2006

A stochastic mesoscopic cell-transmission model for operational analysis of large-scale transportation networks

Ciprian Danut Alecsandru
Louisiana State University and Agricultural and Mechanical College, calecs1@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations

Part of the Civil and Environmental Engineering Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_dissertations/705

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
A STOCHASTIC MESOSCOPIC CELL-TRANSMISSION MODEL FOR
OPERATIONAL ANALYSIS OF LARGE-SCALE TRANSPORTATION
NETWORKS

A Dissertation
Submitted to Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Civil and Environmental Engineering

by
Ciprian D. Alecsandru
B. Engr., Polytechnic University of Bucharest, 1997
M.S. in C.E., Louisiana State University, 2003
August 2006
To my beautiful family,
Ancuta, Ilinca, and Mati

Baton Rouge, June 2006
ACKNOWLEDGEMENTS

I am very grateful to my major professor, Dr Sherif Ishak, whose encouragement, guidance, hours of assistance, sound advice, and dedication immeasurably improved this dissertation, and indeed were indispensable to the completion of the project. I am thankful for thoughtful and insightful comments and his willingness to give of his time to ensure that a quality document will result from this endeavor. I am especially thankful for not just for the wonderful collaboration with Dr. Ishak but also for a great friendship that we shared for about five years and I am confident will persist over the years. I would like to thank the professors who served on my doctoral committee, Dr. Chester Wilmot, Dr. Brian Wolshon, Dr. Charles McAllister, and Dr Randy Price, for their help, guidance and understanding. Their valuable comments helped finalizing a clearer version of my dissertation document. I extend my special thanks to Professor Clifford Mugnier, who’s pertinent advice and honesty I will always value and remember.

All these years at LSU, too many miles away from our home and family, would have been considerably more difficult without our new and old friends from Baton Rouge and all over the US. Special thanks to Stacey Jones and her family for nurturing us in the southern culture and for being our beloved foster family. Nevertheless, I am grateful for being blessed with a wonderful family of my own. My wife was always next to me when difficult moments were overwhelming while also dealing with her PhD program and raising two wonderful children. Of course, without the help of our families we could not have reached so far in our life and career and we will strive to educate our children at least as well as our parents did with us.

Most of all, I am grateful to our Lord Jesus Christ for overseeing us in all our endeavors.
# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS** ........................................................................................................ III

**LIST OF TABLES** ................................................................................................................ VII

**LIST OF FIGURES** .............................................................................................................. IX

**ABSTRACT** .......................................................................................................................... XIII

**CHAPTER 1 INTRODUCTION** ............................................................................................ 1
  1.1 Research Motivation ........................................................................................................ 3
  1.2 Problem Statement .......................................................................................................... 4
  1.3 Objectives ....................................................................................................................... 5

**CHAPTER 2 BACKGROUND** ............................................................................................ 9
  2.1 Literature Review ........................................................................................................... 9
    2.1.1 Dynamic Traffic Assignment Applications ....................................................... 10
    2.1.2 Network Design Problem Applications ......................................................... 12
    2.1.3 Traffic Operations Applications ..................................................................... 13
    2.1.4 Emergency Evacuation Applications ............................................................. 14
    2.1.5 Hybrid Simulation Applications ..................................................................... 15
  2.2 The LWR Model .......................................................................................................... 16
  2.3 Network Topology for CTM ....................................................................................... 17
    2.3.1 Flow Advancing Equations of CTM ............................................................... 21
    2.3.2 The Flow Conservation Equation ................................................................... 24
    2.3.3 Advantages and Disadvantages of CTM ......................................................... 24

**CHAPTER 3 VARIABLE CELL LENGTH MODIFICATIONS** ............................................. 26
  3.1 Introduction ................................................................................................................... 26
  3.2 Subcells ......................................................................................................................... 26
    3.2.1 Internal Update for Subcells .......................................................................... 27
    3.2.2 External Update for Cells ............................................................................. 30
    3.2.3 Vehicle Wait Times ....................................................................................... 32
    3.2.4 Determination of $x'_j$ .................................................................................. 37
    3.2.5 Determination of $\delta_j$ ................................................................................ 38
  3.3 Experimental Analysis ................................................................................................. 39
    3.3.1 Effect of Variable Cell Length on Simulation Results ..................................... 39
    3.3.2 Non-Discrete Vehicle Movements .................................................................. 44
  3.4 Summary ....................................................................................................................... 47
LIST OF TABLES

Table 1  Traffic Simulation Models........................................................................................................4
Table 2:  Network properties (Base Case) ......................................................................................................40
Table 3:  Network properties (Case 1) .........................................................................................................41
Table 4:  Network properties (Case 2) .........................................................................................................41
Table 5:  Traffic demand for all cases ...........................................................................................................42
Table 6:  Network characteristics used to examine discretionary LC behavior ...........................................59
Table 7:  Traffic demand used to examine discretionary LC behavior..........................................................59
Table 8:  Network parameters.......................................................................................................................70
Table 9:  Traffic demand................................................................................................................................70
Table 10: Merging demand (free-flow) .........................................................................................................90
Table 11: Merging traffic demand (congestion) ............................................................................................92
Table 12: Network parameters – (auxiliary lane case) ...............................................................................95
Table 13: Network parameters – (no auxiliary lane case) .............................................................................96
Table 14: Traffic demand..............................................................................................................................96
Table 15: Network details.............................................................................................................................106
Table 16: Traffic demand.............................................................................................................................107
Table 17: NGSIM data structure...................................................................................................................125
Table 18: Third order polynomials for fitting $\delta_j$ ......................................................................................130
Table 19: Testing dynamic variations in $\delta_j$ (network description) ............................................................134
Table 20: Testing dynamic variations in $\delta_j$ (traffic demand) .................................................................134
Table 21: Network configuration..................................................................................................................140
Table 22: Traffic demand.................................................................................................................140
Table 23: I-10 freeway (network details)..........................................................................................148
Table 24: I-10 freeway (traffic demand)..........................................................................................148
## LIST OF FIGURES

