. 3. GA
    read in T; //period of DMS
    read in rolling_h; //rolling horizon of DMS
    read in rolling_h_sig;//rolling horizon of signal
    read in FFS;    //free flow speed of freeway
    read in FFS_d; //free flow speed of diversion road
    read in DMS_Level;//total number of levels of DMS
for all the DMS levels
    {
        read in diversion_rate_desired of each level;//diversion percentage/probability related to each
level.
    }

    read in total_effect_freedetsta;//number of detector stations will be used for freeway speed monitor
for freeway speed monitor detector stations
    {

```

```

        read in freeway_detector station number; //each detector station number will monitor a
section
    }
    read in F0_sta; //detector station number of freeway demand
    read in    Fc_sta; //detector station number of work zone capacity

    read in.close();
}
//write time independent network geometric data
int network()
{
    write.open("network.txt", ios::out);
    for all nodes in the network
    {
        writenode.global_id; //global id in CORSIM
        writenode.network_id; //net type, FRESIM or NETSIM
        writenode.internal_id; //internal id used by NETSIM or FRESIM
        writenode.user_id; //user id, which is from user created network
    }
    for all FRESIM links
    {
        save flinks.link_id; //FRESIM link id
        save flinks.user_up; //upnode of link, user node number
        save flinks.user_down; //downnode of link, user node number
        save flinks.internal_up; //upnode of link, FRESIM node number
        save flinks.internal_down; //downnode of link, FRESIM node number
        save flinks.length; // link length, ft, convert from float to int here.
        save offramp_link; //FRESIM off ramp link, only one offramp for the proposed model
        save diversion_link; //FRESIM link that has the exit (only one)
    }
    for all NETSIM links
    {
        save nlinks.link_id; //NETSIM link id
        save nlinks.user_up; //upnode of link, user node number
        save nlinks.user_down; //downnode of link, user node number
        save nlinks.internal_up; //upnode of link, FRESIM node number
        save nlinks.internal_down; //downnode of link, FRESIM node number
        save nlinks.length; //Length of link, in feet
        save nlinks.L; //left turn %
        save nlinks.Th; // through %
    }
}

```

```

        save nlinks.R;//right %
    }
    for all global links
    {
        save glinks.glk_id;//global link id
        save glinks.up;//up node
        save glinks.down;//down node
        save glinks.net;//network type, 3, FRESIM, 8 NETSIM
        save glinks.netlk_id;//network id
        save glinks.net_up;//upnode, network node id
        save glinks.net_down;//downnode network node id
        save glinks.length;//length, ft
    }
    save main path links vector;//main path links
    save diversion path links vector;//diversion path links
    writeactually highest internal node#;// max number of internal node of CORSIM
    writeMaximum number of global links;// max number of global links
    writetotal freeway nodes;//FRESIM nodes
    writetotal NETSIM nodes;//NETSIM nodes
    writetotal freeway links;//FRESIM links
    writetotal NETSIM links;//NETSIM links
    writetotal NETSIM lk, no interface lk;//NETSIM links without interface link
    writenodes;//total nodes data
    writeglobal links;//global link data
    writeFRESIM links;//FRESIM link data
    writeNETSIM links;//NETSIM link data
    writeoff_ramp link number;//off ramp link number, FRESIM link id
    writeintersection nodes and their approach links;//all the intersection legs' link id and nodes
    writeTotal length of main path, Effective path, Diversion_node distance from 0, merge node dist
from 0;
    writeTotal length of diversion path, Effective path;
    for all detectors in Freesim
    {
        writestation_id;
        writedetector_id;
        writelink_id;//FRESIM link id that detector is on
        writeflane_id;//FRESIM lane id that detector is on
        writedist_upnode; //dist to upnode this link
    }
    writedistance from path main first node;

```

```

for all detector stations
{
    writestation_id;
    writedist_upnode;//distance from upnode
    writeflink_id;//FRESIM link id that detector station is on
    writedist from 0;//FRESIM detector station distance from the first node on main path
}
writetruck percentage of each freeway link;//truck %, for possible PCE calculation
}
//this function is called at each time step in CORSIM. real time monitoring of CORSIM run
int time_step_call() //in sec
{
    if time is multiple times of T
    {
        call_type=1; //this is a T call
        signal_change_turn_sw = 1; //turn on signal change switch because of diversion volume
variance;
        signal_change_delay_sw = 1; //turn on signal change switch because of total delay variance
        turn_change_sw = 1; //turn on path assign switch
        MAIN(call_type); //call the main function of RTE
        first_T_done = 1; //only after finished first T, can start t and t_sig rolling
    }
    //only call type 2 has some check, signal will be updated, but the DMS level depends on compliance
degree.
    if time is multiple times of t && first T done
    {
        call_type=2; //it is a t call
        write_file_periodic();//output data of current period for MATLAB
        if first call
        {
            turn_change_sw = 1; //first run, must change the turn and signal.
        }
        else
        {
            if (driver compliance <= COMPLIANCE_TORLERANCE)
            {
                turn_change_sw = 0; //turn off the turning percentage change because of diversion
volume variance
            }
            else

```

```

        {
            turn_change_sw = 1; //turn on the turning percentage change because of diversion
volume variance
        }
    }
    MAINa(call_type); //part of functions in MAIN.steps after call MATLAB in MAIN.
}
if time is multiple times of t && first T done
{
    call_type=3; //t_sig call
    write_file_periodic(); //output data of current period for MATLAB
    MAINa(call_type); //part of functions in MAIN.steps after call MATLAB in MAIN.
}
}
//take detector initial value, because freeway detector data are from initial period, so, must take the end
of initial vol as the clock0(start) vol
void dt_initial_vol()
{
    for all FRESIM detectors
    {
        save call_time_counter; //call counter increased in save function
        save station_id; //FRESIM detector station id
        save detector_id; //Fresim detector id
        save cum_vol; //Fresim detector cumulated volume until current call
        save cum_speed; //Fresim detector cumulated speed until current call
        save vol; //Fresim detector volume of current call
        save speed; //Fresim detector speed of current call
    }
}
//network state snap data
int write_file_periodic()
{
    call_counter++; //counter of call times
    cm_file_written=0; //reset the file written flag to NO, after all data ready for c_m.txt, it will be set
to 1, then MATLAB can start.
    save path_assigned_vehs; //keep the vehicle number that is assigned diversion path in
veh_path_assign function
    save current call data(call_time_counter, call_type, clock_from0, TP, interval);
    save each FRESIM detector data
    (

```

```

        call_time_counter;
        detector_id;
        station_id;
        cum_vol;//cumulative volume
        cum_speed;//cumulative speed
        speed;
    )
    process and save FRESIM detector station data;
    forecast volume and density of next call; //3 points moving average, except 1st and 2nd call,
    NOforcaste_freespeed(call_counter);//copy the current speed for next call,
forcaste_freespeed(call_counter) will do forecast;
    save each approach traffic data
    (
        call_time_counter;
        nlink_id; //NETSIM link ID
        L_pre;    //cumulative L volume of previous call
        Th_pre;   //cumulative Th volume of previous call
        R_pre;    //cumulative R volume of previous call
        L_now;    //cumulative L volume of current call
        Th_now;   //cumulative Th volume of current call
        R_now;    //cumulative R volume of current call
        L;        //left turn volume of this approach
        Th;       //through volume of this approach
        R;        //right turn volume of this approach
    );
    save each approach geometric data
    (
        call_time_counter;
        nlink_id; //NETSIM link ID
        Lper;     //Left truning percentage of this approach
        Thper;    //Th truning percentage of this approach
        Rper;     //R truning percentage of this approach
    );
    save diversion time at each intersection node(from 0, from diversion node);
    write general control data
    (
        total_x; //total_X
        call_dur;//call duration time
        ttdtst; //total detector stations, for MATLAB read speed data and their distance
        diversion_time_current;//diversion time

```

```

call_type; //call type, get value at ambin call, 1 T 2 rolling_hz t.3 sig rolling hz
path_length_main_effect; //path lengths of main path
path_length_diversion_effect; //path lengths of diversion path
FFS; //freeflow speed of mainline
FFS_d; //freeflow speed of alternative road
diversion_node_from0_dist; //diversion node distance from start point
merge_node_dist_from0; //merge node distance from start point
);
write volume and capacity data of current call
(
    freeway demand and capacity;
    off ramp volume and capacity;
    forecasted volume and saturation flow of all studied intersection groups;
);
write all detector stations data on mainline effect segment
(
    speed;
    dist0;    //distance from start node, 8001 in example network
    vol; //forecasted vol, for MATLAB speed forecast use
);
write DMS_Levels and their related probabilities;
write effect detector station and its id;
write FRESIM detector station volume of freeway demand and work zone capacity;
write queue length from last call;
write previous dms_level;
cm_file_written=1; //success of file written, MATLAB start only after this file is written.
}
//start MATLAB
int call_MATLAB(int call types)
{
    if (clock time < simulation length && c_m.txt file is ready)
    {
        for (int j=0; j<60; j++) //max MATLAB run time 120 = 60 * 2 seconds sleep after each inquire
        {
            check if MATLAB is active;
            if MATLAB is active,    wait 3 seconds;
            else
            {
                start MATLAB and run the related MATLAB code indicated by algorithm and call
types from user input file;

```



```

        //algorithm 1 SQP, 2 PS, 3 GA
        if call_type == 1 || call_type == 2
            call full optimization from model;
        if call_type == 3
            call signal optimization only;
        }
    }
}
record the running time of MATLAB;
}
//read periodic data, after MATLAB running, MATLAB does not need to keep data
int read_file_periodic()
{
    initial intersection interval data with MIN_GREEN, YELLOW ALLRED;
    read in Total_delay, endt_Queue_length, NETSIM_delay; // total delay from model estimation,
    queue length at the end of call, signal control delay
    read in all independent variable value; //total TotalX, 29 in example network
    check diversion compliance and then decide to use the new dms_level or not, for call type 2 and 3,
    call type 1 will use the new dms_level;
    for call_type 3, turn on all switches for signal update;
    save signal update data into a new vector in time order for updating;
    start_to_assign_veh=1; // switch on path assignment of vehicles
}
//read the interval of an intersection
void take_node_signal_interval(int NETSIMnode)
{
}
//*****
//change it at time int (offset/cycle_length), offset%cycle_length is write to true offset in intervals
//this actually happen in this call, call time is the offset start point,
//=====
int signal_change_at_offset()
{
    if (clock == a point on timeline && clock < simulation time)
    {
        call change_node_signal_interval(* signal_change_itr);
    }
}
//change the signal interval of an intersection, NETSIM internal node number
int change_node_signal_interval(SIGNAL_CHANGE change_one)

```