1: An example of the basic building block of CTM .............................................................18
2: Variation of cell types in CTM network topology............................................................20
3: CTM topological restrictions..........................................................................................20
4: Piecewise Linear Approximation of the q-k Relationship............................................21
5: Discretization of long cells into subcells .......................................................................27
6: External update for ordinary cells..................................................................................33
7: Update of wait time function after external movements into an ordinary cell ..........34
8: Example of a wait time functions for cell \( i \) .................................................................35
9: Update of wait time function after external movements into a diverging cell ..........37
10: Network topology (Base Case)......................................................................................39
11: Detailed topology of study section (Case 1).................................................................40
12: Detailed topology of study section (Case 2).................................................................41
13: Occupancy of the study section (base case).................................................................43
14: Absolute error in occupancy for comparative Cases 1 and 2.................................43
15: Effect of cell size variations on network travel time and average vehicular delay .......45
16: Effect of Movement Precision on Network Travel Time ..............................................46
17: Illustrative target lane disutility function..................................................................50
18: Probability to select a target lane for \( \beta = 0 \) ............................................................53
19: Probability to select a target lane for \( \beta = 0.5 \) .........................................................54
20: Probability to select a target lane for \( \beta = 1.5 \) ........................................................55
21: Example of mandatory lane changing ........................................................................57
22: Network layout and topology used to examine discretionary LC behavior ..........59
23: Lane occupancy distribution in cell 3 (base case) ........................................60
24: Lane occupancy distribution in cell 2 (base case) ........................................61
25: Lane occupancy distribution in cell 1 (base case) ........................................62
26: Occupancy distribution in cell 3 (case 1) .........................................................63
27: Occupancy distribution in cell 2 (case 1) .........................................................64
28: Occupancy distribution in cell 1 (case 1) .........................................................65
29: Lane occupancy distribution in cell 3 (case 2) .........................................................66
30: Occupancy distribution in cell 2 (case 2) .........................................................67
31: Occupancy distribution in cell 1 (case 2) .........................................................68
32: Network layout and topology ..........................................................69
33: Detailed structure of the diverging cell .........................................................69
34: Lane occupancy in the diverging cell (without lane-changing) ..................71
35: Lane occupancy in the diverging cell (with lane-changing) ..................72
36: Occupancy profile by subcells for lane 1 (no lane-changing) ..................75
37: Occupancy profile by subcells for lane 1 (with lane-changing) ..................76
38: Occupancy profile by subcells for lane 2 (no lane-changing) ..................76
39: Occupancy profile by subcells for lane 2 (with lane-changing) ..................77
40: Occupancy profile by subcells for lane 3 (no lane-changing) ..................78
41: Occupancy profile by subcells for lane 3 (with lane-changing) ..................80
42 An example of a merging cell i ........................................................................82
43: Merging Algorithm .......................................................................................85
44: A freeway merging junction ........................................................................86
45: Merging junction with no auxiliary lane ........................................................................... 89
46: Merging junction with one auxiliary lane ....................................................................... 89
47: Merging streams under free-flow conditions (no auxiliary lane case) ......................... 91
48: Merging streams under free-flow conditions (auxiliary lane case) .............................. 92
49: Merging streams under congested conditions (no auxiliary lane case) ...................... 93
50: Merging streams under congested conditions (auxiliary lane case) ............................ 94
51: Network topology (auxiliary lane case) ....................................................................... 95
52: Network topology (no auxiliary lane case) .................................................................... 95
53: Dynamic distribution of the merging flow rates (no auxiliary lane) ............................ 98
54: Dynamic distribution of the merging flow rates (one auxiliary lane) .......................... 99
55: An example of a diverging cell $i$ .................................................................................. 102
56: Diverging Algorithm .................................................................................................... 103
57: Network layout and topology ...................................................................................... 105
58: Lane occupancy in a diverging cell with 85/15 split ratio ........................................... 108
59: Lane occupancy in a diverging cell with 50/50 split ratio ........................................... 110
60: Lane occupancy in a diverging cell with 15/85 split ratio ........................................... 111
61: A hypothetical triangular distribution ......................................................................... 117
62: Random sampling of mean free-flow speed of vehicles in cell $i$ .................................. 119
63: Random sampling of mean minimum headway of vehicles in cell $i$ ............................ 120
64: Random sampling of mean minimum spacing of vehicles in cell $i$ ............................ 121
65: Random sampling of wave speeds in cell $j$ ............................................................... 122
66: NGSIM study area ....................................................................................................... 124
67: Illustrative example of $\delta_j$ computation ................................................................... 127
68: Distribution of means for $\delta_j$ ($x_i = 10 - 15\%$) .................................................................129
69: Polynomial fitting for $\delta_j$ and $x_i \in [0.1 - 0.25]$ ........................................................................131
70: Polynomial fitting for $\delta_j$ and $x_i \in [0.25 - 0.50]$ .......................................................................132
71: Polynomial fitting for $\delta_j$ and $x_i \in [0.50 - 0.75]$ .......................................................................133
72: Testing dynamic variations in $\delta_j$ (network topology) .................................................................134
73: Effect of constant $\delta_j$ on the occupancy of cell 1 ........................................................135
74: Effect of dynamic variations in $\delta_j$ on the occupancy of cell 1 ..................................................136
75: Network layout and topology ........................................................................................................140
76: Effect of traffic mix on the network travel time ..................................................................................141
77: A segment of I-10 freeway (west section) ......................................................................................146
78: A segment of I-10 freeway (east section) .......................................................................................147
79: Comparison between CORSIM and CTM for cell 12 ......................................................................149
80: Comparison between CORSIM and CTM for cell 14 .....................................................................150
81: Comparison between CORSIM and CTM for combined cells 16a and 16b ..................151
ABSTRACT

The cell transmission model (CTM), developed by Daganzo in 1994 was not fully exploited as an operations model for analysis of large-scale traffic networks. Because of its macroscopic / mesoscopic features, CTM offers calibration and computational advantages over microscopic models. This study presents a series of enhancements to the original form of CTM. These enhancements show potential to increase the model’s accuracy and realism of traffic flow representation. For example, topological enhancements and modifications to the flow advancing equation are introduced to allow variable cell lengths and non-discrete movements of vehicles between cells. In addition, implementation of lane-changing behavioral logics and algorithmic enhancements to model vehicle flows at network junctions demonstrate potential in modeling realistic non-homogeneous traffic streams in CTM. A calibration exercise was conducted to account for randomness in driving behavior using vehicle trajectory data. This proves the models potential in modeling stochastic variations of real-life networks. A sample freeway network of I-10 corridor in Baton Rouge was used to evaluate and compare the performance of the improved version of CTM versus CORSIM. The simulation results showed comparable performance of both platforms in terms of link occupancy (density) and total network travel time and demonstrate the potential of employing CTM in traffic operations applications.
CHAPTER 1 INTRODUCTION