```

{
    check if node is an intersection node;
    for each interval
        update interval duration and offset for first interval;
}
//assign path to vehicle, based on optimal result, vehicle module is used here
void veh_path_assign()
{
    if (clock < simulation length && start_to_assign_veh == 1) //if within simulation period and
switch is on
    {
        get min and max generated_vehicle_ids;
        for each vehicle in FRESIM
        {
            if (FRESIM vehicle id !=0 && between [min, max])
            {
                if (a vehicle has no path assigned && its destination is not off ramp link)
                {
                    if (a vehicle is before diversion_node)
                    {
                        call
veh_diversion_choice(rate_desired[recommend_level-1],SIGMA) to see if assign path to this veh;
                        if (return value is yes)
                        {
                            assign_path(FRESIM_VEHICLES_mp_FREGVH[veh],2);//path 2 is diversion path, it is the main
path, can make veh return to freeway
                                path_assigned_veh_counter ++;
                        }
                    }
                }
            }
        }
    }
}
//for simulation-wide time related data output when simulation done
int write_summary()
{
    write running control data
    (
        Length of current TP;

```

```

        Time interval duration;
        length of time interval;
        slices of a time interval;
        time slice duration;
    )
    writeall TP length;
    writeSimulation total run time;
    writeinformation of each call;
    writedetector data from all calls;
    writedetector station data from all calls;
    writeapproaches data form all calls;
    writeapproaches geometric data;
    writediversion path data from all calls;
    writetotal diversion travel time from all calls;
    writevolumes for MATLAB optimization;
    writefreeway demand and off ramp volume;
    writeXs from MATLAB optimization from all calls
    writeforecasted FRESIM detector speed;
    writebefore and after update intersection node's interval data from all calls;
    writecumulative path assigned vehicle number;
    writeSignal change as timeline;
    writestanding queue delay estimation data;
}
double forcaste_freespeed(long call)
{
    use density speed curve to forecast speed given estimated density;
}
//set forecast speed value, no forecasting
double NOforcaste_freespeed(long call)
{
    all FRESIM detector forecasted speed = detected speed;
}
//veh diversion stochastic choice
int veh_diversion_choice(double mean_diversion_rate, double sigma)
{
    generate [0,1] random number;
    generate EV random number, given mu and sigma;
    compare the random number and desired diversion probability and make choice;
}
DLL_EXPORT void __stdcall MAIN(int calltype)

```

```

{
    call write_file_periodic();
    call MATLAB(calltype);
    for (int j=0;j<60;j++) //max MATLAB run time 180 = 60 * 3 seconds sleeping after each inquiring
    {
        check to see if MATLAB is running;
        if MATLAB is running
            sleep 3 seconds;
        else
            calculate MATLAB running time and break;
    }
    call read_file_periodic();
}

//call MATLAB and read in returns from MATLAB, write data is separated for compliance check before
read in function.
DLL_EXPORT void __stdcall MAINa(int calltype)
{
    call MATLAB(calltype);
    for (int j=0;j<60;j++) //max MATLAB run time 180 = 60 * 3 seconds sleeping after each inquiring
    {
        check to see if MATLAB is running;
        if MATLAB is running
            sleep 3 seconds;
        else
            calculate MATLAB running time and break;
    }
    call read_file_periodic();
}

%MATLAB model code
function [x fval ] = model_part1()
open data file from CORSIM c_m.txt
read in data
(
    number of variables;
    call duration;
    diversion time on alternative road;
    call type;
    freeway studied segment distance ;
    alternative road studied segment distance;
    free flow speed on mainline;

```

```

    free flow speed on alternative road;
    diversion node distance from 0;
    merge node distance from 0;
    volume and saturation flow rate of each group of all intersections and mainline;
    all detector stations' speed and distance from 0;
    total VMS levels;
    turning probability of each VMS level;
    number of effective detector stations on freeway;
    effective detector station number ;
    mainline demand detector volume;
    work zone bottleneck capacity detector volume;
    standing queue length at the end of last call;
)
compute minimal cycle length and use it as low bound of cycle length
%=====
function [Total_delay,NETSIM_delay,new_cumQ] = modelb(x)
    diversion volume = turning percentage * freeway demand;
    standing queue length estimation;
    detector station under standing queue influence check;
    forecast freeway effective detector station speeds f as flow-seed curve;
    compute    control delay;
    compute    freeway travel delay;
    compute    freeway stochastic queue delay;
    if standing queue on freeway
        compute freeway standing queue delay;
    end
    compute    diversion travel time delay
    compute    Total_delay = double(Total_delay_Freeway + Total_delay_control +
Total_delay_detour);%
    initialize variables and options for algorithm;
    switch (algorithm indication from user input file)
        case 1: %SQP
            call    fmincon(@modelb,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
        case 2: %PS
            call    patternsearch(@modelb,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
        case 3: %GA
            call    GA(@modelb,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
        default: %SQP
            call    fmincon(@modelb,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
    end
end

```

```

end
find VMS_level from returned diversion volume and approximate it to the closest level;
%because the level approximation, actually diversion volume changed, signal
%needs to be optimized for this new diversion volume
call model_part2; %signal optimization only
call modelb; % to get total_delay etc for the final optimized results;
write returned variables value to m_c.txt
exit MATLAB;
end
function [x fval ] = model_part2()
open data file from CORSIM c_m.txt
read in data
(
    number of variables;
    call duration;
    diversion time on alternative road;
    call type;
    freeway studied segment distance ;
    alternative road studied segment distance;
    free flow speed on mainline;
    free flow speed on alternative road;
    diversion node distance from 0;
    merge node distance from 0;
    volume and saturation flow rate of each group of all intersections and mainline;
    all detector stations' speed and distance from 0;
    total VMS levels;
    turning probability of each VMS level;
    number of effective detector stations on freeway;
    effective detector station number ;
    mainline demand detector volume;
    work zone bottleneck capacity detector volume;
    standing queue length at the end of last call;
)
compute minimal cycle length and use it as low bound of cycle length
%=====
function [Total_delay,NETSIM_delay,new_cumQ] = modelb_sig(x)
    diversion volume = turning percentage * freeway demand;
    compute control delay;
    initialize variables and options for algorithm;
    switch (algorithm indication from user input file)

```

```

case 1: %SQP
    call fmincon(@modelb_sig,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
case 2: %PS
    call patternsearch(@modelb_sig,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
case 3: %GA
    call GA(@modelb,X0_sig,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
default: %SQP
    call fmincon(@modelb_sig,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
end
end
call model_part2; %signal optimization only
call modelb_sig to get total_delay for the final optimized timing plan;
write returned variables value to m_c.txt
end

function [x fval ] = model_part2_sig()
open data file from CORSIM c_m.txt
read in data
(
    number of variables;
    call duration;
    diversion time on alternative road;
    call type;
    freeway studied segment distance ;
    alternative road studied segment distance;
    free flow speed on mainline;
    free flow speed on alternative road;
    diversion node distance from 0;
    merge node distance from 0;
    volume and saturation flow rate of each group of all intersections and mainline;
    all detector stations' speed and distance from 0;
    total VMS levels;
    turning probability of each VMS level;
    number of effective detector stations on freeway;
    effective detector station number ;
    mainline demand detector volume;
    work zone bottleneck capacity detector volume;
    standing queue length at the end of last call;
)
compute minimal cycle length and use it as low bound of cycle length

```

```

%=====
function [Total_delay,NETSIM_delay,new_cumQ] = modelb_sig(x)
    diversion volume = turning percentage * freeway demand;
    compute    control delay;
    initialize variables and options for algorithm;
    switch (algorithm indication from user input file)
        case 1: %SQP
            call  fmincon(@modelb_sig,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
        case 2: %PS
            call  patternsearch(@modelb_sig,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
        case 3: %GA
            call  GA(@modelb,X0_sig,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
        default: %SQP
            call  fmincon(@modelb_sig,X0,Aineq,Bineq,Aeq,Beq,lb,ub,nonlconFunction,options);
    end
end
call  model_part2; %signal optimization only
call  modelb_sig to get total_delay for the final optimized timing plan;
write  returned variables value to m_c.txt
exit  MATLAB;
end

```


APPENDIX B.

EXAMPLE OF USER INPUT DATA FILE