The past few years have witnessed substantial development of transportation network modeling tools and stronger emphasis on addressing the need to model large-scale networks more accurately and efficiently. By comparison with analytical solutions, simulation environments offer a more suitable platform for off-line operational, design and planning analyses of transportation networks. Examples range from designing an optimal traffic signal timing plan to measuring the traffic impact of regional developments in large urban areas. Moreover, simulation systems can be effectively used to assess the impact of various policies and to evaluate the potential benefits of new real-time traffic control and management functions within the framework of intelligent transportation systems (ITS). With today’s remarkable advancement in computational resources the computational performance of simulation systems has substantially improved and subsequently made simulation tools more appealing to both practitioners and researchers.

The development of simulation models for transportation networks varies by the desired level of analysis and inherent stochasticity of the real-world network. Based on the level of analysis, traffic simulation models are typically defined as microscopic, macroscopic, or mesoscopic (a mixture of both). Macroscopic, also referred to as low-fidelity, models assume that traffic flow can be modeled as a one-dimensional continuous fluid and thus place more emphasis on the aggregate behavior and characteristics of the traffic stream. Macroscopic models utilize the fundamental traffic flow relationships between flow, density, and speed, and generally require less computational resources. However, macroscopic models do not retain the ability to account explicitly for possible stochastic and random variations in the simulated environment (e.g. driving behavior).
On the contrary, microscopic, also referred to as high fidelity, models are capable of tracing the movements of individual vehicles in time and space within the transportation network. In microscopic models, the traffic stream characteristics are derived from behavioral models that explicitly consider the interactions among vehicles in the traffic stream and the mathematical representation of both car-following and lane-changing maneuvers. Being high fidelity in nature, microscopic models typically require more computational resources and are more difficult to calibrate than macroscopic models. This is because the aggregate behavior of vehicles in any traffic stream is much easier to observe than the disaggregate behavior of individual vehicles.

The third group of simulation models combines both microscopic and macroscopic features into mesoscopic (medium-fidelity) models that typically incorporate the movement of clusters or platoons of vehicles and their interactions. Clearly, the process of selecting the most appropriate simulation tool largely depends on the type and scope of application as well as the desired level of analysis. For instance, regional emergency evacuation strategies are better evaluated with macroscopic simulation models because of the computational efficiency and greater interest in the overall aggregate behavior of the network. On the other hand, modeling a single corridor with multiple signalized intersections may be best carried out with microscopic simulation models, where stochastic characteristics are more readily accounted for and computational requirements are less significant.

Simulation models can also be classified as deterministic or stochastic. In stochastic models, random variations in simulation parameters are derived from probability distribution functions that describe the randomness inherent in both human and network
characteristics. Consequently, stochastic models generally allow for a more realistic representation of traffic flow dynamics in transportation networks. Unlike deterministic models, the performance measures derived from stochastic simulation models inherit a probabilistic feature from the network stochastic parameters, and therefore, must be collected from multiple simulation runs to achieve a desired level of confidence.

1.1 Research Motivation

To date, the review of the state-of-the-art traffic simulation models reveals that most of the existing models have mixed success in striking the balance between the simplicity, versatility, efficiency, comprehensiveness, and flexibility of modeling the transportation networks. A summary of the features of several major simulation models is presented in Table 1. It can be seen that macroscopic models, such as SYNCRO and TRANSYT-7F, which have a simple representation of traffic flow, are not suitable for planning applications and have not been designed for integration with ITS and traffic management strategies. On the other hand, microscopic models (e.g. CORSIM, PARAMICS, VISSIM, TRANSIMS, TransModeler, etc.) have a better representation of traffic by integrating various network components and type of flows (multimodal modeling), but they are difficult to calibrate. Moreover, when employed for area-wide systems, the computational resources can be expensive and complex, such as parallel computing. There is also a family of mesoscopic simulators such as NETFLO 1 and DYNASMART. However, the network flow representation in these systems is limited (e.g. the effect of traffic mix due to various vehicle types is not captured) and they have limited integration with ITS applications (DYNASMART is somewhat more flexible, but its limitation consists in the deterministic approach used). To the author’s knowledge, a
simple, efficient and comprehensive model that addresses both planning and operational analysis needs of large-scale traffic networks at a macroscopic scale has not been developed yet. This model would serve as a valuable tool for supporting applications that involve modeling of large-scale transportation networks such as evaluation of regional emergency evacuation strategies, as well as real-time control policies and *ad hoc* responses to unscheduled events.

<table>
<thead>
<tr>
<th>Name</th>
<th>Classification</th>
<th>Application Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Discrete Time</td>
<td>Discrete Event</td>
</tr>
<tr>
<td>CORSIM</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>DYNASMART</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NETFLO 1</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PARAMICS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>SYNCRO</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>VISSIM</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TRANSIMS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TransModeler*</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TRANSYT-7F</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

### 1.2 Problem Statement

The cell-transmission model (CTM), was developed by Daganzo (1994, 1995) as a solution to the Lighthill and Whitham (1955) and Richards (1956) (LWR) model. In its original form, CTM relies on a set of assumptions that limit its expandability and applicability to transportation networks at the operational analysis level. In essence, the original model requires that the network update should be performed at equal time steps for the whole simulation period and that the cell length be equal to the distance traveled at the cell free-flow speed during one simulation time step. This constraint imposes a limitation
on the selection of cell lengths, which may lead to approximation errors - if relatively large update time steps are used to improve the computational efficiency. While this limitation can be easily overcome by dividing the network links into short cells, such approach leads to small simulation time steps, and consequently, longer simulation run times.