```

//For comments, must be 3 lines before data
//For additional comments,
//Section1// Total intersection and intersection properties
4   3   //total_intersections>>num_interX_property;
//# Legs property 2, default offset
39 3 450
41 4 1050
43 4 1600
45 4 2000
//section 2// main path and diversion path,
12 13 67 37 46 //diversion_node>>after_diversion_node>>merge node>>path_nodes1 path_nodes2
8001 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 70 27 28 29 30 31 67 32 33 8002
//main path, after diversion node
8001 1 2 3 4 5 6 7 8 9 10 11 12 7001 34 35 36 37 38 39 40 71 41 72 42 73 43 74 44 75 45 76 46 47 48 49 50
51 52 53 7002 68 67 32 33 8002 //diversion path
//section 3// variables
29 //total number of x
//mcm_input/output data order, vol and saturation flow in c_m.txt, x in m_c.txt (diversion vol, cycle length,
Xs)
//# linkUN, linkDN, Turn #, interval#, saturation flow, first 2 rows, station# and capacity/saturation flow
1   1   2300
2   8   1800
3   38  39  4   1   1800
4   69  39  2   2   3600
5   40  39  2   2   3600
6   71  41  1   2   1800
7   71  41  5   3   3600
8   56  41  1   2   1800
9   56  41  5   3   3600
10  72  41  1   4   1800
11  72  41  5   1   3600
12  55  41  1   4   1800
13  55  41  5   1   3600
14  58  43  1   2   1800
15  58  43  5   3   3600
16  60  43  1   2   1800
17  60  43  5   3   3600
18  74  43  1   4   1800
19  74  43  5   1   3600
20  73  43  1   4   1800

```

21	73	43	5	1	3600
22	76	45	1	2	1800
23	76	45	5	3	3600
24	62	45	1	2	1800
25	62	45	5	3	3600
26	64	45	1	4	1800
27	64	45	5	1	3600
28	75	45	1	4	1800
29	75	45	5	1	3600

//Optimization alg 1,SQP, 2, PS, 3 GA. T sec and rolling sec, free FFS and detour FFS mph

1 3600 1800 900 65 30

//VMS levels and desired mean probability

8 0.01 0.1 0.2 0.3 0.4 0.5 0.7 0.9

//effective detector station ID, as in order after diversion node, total, ID1 ID2...

6 2 3 4 5 10 6

//Fa and Fc detector station, for standing Queue delay estimation

10

APPENDIX C.

EXAMPLE OF RTE OUTPUT DATA FILE

29	900	10	3584	1	38891	62797	65	30	34323	73214
724		2300								
297		1800								
281		1800								
175		3600								
167		3600								
234		1800								
86		3600								
73		1800								
126		3600								
41		1800								
160		3600								
46		1800								
151		3600								
37		1800								
134		3600								
44		1800								
151		3600								
82		1800								
189		3600								
30		1800								
355		3600								
52		1800								
130		3600								
86		1800								
102		3600								
67		1800								
148		3600								
203		1800								
81		3600								
55	24830	724								
57	35331	425								
56	49342	426								
56	55319	440								
36	61362	541								
57	71372	552								
55	75104	772								
53	0	297								
30	0	220								
57	66229	547								

8
0.0100000
0.100000
0.200000
0.300000
0.400000
0.500000
0.700000
0.900000
6
2
3
4
5
10
6
724
547
785
4

APPENDIX D.

EXAMPLE OF MATLAB OUTPUT DATA FILE

5139 785 924

4

223

178

39

39

112

27

112

27

40

33

40

33

29

57

29

57

41

85

41

85

41

33

41

33

78

61

78

61

APPENDIX E.

EXAMPLE OF CORSIM RTE RUNNING LOG

Input File: C:\VMS\RTE_run_fmin\09 light\light20.trf

Output File: C:\VMS\RTE_run_fmin\09 light\light20.out

Animation Files Enabled

FHWA TRAF-CORSIM SIMULATION MODEL: VERSION 6.0

MARCH 2006

FHWA OFFICE OF OPERATIONS RESEARCH, DEVELOPMENT AND TECHNOLOGY

CURRENT DATE AND TIME : Tue Sep 23 04:44:38 2008

BEGINNING CHECK FOR OBSOLETE INPUT FORMATS

.... DONE CHECKING FORMATS

PROCESSING TIME PERIOD 1 INPUT DATA.....

CHECKING NETSIM SUBNETWORK DATA.....

DONE NETSIM SUBNETWORK DATA (0.00 CPU Seconds)

CHECKING FRESIM SUBNETWORK DATA.....

DONE FRESIM SUBNETWORK DATA (0.04 CPU Seconds)

PROCESSING TIME PERIOD 2 INPUT DATA.....

CHECKING FRESIM SUBNETWORK DATA.....

DONE FRESIM SUBNETWORK DATA (0.01 CPU Seconds)

CHECKING NETSIM SUBNETWORK DATA.....

DONE NETSIM SUBNETWORK DATA (0.00 CPU Seconds)

PROCESSING TIME PERIOD 3 INPUT DATA.....

CHECKING FRESIM SUBNETWORK DATA.....

DONE FRESIM SUBNETWORK DATA (0.00 CPU Seconds)

CHECKING NETSIM SUBNETWORK DATA.....

DONE NETSIM SUBNETWORK DATA (0.00 CPU Seconds)

CHECKING NETSIM SUBNETWORK DATA.....

DONE NETSIM SUBNETWORK DATA (0.00 CPU Seconds)

CHECKING FRESIM SUBNETWORK DATA.....

DONE FRESIM SUBNETWORK DATA (0.02 CPU Seconds)

CHECKING FOR SHORT LINKS

.... DONE CHECKING FOR SHORT LINKS

Initial stage, Output time independent data

FAILED TO ALLOCATE MEMORY

START THE INITIALIZATION PROCESS.....

COMPLETED 120 SECONDS OF INITIALIZATION..... (CORSIM) (0.17 CPU Seconds)

COMPLETED 240 SECONDS OF INITIALIZATION..... (CORSIM) (0.24 CPU Seconds)

COMPLETED 360 SECONDS OF INITIALIZATION..... (CORSIM) (0.31 CPU Seconds)

COMPLETED 480 SECONDS OF INITIALIZATION..... (CORSIM) (0.37 CPU Seconds)

COMPLETED 600 SECONDS OF INITIALIZATION..... (CORSIM) (0.44 CPU Seconds)

NETWORK DID NOT REACH EQUILIBRIUM.

COMPLETED 120 SECONDS OF SIMULATION..... (CORSIM) (1.41 CPU Seconds)

COMPLETED 240 SECONDS OF SIMULATION..... (CORSIM) (1.70 CPU Seconds)

COMPLETED 360 SECONDS OF SIMULATION..... (CORSIM) (1.80 CPU Seconds)
 COMPLETED 480 SECONDS OF SIMULATION..... (CORSIM) (1.86 CPU Seconds)
 COMPLETED 600 SECONDS OF SIMULATION..... (CORSIM) (1.87 CPU Seconds)