The original form of CTM has a limited scope of applicability to realistic networks. Various network facilities, such as those with traffic control devices (signalized and stop-controlled intersections, toll plazas, etc.), were either omitted from or inadequately addressed in the model. Clearly, the ability of CTM to meet operational analysis requirements relies on its capability to model the operation of such facilities. In addition, the model lacks the expected accuracy under certain assumptions and the current literature reveals that no calibration has been attempted yet. Calibrating a theoretical model is critical for the reliability of the simulation results from real-life experimental applications. Moreover, CTM is formulated as a deterministic model, where stochastic variations (due to various individual driving behaviors) in the traffic network are not accounted for. Essentially, CTM in its original form was developed to support planning analysis rather than traffic operations applications. The model, however, has a great potential for operational analysis use, if improvements are made to overcome the current limitations and to expand its capabilities of modeling large-scale traffic networks.

1.3 Objectives

The primary goal of this research is to transform CTM from its original limited form into a more flexible and realistic traffic simulation model that strikes the balance between the realism of traffic flow representation and computational efficiency of modeling large-scale traffic networks. This research further develops CTM, by enhancing
its functionality and improving its flexibility, while retaining its effectiveness and computational efficiency in simulating large-scale traffic networks. This was achieved by relaxing the existing limitations of the original model and introducing new features such as stochastic and multi-modal modeling components, which allow for a more realistic representation of traffic evolution in traffic networks. The final product of this research is a comprehensive stochastic mesoscopic simulation tool that can be more effectively used at both planning and operational analysis stages. The following specific objectives were accomplished in this study:

1. To relax the constraint on arbitrary selection of cell length, which will facilitate the modelers to select variable cell lengths that are best aligned with the geometric configurations of the traffic network.

2. To improve the ability of the model to more accurately represent the network traffic flow. This objective was achieved by eliminating approximation errors that may result from restricting the flow advancing equations to discrete movements of vehicles, as well as introducing operational improvements of traffic flow representations at both merging and diverging junctions. In addition, specific mesoscopic features were introduced such as disaggregating the traffic flow by lanes and explicitly modeling the effects of individual lane-changing maneuvers.

3. To capture the effect of the random driving behavior in the representation of traffic flow. This objective was accomplished by replacing some of the original parameters in the analyzed network with stochastic variables.
4. To integrate a multi-modal representation of traffic. Specifically, changes to
the fundamental modeling equations are made to capture the effect of various vehicles
types that compose real-life traffic streams.

5. To develop and implement a fully functional software module that
encompasses all improvements made to CTM.

6. To evaluate the performance of the improved model using a real-life network
simulated with CORSIM, an extensively used microscopic traffic simulation model.

This dissertation document is organized as follows:

Chapter 1 introduces the topic of this study, emphasizes the problem statement
and lists the objectives of this study.

Chapter 2 contains background information on CTM. The first part presents a
literature review of previous studies that employed CTM in various applications. The
second part of the chapter presents details about the topological conventions in CTM.

Chapter 3 introduces the development of the topological enhancements in CTM
that allows for variable cell length selection. In addition, the effects in the simulation
results of a special operational improvement that allows for non-discrete vehicle
movements are investigated.

Chapter 4 presents another operational improvements that allows modeling traffic
flows inside cell across individual lanes. In this chapter two lane-changing algorithms are
developed and tested.

Chapters 5 and 6 address two operational improvement of traffic flow at the
merging and diverging junctions, respectively. Numerical examples demonstrate that the
newly developed merging and diverging algorithms help in capturing more accurately
traffic flows under various traffic conditions, especially when non-homogeneous
distribution across lanes inside cells occur.

Chapter 7 introduces a new methodology to account for random driving behavior
in CTM. A calibration exercise shows how microscopic vehicle trajectory data can be used
to model dynamic variations in backward moving wave speed.

Chapter 8 presents a modification to the CTM equations that allows explicit
modeling of multimodal flows.

Chapter 9 is dedicated to the results of a final performance comparison with a
microscopic simulator, CORSIM. In addition, this chapter presents specific details on the
special software module that integrates all the enhancements in CTM.

Chapter 10 presents concluding statements and identifies directions for future
research work.
CHAPTER 2  BACKGROUND

The diversity and availability of traffic simulation tools make simulation an attractive approach for the transportation community, especially if analytical solutions are not suitable or require an extensive amount of effort. Therefore, the last two decades have witnessed a surge of published reports and research papers on various traffic simulation related subjects.

2.1 Literature Review

Daganzo (1994, 1995) introduced a discrete approach for predicting the evolution of traffic over time and space without the need for complex shockwave calculations. The approach transforms the differential equations of the LWR hydrodynamic model into simple difference equations. The LWR model provides reasonable approximation of traffic flow evolution in realistic networks. CTM was sufficiently validated by field data in Lin and Daganzo (1994), Lin and Ahanotu (1995). Daganzo’s model divides the transportation network into small homogeneous and interconnected segments (referred to as cells), and assumes piecewise linear relationships between flow and density at the cell level. Despite its simplicity, the cell-transmission model (CTM) is able to describe and accurately capture traffic propagation phenomena such as disturbances and shockwaves in traffic networks. Although CTM belongs to the class of macroscopic simulation models, it is relatively easy to transform CTM into a mesoscopic model, where vehicles can be individually tracked between origins and destinations for possible dynamic traffic assignment applications.
Recently, Sun et al. (2006) developed a parameter calibration methodology using a piecewise-linearized version of the cell transmission model. The authors’ approach investigates the potential of CTM to be used in a mode switch pattern defined as a set of combinations of free-flow and congested conditions that may characterize a freeway section at upstream and downstream locations. The results of the study demonstrated good replication of real-life traffic conditions. However, the authors recognized some limitations of the proposed model, such as the need for a priori definition of switching modes or the limitation to the representation of relatively short freeway sections due to the capturing of single-wave traffic behavior only.

The recent literature shows several research studies that used CTM to support various applications such as dynamic traffic assignment problem, dynamic network design problem, signal timing optimization, travel time prediction, etc. The research on each of these application types are reviewed in the following subsections.

### 2.1.1 Dynamic Traffic Assignment Applications

Several other CTM-based applications investigated additional features of DTA. For example, Li et al (1999) examined the solution of a SO DTA problem in which drivers fix their arrival times, rather than their departure times. Their model is formulated as an LP solution for CTM that simultaneously optimizes departure time and route choice. The authors concluded that the model, which applies to multi-origin/multi-destination problems, generally preserved the first-in-first-out (FIFO) principle. However, one can construct pathological cases where FIFO principle is violated. Golani and Waller (2004) used CTM to develop an algorithm to solve the user-optimal dynamic traffic assignment (UO DTA) problem. The authors based their formulation on CTM to overcome the
drawbacks of most DTA models, which often fail to adequately capture all realities of the street networks because of simplification. The authors concluded that their approach guarantees a user optimal solution for a single-destination problem.

Ziliaskopoulos (2000) demonstrated that CTM could be used to solve and give insight to the single destination system optimum dynamic traffic assignment (SO DTA) problem. The author showed that a linear program (LP) approach can be applied to solve the SO DTA problem for a small network and introduces the concept of marginal travel time in a dynamic network along with system optimum necessary and sufficient conditions. However, the approach was not presented as an operational model for actual applications.

Ziliaskopoulos et al. (2004) demonstrated the CTM applicability within the simulation based DTA context for large-scale realistic networks. The authors investigated the implementation challenges such as modeling turning movements, computation of travel times and alike, while coding an urban network part of the Columbus, OH MPO. The study engaged various tests to evaluate the performance of the developed DTA methodology and the suitability of the model for applications such as infrastructure improvement evaluation, congestion pricing, and effectiveness of information provision systems.

Karoonsoontawong and Waller (2005) used Daganzo’s cell transmission model to compare two stochastic dynamic network design models under user-equilibrium (UE) and system-optimum (SO) assumptions. The authors selected CTM as the foundation of their methodology because of its efficient and realistic representation of traffic flow. The study investigated the model behavior under stochastic and temporal variation assumptions at the demand level as well as cell storage capacity and cell flow capacity. The authors
concluded that stochastic behavior is always better for SO models, while the UE models may lead to some erroneous solutions under certain circumstances. Some of the failures identified in the developed model may occur due to the lack of calibration of the stochastic variables. The overlooked calibration may affect the validity of the initially assumed fundamental traffic flow diagram for the model.

2.1.2 Network Design Problem Applications

A few other studies took advantage of the feasibility of LP formulation with CTM in solving the network design problem (NDP) under various DTA formulations. For example, Waller and Ziliaskopoulos (2001) introduced a chance-constraint and two-stage LP formulation based on SO DTA to model the continuous NDP that occurs when the origin-destination time-dependent demands are random variables with known probability distributions. The authors also used the finite difference equations of CTM as a solution for the LWR traffic flow model.

Another study by Ukkusuri and Waller (2004) used CTM to capture the time dependent traffic characteristics and proposed a NDP solution approach that could be used for dynamic planning applications. The authors developed and compared the user-optimal (UO) and system-optimum (SO) NDP solutions in the context of single destination problem. The authors recognized the limitations of the presented formulation, but emphasized that LP modeling of the NDP under dynamic traffic conditions was only possible because of the availability of CTM.
2.1.3 Traffic Operations Applications

Studies by Lo (1999, 2001), and Lo and Chow (2004) used CTM to evaluate the design of intersections timing plans. The authors showed that a mixed integer programming approach could be successfully applied to model the intersection timing under different traffic conditions. They showed that the model produced signal timing consistent with models for unsaturated conditions. For gridlock conditions, however, it produced a timing plan that works better than conventional queue management practices. The studies concluded that green progression is possible for a wide range of traffic demands, including congested and gridlock conditions. In a later study by Lo et al. (2001), a dynamic traffic control formulation was developed to derive timing plans for time-variant traffic patterns. The study applied genetic algorithms to optimize signal timing under congested traffic conditions.

In a recent study by Bear and Ziliaskopoulos (2006), CTM was used in developing a system optimal signal optimization formulation. The authors considered constraints such as oversaturated traffic conditions, turning movement and gap acceptance, and adaptive and/or pre-timed traffic controllers. In addition, special conditions such as incidents and road closures are treated. The proposed solution in this study is a linear mathematical program, but this approach may lead to holding vehicles on links unrealistically in order to optimize the total network travel time. Another limitation is that the development of heuristic approaches is necessary to overcome the computationally expensive modeling of real-life networks.

Another study by Mark and Sadek (2004) attempted to use CTM in traffic operations applications. The study made use of Daganzo’s original development to
construct a travel-time prediction system using soft computing tools such as artificial neural networks (ANN). To prove the effectiveness of their approach, the authors selected CTM for its fast execution time and inherent capability of modeling traffic flow realities. The major finding was that, with correct input and careful selection of network parameters, ANNs were able to predict travel times reasonably well under transient traffic conditions with the presence of accidents. The study demonstrated the potential use of CTM to support traffic operations as the input traffic data was collected from roadway sensors in real-time.

Another study by Ziliaskopoulos and Lee (1997) attempted to relax the constant cell size requirement and extend the applicability of CTM to signalized intersections. However, the approach presented lacks consideration of possible queuing conditions within large cells and the resulting non-uniformity or non-stationarity of traffic conditions. Rather, the approach assumed uniform distribution of traffic within the cell, even under mixed traffic conditions. This may lead to misrepresentation of the actual propagation of traffic flow and the possible underutilization of the actual roadway capacity. Moreover, the approach limits the length of large cells to be multiples of the length of the other smaller cells. In addition, modeling signalized intersections suffers from deficiencies in terms of cell length constraints, as well as modeling the gap acceptance for permitted turn movements.

### 2.1.4 Emergency Evacuation Applications

CTM was employed in several research studies as an underlying methodology for solving various problems related to regional evacuation strategies. For example, Tuydes and Ziliaskopoulos (2004) introduced a network evacuation model integrated within a
dynamic traffic assignment model based on CTM. The authors optimized the system travel time, while simultaneously computing the optimal capacity reversibility in the network. The results described in the show that a significant reduction in system travel time will be achieved when reversible lanes are implemented. However, the study did not account for the effect of various implementation issues such as lane-base versus all-or-nothing reversibility cases or the various costs related to reversing a road segment. The authors recognized that accounting for these problems might enhance the realism of the proposed model.

In another study, Liu et al. (2006) developed an evacuation optimization process based on the staged evacuation concept. The applicability of the model is limited by several assumptions such as the area subject to evacuation being composed of distinct and concentric zones with the zone having highest level of severity in the middle or the loading demand pattern is known a priori. Despite these limitations, the proposed model features a viable alternative to existing evacuation strategies provided that one can find solutions to account for the impact of human behavior in the model.