 COMPLETED 720 SECONDS OF SIMULATION..... (CORSIM) (1.92 CPU Seconds)
 COMPLETED 840 SECONDS OF SIMULATION..... (CORSIM) (1.94 CPU Seconds)
 COMPLETED 960 SECONDS OF SIMULATION..... (CORSIM) (1.97 CPU Seconds)
 COMPLETED 1080 SECONDS OF SIMULATION..... (CORSIM) (1.94 CPU Seconds)
 COMPLETED 1200 SECONDS OF SIMULATION..... (CORSIM) (1.93 CPU Seconds)
 COMPLETED 1320 SECONDS OF SIMULATION..... (CORSIM) (1.94 CPU Seconds)
 COMPLETED 1440 SECONDS OF SIMULATION..... (CORSIM) (1.96 CPU Seconds)
 COMPLETED 1560 SECONDS OF SIMULATION..... (CORSIM) (1.95 CPU Seconds)
 COMPLETED 1680 SECONDS OF SIMULATION..... (CORSIM) (1.98 CPU Seconds)
 COMPLETED 1800 SECONDS OF SIMULATION..... (CORSIM) (2.00 CPU Seconds)
 COMPLETED 1920 SECONDS OF SIMULATION..... (CORSIM) (2.02 CPU Seconds)
 COMPLETED 2040 SECONDS OF SIMULATION..... (CORSIM) (2.03 CPU Seconds)
 COMPLETED 2160 SECONDS OF SIMULATION..... (CORSIM) (2.00 CPU Seconds)
 COMPLETED 2280 SECONDS OF SIMULATION..... (CORSIM) (2.17 CPU Seconds)
 COMPLETED 2400 SECONDS OF SIMULATION..... (CORSIM) (2.91 CPU Seconds)
 COMPLETED 2520 SECONDS OF SIMULATION..... (CORSIM) (2.03 CPU Seconds)
 COMPLETED 2640 SECONDS OF SIMULATION..... (CORSIM) (2.19 CPU Seconds)
 COMPLETED 2760 SECONDS OF SIMULATION..... (CORSIM) (2.20 CPU Seconds)
 COMPLETED 2880 SECONDS OF SIMULATION..... (CORSIM) (2.11 CPU Seconds)
 COMPLETED 3000 SECONDS OF SIMULATION..... (CORSIM) (2.12 CPU Seconds)
 COMPLETED 3120 SECONDS OF SIMULATION..... (CORSIM) (2.16 CPU Seconds)
 COMPLETED 3240 SECONDS OF SIMULATION..... (CORSIM) (2.08 CPU Seconds)
 COMPLETED 3360 SECONDS OF SIMULATION..... (CORSIM) (2.08 CPU Seconds)
 COMPLETED 3480 SECONDS OF SIMULATION..... (CORSIM) (2.09 CPU Seconds)
 COMPLETED 3600 SECONDS OF SIMULATION..... (CORSIM) (2.22 CPU Seconds)

CHECKING FRESIM SUBNETWORK DATA.....

DONE FRESIM SUBNETWORK DATA (0.01 CPU Seconds)

Reach a T, opti and change all!...

Write file for MATLAB done!...

MATLAB is running...

MATLAB has run 20 Seconds

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB Optimization done, MATLAB ran 50 Seconds Finally
 Output from MATLAB done -Main!...
 Changed one node: 39 After 0 Cycle Length 192, gclk 3600
 COMPLETED 3720 SECONDS OF SIMULATION..... (CORSIM) (55.37 CPU Seconds)
 COMPLETED 3840 SECONDS OF SIMULATION..... (CORSIM) (2.11 CPU Seconds)
 COMPLETED 3960 SECONDS OF SIMULATION..... (CORSIM) (2.11 CPU Seconds)
 Changed one node: 41 After 2 Cycle Length 192, gclk 3984
 COMPLETED 4080 SECONDS OF SIMULATION..... (CORSIM) (2.14 CPU Seconds)
 COMPLETED 4200 SECONDS OF SIMULATION..... (CORSIM) (2.07 CPU Seconds)
 COMPLETED 4320 SECONDS OF SIMULATION..... (CORSIM) (2.09 CPU Seconds)
 Changed one node: 43 After 4 Cycle Length 192, gclk 4368
 COMPLETED 4440 SECONDS OF SIMULATION..... (CORSIM) (2.12 CPU Seconds)
 Reach a t_SIG, opti signal and update without VMS LVL changed!...
 Write file for MATLAB t_SIG done!...
 MATLAB is running...
 MATLAB has run 20 Seconds
 MATLAB is still running, CORSIM should wait...
 MATLAB Optimization done, MATLAB ran 23 Seconds Finally
 Read data file from MATLAB done -MAINa!...
 COMPLETED 4560 SECONDS OF SIMULATION..... (CORSIM) (28.29 CPU Seconds)
 COMPLETED 4680 SECONDS OF SIMULATION..... (CORSIM) (2.17 CPU Seconds)
 Changed one node: 39 After 1 Cycle Length 186, gclk 4686
 Changed one node: 45 After 6 Cycle Length 192, gclk 4752
 COMPLETED 4800 SECONDS OF SIMULATION..... (CORSIM) (2.17 CPU Seconds)
 COMPLETED 4920 SECONDS OF SIMULATION..... (CORSIM) (2.19 CPU Seconds)
 COMPLETED 5040 SECONDS OF SIMULATION..... (CORSIM) (2.22 CPU Seconds)
 Changed one node: 41 After 3 Cycle Length 186, gclk 5058
 COMPLETED 5160 SECONDS OF SIMULATION..... (CORSIM) (2.21 CPU Seconds)
 COMPLETED 5280 SECONDS OF SIMULATION..... (CORSIM) (2.24 CPU Seconds)
 COMPLETED 5400 SECONDS OF SIMULATION..... (CORSIM) (2.26 CPU Seconds)
 Reach a t, Check compliance to see if LEVEL change needed, SIGNAL WILL BE CHANGED!...
 Write file for MATLAB VMS rolling t done!...
 MATLAB is running...
 MATLAB has run 20 Seconds
 MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB Optimization done, MATLAB ran 35 Seconds Finally
 Read data file from MATLAB done -MAINa!...
 COMPLETED 5520 SECONDS OF SIMULATION..... (CORSIM) (40.44 CPU Seconds)
 Changed one node: 39 After 1 Cycle Length 164, gclk 5564
 Changed one node: 43 After 6 Cycle Length 186, gclk 5616
 COMPLETED 5640 SECONDS OF SIMULATION..... (CORSIM) (2.28 CPU Seconds)
 COMPLETED 5760 SECONDS OF SIMULATION..... (CORSIM) (2.27 CPU Seconds)
 COMPLETED 5880 SECONDS OF SIMULATION..... (CORSIM) (2.28 CPU Seconds)
 Changed one node: 45 After 8 Cycle Length 186, gclk 5988
 COMPLETED 6000 SECONDS OF SIMULATION..... (CORSIM) (2.28 CPU Seconds)
 Changed one node: 41 After 4 Cycle Length 164, gclk 6056
 COMPLETED 6120 SECONDS OF SIMULATION..... (CORSIM) (2.25 CPU Seconds)
 COMPLETED 6240 SECONDS OF SIMULATION..... (CORSIM) (2.20 CPU Seconds)
 Reach a t_SIG, opti signal and update without VMS LVL changed!...
 Write file for MATLAB t_SIG done!...
 MATLAB is running...
 MATLAB has run 20 Seconds
 MATLAB Optimization done, MATLAB ran 20 Seconds Finally
 Read data file from MATLAB done -MAINa!...
 COMPLETED 6360 SECONDS OF SIMULATION..... (CORSIM) (25.50 CPU Seconds)