2.1.5 Hybrid Simulation Applications

Shi et al. (2006) designed and implemented a hybrid simulation model that combines the cell transmission concept with a microscopic control theory model. The integrated model is claimed to be suitable for regional simulation by allowing modeling various levels of detail to the user’s discretion. The authors developed a complex methodology to process traffic flow propagation at the interfaces microscopic and mesoscopic links. The feasibility of the approach was tested with a small traffic network and the study showed consistent traffic flow representation under both congested and free
flow conditions. However, the authors recognize a possible synchronization error at the interface between microscopic and mesoscopic links, which is estimated that the impact is generally not very significant.

2.2 The LWR Model

Arguably, the LWR kinematic wave theory, developed by Lighthill and Whitham (1955) and Richards (1956), remains the most reasonable approximation of the macroscopic behavior of traffic flow. Essentially, the LWR model explicitly uses macroscopic variables of flow \( (q) \) and density \( (k) \), and is valid for the whole range of the fundamental q-k diagram, including shockwaves and queue formation and dissipation in both congested and un-congested regimes. Although multiple solutions to the LWR wave theory may exist, only one solution is considered physically relevant and is sought by most solution methods such as CTM. First, a brief review of the LWR model is presented in this section, followed by a detailed description of CTM.

For a homogeneous highway segment with no change in net flow, the vehicle conservation Eq. must apply such that:

\[
\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = 0
\]  

(1)

Where \( q_x \) is the rate of change of flow with respect to space \( (x) \) and \( k_t \) is the rate of change of density with respect to time \( (t) \). In addition to the flow conservation Eq. in (1), the assumed relationship between flow and density is a continuous and piecewise differentiable function of state in the form:

\[
q = f(k, x, t)
\]  

(2)
Given the Eq. of state in (2) and a set of initial or boundary conditions, solutions for the LWR model can be obtained in the form of density over time and space, $k(x,t)$. The solution, however, is not defined along the paths of shockwaves, where discontinuity in density is encountered. A common procedure to obtain solution for the LWR model is the method of characteristics, which are considered causality lines indicating how points influence each other in the time-space domain (see for instance, Daganzo, 1997). The method, however, requires locating the shocks in the time-space domain. In the traditional method of solution, the speed of a shockwave is determined by:

$$U = \frac{q_b - q_a}{k_b - k_a}$$ (3)

Where, $(q_a, k_a)$ and $(q_b, k_b)$ are the flows and densities at points $a$ and $b$, respectively and located on opposite sides of the shock. This solution method however, is considered rather lengthy for complicated networks. Other methods were demonstrated in the open literature (e.g., Luke, 1972; Newell, 1993; Daganzo, 1994). The cell-transmission model developed by Daganzo (1994) offers a solution for the LWR model using a finite difference equations (FDE) method, which approximates the partial differential Eq. (PDE) of the LWR model. The next section explains in detail how CTM works.

### 2.3 Network Topology for CTM

According to Daganzo (1994, 1995), the building block of CTM is the *cell*, which represents a homogeneous segment of the highway. Each link in a traffic network is divided into one or more interconnected cells, as depicted in Figure 1. Each pair of cells may be interlinked with one connector only; multiple connectors between an upstream cell
and a downstream cell are not permitted. Connectors have no physical dimensions and are mere representations of the interface between two consecutive cells. The cell length is not arbitrarily chosen, but is equal to the distance traveled by vehicles in one simulation time step at the free-flow speed of the cell. This ensures that in each simulation update interval, vehicles cannot skip cells along the direction of their travel. In other words, vehicles can advance at most one cell under free-flow conditions. For instance, if the free-flow speed in a cell is 60 mph and the desired simulation time step is assumed 10 seconds, then the cell length has to be equal to 1/6 of a mile. This is necessary to ensure that the vehicle conservation equation is satisfied for all cells in the network.

Figure 1: An example of the basic building block of CTM

The network topology for CTM is summarized in this section based on the succinct representation adopted from Ziliaskopoulos (2000). Given a network of a set of cells, $\mathcal{C}$, and a set of connectors, $\mathcal{E}$, each cell/connector is classified into one of five distinct types: ordinary, merging, diverging, source, or sink. Each cell $j$ in the set $\mathcal{C}$ may be connected to a set of predecessor cells, $\Gamma^{-1}(j)$, and a set of successor cells, $\Gamma(j)$. The type of each cell is determined by the number of predecessor and successor cells. Cell $j$ is defined as ordinary if $|\Gamma^{-1}(j)| = 1$ and $|\Gamma(j)| = 1$, as shown in Figure 2a. If $|\Gamma^{-1}(j)| > 1$ and $|\Gamma(j)| = 1$, then cell $j$ in Figure 2b is of the merging type, since it receives inflows from
more than one predecessor cell. An example of merging cells can be found at freeway on-

ramp junctions.

Similarly, if $|\Gamma^{-1}(j)| = 1$ and $|\Gamma(j)| > 1$, then cell $j$ in Figure 2c is diverging, since

it sends outflows to more than one successor cell. Diverging cells are locations where
drivers must make a decision on which route to take to their final destination. In
simulation, such decision can be based on static assignment methods where turning ratios
are known a priori, or determined by route choice models (shortest remaining path to
destination) in dynamic traffic assignment applications.

Cells can also be defined as either source cells, if $|\Gamma^{-1}(j)| = 0$ and $|\Gamma(j)| = 1$, or
sink cells, if $|\Gamma^{-1}(j)| = 1$ and $|\Gamma(j)| = 0$. Source cells represent the entry points to the
network where traffic is released from an assumed time-dependent origin-destination
demand matrix. Sink cells serve as the final destinations for all vehicles in the network.
Clearly, all trips must begin at source cells and end at sink cells during the network
simulation time.

Connectors are also classified as ordinary if both end cells are ordinary, merging
if the upstream end cell is ordinary and the downstream end cell is merging, diverging if
the upstream end cell is diverging and the downstream end cell is ordinary, source if the
upstream end cell is source, and sink if the downstream end cell is sink. A cell or
connector cannot be of the merging and diverging type simultaneously, as shown in Figure
3(a), since this would further complicate the flow advancing equations. An equivalent,
valid representation can be made by splitting the cell into one merging and another
diverging cell, as shown in Figure 3(b).
(a) An example of an ordinary cell $j \left( \Gamma^{-1}(j) = 1, \Gamma(j) = 1 \right)$

(b) An example of a merging cell $j \left( \Gamma^{-1}(j) > 1, \Gamma(j) = 1 \right)$

(c) An example of a diverging cell $j \left( \Gamma^{-1}(j) = 1, \Gamma(j) > 1 \right)$

Figure 2: Variation of cell types in CTM network topology

(a) Invalid representation: simultaneous merging and diverging maneuvers

(b) Valid representation: separation of merging and diverging maneuvers

Figure 3: CTM topological restrictions
2.3.1 Flow Advancing Equations of CTM

In each cell, CTM assumes a piecewise linear relationship between flow and density. This relationship is depicted in Figure 4 and expressed by:

\[ q = \min\{V_k, Q, W(K - k)\} \quad \forall 0 \leq k \leq K \]  

(4)

where,

\( V = \) the free-flow speed of the highway segment,

\( K = \) the jam density,

\( Q = \) the maximum flow rate (capacity), and

\( W = \) the speed of backward moving waves.

Modification to the previous equation leads to the following flow advancing equations for each type of cell and connector.

![Figure 4: Piecewise Linear Approximation of the q-k Relationship](image-url)
2.3.1.1 Ordinary Cells and Connectors

The flow advancing equation for an ordinary connector \((i, j)\) is:

\[
y_{ij}(t, t + \Delta t) = \min\{x_i(t), \min[Q_i, Q_j], \delta_j(X_j - x_j(t))\} \quad \forall (i, j) \in \mathcal{E}_e,
\]

where,

\[
y_{ij}(t, t + \Delta t) = \text{the number of vehicles advancing from cell } i \text{ to cell } j \text{ in one simulation time step } \Delta t \text{ at time } t,
\]

\[
\mathcal{E}_e = \text{the set of ordinary cells},
\]

\[
x_i(t) \text{ and } x_j(t) = \text{the occupancy of cells } i \text{ and } j \text{ at time } t,
\]

\[
X_j = \text{the maximum occupancy of cell } j,
\]

\[
Q_i, Q_j = \text{the flow capacity of cells } i \text{ and } j, \text{ during } \Delta t, \text{ and}
\]

\[
\delta_j = \begin{cases} 
1 & \text{if } x_i(t) \leq Q_j \text{(free-flow conditions)} \\
\frac{W_j}{V_j} & \text{otherwise (forced-flow conditions)}
\end{cases}
\]

For simplicity of notations, the time variable \(t\) is omitted from forthcoming equations; thus, equation (5) reduces to:

\[
y_{ij} = \min\{x_i, Q_i, Q_j, \delta_j(X_j - x_j)\} \quad \forall (i, j) \in \mathcal{E}_e.
\]

2.3.1.2 Merging Cells and Connectors

For each merging cell \(j\) in the set of merging cells \(\mathcal{E}_m\), the inflows on merging connectors are determined by the simple linear program:

\[
\max \sum_{\forall i \in \Gamma^{-}(j)} y_{ij}
\]

subject to:
The equations above account for specific priorities on merging connectors, $\rho_i$, and thus, establish quantitative rules for assigning right of way to merging vehicles from roads with different hierarchies, as shown in equation (9).

$$\rho_i \leq 1, \quad \forall i \in \Gamma^{-1}(j)$$

$$\sum_{\forall i \in \Gamma^{-1}(j)} \rho_i = 1$$  \hspace{1cm} (9)

It should be also noted that the type of outgoing connector from a merging cell is ordinary, and therefore, its flow is determined by equation (6).

2.3.1.3 Diverging Cells and Connectors

For each diverging cell $j$ in the set of diverging cells ($\mathcal{E}_d$), the outflows on diverging connectors are determined by the simple linear program:

$$\max \sum_{\forall j \in \Gamma(i)} y_{ij},$$ \hspace{1cm} (10)

subject to:

$$y_{ij} \leq \min \{r_j Q_j, r_j \delta_j (X_j - x_j)\}, \forall j \in \Gamma(i)$$ \hspace{1cm} (11)

$$\sum_{\forall j \in \Gamma(i)} y_{ij} \leq \min \{x_i, Q_i\}$$ \hspace{1cm} (12)

$$r_j \leq 1, \quad \forall j \in \Gamma(i)$$

$$\sum_{\forall j \in \Gamma(i)} r_j = 1$$ \hspace{1cm} (13)

The previous equations above assume that the turning ratio $r_j$ from cell $i$ to cell $j$ is known a priori. Other settings can also be made to determine the diverging flows using route choice models (e.g., shortest path from the diverging cell to final destination of...
each vehicle). The type of incoming connector to a diverging cell is ordinary and therefore its flow is determined by equation (6).

### 2.3.2 The Flow Conservation Equation

CTM applies a conservation equation of the flow modeled in the form:

\[
x_j = x_j + \sum_{i \in \Gamma^{-1}(j)} y_{ij} - \sum_{k \in \Gamma(j)} y_{jk} \quad \forall j \in \mathcal{E}.
\]  

(14)

The above recursion applies to all types of cells once the connector flows are determined by the appropriate flow advancing equations. It should be noted that the left side of the equation updates the occupancy of the cell at the end of the current simulation time step, i.e., at \( t + \Delta t \), based on the occupancy at time \( t \) and the difference between the sum of inflows and outflows. Equation (14) ensures that the flow conservation equation is satisfied for all cells at any time.

### 2.3.3 Advantages and Disadvantages of CTM

The literature review reveals that CTM has several noticeable advantages. The model is relatively simple and sufficiently accurate for planning analysis purposes. In addition, its macroscopic nature leads to higher computational efficiency and less calibration efforts. Several studies demonstrated that the model enhances the realism of the traffic flow representation for a variety of applications including both static and dynamic traffic assignment procedures. Two efficient modeling approaches were identified in conjunction with CTM, linear mathematical programming and parallel computing. The appropriateness of the two approaches is derived from the fact that simulation results are independent of the order in which the cells are updated at each simulation time step. Parallel computing may introduce an overhead at the implementation level, but provides
efficient computational alternatives of large traffic networks. Overall, the model is a good fit for a wide range of applications, including analysis of disaster evacuation strategies at the planning level.