 COMPLETED 6480 SECONDS OF SIMULATION..... (CORSIM) (2.27 CPU Seconds)
 Changed one node: 43 After 7 Cycle Length 164, gclk 6548
 Changed one node: 39 After 4 Cycle Length 72, gclk 6588
 COMPLETED 6600 SECONDS OF SIMULATION..... (CORSIM) (2.33 CPU Seconds)
 COMPLETED 6720 SECONDS OF SIMULATION..... (CORSIM) (2.28 CPU Seconds)
 COMPLETED 6840 SECONDS OF SIMULATION..... (CORSIM) (2.27 CPU Seconds)
 COMPLETED 6960 SECONDS OF SIMULATION..... (CORSIM) (2.29 CPU Seconds)
 Changed one node: 41 After 10 Cycle Length 72, gclk 7020
 Changed one node: 45 After 10 Cycle Length 164, gclk 7040
 COMPLETED 7080 SECONDS OF SIMULATION..... (CORSIM) (2.33 CPU Seconds)
 COMPLETED 7200 SECONDS OF SIMULATION..... (CORSIM) (2.45 CPU Seconds)
 CHECKING FRESIM SUBNETWORK DATA.....
 DONE FRESIM SUBNETWORK DATA (0.00 CPU Seconds)
 Reach a T, opti and change all!...
 Write file for MATLAB done!...
 MATLAB is running...
 MATLAB has run 20 Seconds

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...

MATLAB Optimization done, MATLAB ran 32 Seconds Finally

Output from MATLAB done -Main!...

 COMPLETED 7320 SECONDS OF SIMULATION..... (CORSIM) (37.46 CPU Seconds)

 COMPLETED 7440 SECONDS OF SIMULATION..... (CORSIM) (2.22 CPU Seconds)

Changed one node: 39 After 5 Cycle Length 66, gclk 7530

 COMPLETED 7560 SECONDS OF SIMULATION..... (CORSIM) (2.21 CPU Seconds)

Changed one node: 43 After 18 Cycle Length 72, gclk 7596

 COMPLETED 7680 SECONDS OF SIMULATION..... (CORSIM) (2.25 CPU Seconds)

 COMPLETED 7800 SECONDS OF SIMULATION..... (CORSIM) (2.22 CPU Seconds)

 COMPLETED 7920 SECONDS OF SIMULATION..... (CORSIM) (2.21 CPU Seconds)

Changed one node: 41 After 11 Cycle Length 66, gclk 7926

Changed one node: 45 After 24 Cycle Length 72, gclk 8028

 COMPLETED 8040 SECONDS OF SIMULATION..... (CORSIM) (2.20 CPU Seconds)

Reach a t_SIG, opti signal and update without VMS LVL changed!...

Write file for MATLAB t_SIG done!...

MATLAB is running...

MATLAB has run 20 Seconds

MATLAB Optimization done, MATLAB ran 20 Seconds Finally

Read data file from MATLAB done -MAINa!...

 COMPLETED 8160 SECONDS OF SIMULATION..... (CORSIM) (25.36 CPU Seconds)

 COMPLETED 8280 SECONDS OF SIMULATION..... (CORSIM) (2.17 CPU Seconds)

Changed one node: 39 After 2 Cycle Length 144, gclk 8388

 COMPLETED 8400 SECONDS OF SIMULATION..... (CORSIM) (2.16 CPU Seconds)

Changed one node: 43 After 19 Cycle Length 66, gclk 8454

 COMPLETED 8520 SECONDS OF SIMULATION..... (CORSIM) (2.29 CPU Seconds)

 COMPLETED 8640 SECONDS OF SIMULATION..... (CORSIM) (2.16 CPU Seconds)

 COMPLETED 8760 SECONDS OF SIMULATION..... (CORSIM) (2.12 CPU Seconds)

Changed one node: 41 After 5 Cycle Length 144, gclk 8820

 COMPLETED 8880 SECONDS OF SIMULATION..... (CORSIM) (2.15 CPU Seconds)

Changed one node: 45 After 27 Cycle Length 66, gclk 8982

 COMPLETED 9000 SECONDS OF SIMULATION..... (CORSIM) (2.15 CPU Seconds)

Reach a t, Check compliance to see if LEVEL change needed, SIGNAL WILL BE CHANGED!...

Write file for MATLAB VMS rolling t done!...

MATLAB is running...

MATLAB has run 20 Seconds

MATLAB is still running, CORSIM should wait...

MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB is still running, CORSIM should wait...
 MATLAB Optimization done, MATLAB ran 35 Seconds Finally
 Read data file from MATLAB done -MAINa!...
 COMPLETED 9120 SECONDS OF SIMULATION..... (CORSIM) (40.42 CPU Seconds)
 COMPLETED 9240 SECONDS OF SIMULATION..... (CORSIM) (2.14 CPU Seconds)
 COMPLETED 9360 SECONDS OF SIMULATION..... (CORSIM) (2.19 CPU Seconds)
 Changed one node: 43 After 9 Cycle Length 144, gclk 9396
 COMPLETED 9480 SECONDS OF SIMULATION..... (CORSIM) (2.17 CPU Seconds)
 COMPLETED 9600 SECONDS OF SIMULATION..... (CORSIM) (2.21 CPU Seconds)
 COMPLETED 9720 SECONDS OF SIMULATION..... (CORSIM) (2.18 CPU Seconds)
 Changed one node: 45 After 12 Cycle Length 144, gclk 9828
 COMPLETED 9840 SECONDS OF SIMULATION..... (CORSIM) (2.18 CPU Seconds)
 Reach a t_SIG, opti signal and update without VMS LVL changed!...
 Write file for MATLAB t_SIG done!...
 MATLAB is running...
 MATLAB has run 20 Seconds
 MATLAB Optimization done, MATLAB ran 20 Seconds Finally
 Read data file from MATLAB done -MAINa!...
 COMPLETED 9960 SECONDS OF SIMULATION..... (CORSIM) (25.36 CPU Seconds)
 COMPLETED 10080 SECONDS OF SIMULATION..... (CORSIM) (2.17 CPU Seconds)
 Changed one node: 39 After 2 Cycle Length 124, gclk 10148
 COMPLETED 10200 SECONDS OF SIMULATION..... (CORSIM) (2.20 CPU Seconds)
 COMPLETED 10320 SECONDS OF SIMULATION..... (CORSIM) (2.19 CPU Seconds)
 COMPLETED 10440 SECONDS OF SIMULATION..... (CORSIM) (2.22 CPU Seconds)
 COMPLETED 10560 SECONDS OF SIMULATION..... (CORSIM) (2.22 CPU Seconds)
 Changed one node: 41 After 6 Cycle Length 124, gclk 10644
 COMPLETED 10680 SECONDS OF SIMULATION..... (CORSIM) (2.20 CPU Seconds)
 COMPLETED 10800 SECONDS OF SIMULATION..... (CORSIM) (2.33 CPU Seconds)
 TOTAL CPU TIME FOR THIS RUN = 455.51 SECONDS
 Reach a T, opti and change all!...
 Write file for MATLAB done!...
 MATLAB Optimization done, MATLAB ran 0 Seconds Finally
 Output from MATLAB done -Main!...
 RTE job done, Exit RTE.

APPENDIX F.

MATLAB ALGORITHM OPTIONS FOR CASE STUDY

GA Options in MATLAB

```
options = gaoptimset(options, 'PopInitRange' ,[ 20 ; 800 ]);
options = gaoptimset(options, 'EliteCount' ,1);
options = gaoptimset(options, 'CrossoverFraction' ,0.74479);
options = gaoptimset(options, 'MigrationDirection' , 'both');
options = gaoptimset(options, 'MigrationInterval' ,10);
options = gaoptimset(options, 'MigrationFraction' ,0.93954);
options = gaoptimset(options, 'Generations' ,100);
options = gaoptimset(options, 'StallGenLimit' ,100);
options = gaoptimset(options, 'StallTimeLimit' ,100);
options = gaoptimset(options, 'InitialPopulation' ,X0);
options = gaoptimset(options, 'InitialPenalty' ,30);
options = gaoptimset(options, 'SelectionFcn' ,@selectionremainder);
options = gaoptimset(options, 'CrossoverFcn' ,{ @crossoverheuristic 1.932 });
options = gaoptimset(options, 'MutationFcn' ,@mutationadaptfeasible);
```

PS Options in MATLAB

```
options = psoptimset(options, 'TolMesh' ,0.001);
options = psoptimset(options, 'MaxIter' ,200);
options = psoptimset(options, 'TimeLimit' ,60);
options = psoptimset(options, 'MeshContraction' ,0.3);
options = psoptimset(options, 'MeshExpansion' ,250);
options = psoptimset(options, 'MeshAccelerator' , 'on');
options = psoptimset(options, 'MeshRotate' , 'off');
options = psoptimset(options, 'InitialMeshSize' ,100);
options = psoptimset(options, 'InitialPenalty' ,30);
options = psoptimset(options, 'MaxMeshSize' ,5000);
options = psoptimset(options, 'CompletePoll' , 'on');
options = psoptimset(options, 'PollingOrder' , 'success');
options = psoptimset(options, 'SearchMethod' ,@GPSPositiveBasis2N);
options = psoptimset(options, 'CompleteSearch' , 'on');
options = psoptimset(options, 'Display' , 'iter');
options = psoptimset(options, 'Cache' , 'on');
```

fmincon Options in MATLAB

```
options = optimset(options, 'LargeScale' , 'off');
```