Nevertheless, several limitations of CTM must be overcome to address more complex operational analysis needs. For example, the model in its original form requires that the network be decomposed into cells with length corresponding to the simulation time step. This limitation may not be appropriate for large-scale networks due to possible geometric restrictions or inconsistencies when dividing the network links into cells, and the additional memory requirements when dividing the network into too many small cells. A consequence of the fixed cell length is the constant simulation time steps throughout the entire simulation period. This limitation may lead to computationally inefficient modeling of large-scale networks – especially when combinations of small cells and various facility types are used, such as freeways and intersections on city streets. Moreover, the model is deterministic and therefore, does not allow modelers to study the effect of stochastic variations on the simulation results. Some of these limitations are addressed in this research study, while others are planned for future development.
CHAPTER 3 VARIABLE CELL LENGTH MODIFICATIONS

3.1 Introduction

This chapter presents a methodology that allows modeling a variable cell length in CTM. Long cells are split into subcells and specific changes in the existing formulation of the flow advancing equation of the model are made. As a result, the concept of wait times associated with vehicle platoons inside the cells is introduced. It is shown that the flow advancing equation can be used to advance vehicle through the network between the cells and between the subcells inside the long cells. The last section of the chapter presents the implications of using non-discrete vehicle movements.

3.2 Subcells

Recall that CTM assumes homogeneous traffic conditions inside the cells. A simple way of accounting for variation in the length of the cells while preserving the homogeneity assumption is to split them into virtual subcells of equal length. This approach was previously suggested by Ziliaskopoulos and Lee (1999), but the authors limited the representation of cells in integer multiples of an arbitrarily selected subcell. With this approach, not any cell length can be represented in a traffic stream, unless the selected subcell is equal to the unit distance. Using very small subcells translates in a computational burden. However, a more efficient solution is developed and tested in this study. This solution uses the same discretization of cells into equal length subcells, called base subcells, but only the last subcell is allowed to be shorter. Hence, the base subcell
may have an arbitrarily selected length, without any constraint on the length of the cells. The flow advancing equation needs to be modified to account for two types of vehicle movements. First, the flow advancing equation is used to estimate the number of vehicles that are allowed to advance externally between two interconnected cells during each simulation update. Second, the flow advancing equation is used to advance vehicle platoons internally inside the long cells, between subcells. The following sections present details on how the flow advancing equation applies in for each type of vehicle movements.

### 3.2.1 Internal Update for Subcells

Consider cell $i$ with free-flow travel time $\tau_i$ and $n_i$ subcells, as shown in Figure 5. The number of subcells is determined by $n_i = \left\lfloor \frac{\tau_i}{\tau} \right\rfloor + 1$. Therefore, all subcells except the last one, can be of equal size such that their free-flow travel times are identical to the simulation update interval, $\tau$.

The free-flow travel time of the last subcell ($\tau_i'$) is determined by $\tau_i' = \tau_i - (n_i - 1)\tau$ such that $0 \leq \tau_i' < \tau$.

![Figure 5: Discretization of long cells into subcells](image-url)
Using the original flow advancing model in CTM, the number of vehicles that will advance between subcell $m$ and subcell $m+1$ ($\forall m=1,...,n_i-2$) of cell $i$, is given by equation (15).

$$y_{m\to m+1,i} = \min\left[ x_{m,i}, Q_i \tau, \delta_{m\to m+1,i} \left( X_{m+1,i} - x_{m+1,i} \right) \right]$$ \hspace{1cm} (15)

Where,

$x_{m,i} =$ the number of vehicles occupying subcell $m$,

$X_{m+1,i} =$ the maximum space capacity of subcell $m+1$, at jam density conditions,

$Q_i =$ flow capacity of cell $i$,

$\delta_{m\to m+1,i} =$ a congestion indicator for subcells $m$ and $m+1$,

$$\delta_{m\to m+1,i} = \begin{cases} \frac{W_i}{V_i} & \text{Otherwise} \\ 1 & x_{m,i} \leq \tau Q_i \end{cases}$$

$V_i$, $W_i =$ the free-flow and backward moving wave speeds in cell $i$

For $m = n_i-1$, the flow advancing equation is adjusted to account for possible variations in the free-flow travel times of the last subcell in cell $i$. The procedure differs by the type of successor cell to cell $i$ as explained next.

### 3.2.1.1 Special Treatment of the Last Subcell in Ordinary and Merging Cells

The following procedure is used to estimate the number of vehicles that will advance between subcell $n_i-1$ and subcell $n_i$ for ordinary or merging cell $i$ that is followed by any type cell $j$.

$$y_{n_i-1\to n_i,j} = \min\left( x_{n_i-1,j}, Q_j \tau, x_{n_i,j}' \right) - \left( y_{j}' - \psi_{j} \right)$$ \hspace{1cm} (16)
and
\[ x_{n_i}^r = \delta_{n_{i-1} \rightarrow n_i, j} (X_{n_i,j} - x_{n_i,j}) + \left(1 - \frac{r_i}{\tau} \right) \min \left[ \tau Q_j, \delta_{ij} (X_{1,j} - x_{1,j}) \right], \tag{17} \]

Where,
\[ \delta_{ij} = \begin{cases} 
1 & x_i^r \leq \tau \min(Q_i, Q_j) \\
\frac{w_j}{V_j} & \text{Otherwise}
\end{cases}, \]

\[ \psi_{ij} = \min \left( y_{ij}, x_{n_i,j} \right), \tag{18} \]

\[ y_{ij} = \text{total flow advancing out of cell } i \text{ into cell } j, \]

\[ \psi_{ij} = \text{the flow advancing out of subcell } n_i \text{ into cell } j, \]

\[ x_{n_i-1} = \text{the number of vehicles occupying subcell } n_i - 1, \]

\[ x_{n_i}^r = \text{the number of vehicles spaces available in the last subcell in cell } i \text{ and the first subcell in cell } j, \text{ usable by the vehicles advancing from subcell } n_i. \]

Due to the possible shorter length of the last subcell, \( n_i \), some of vehicles advancing out of next to last subcell, \( n_i - 1 \), may reach into the first subcell of the connecting downstream \( j \). These vehicles are calculated by the second term, \( (y_{ij} - \psi_{ij}) \), of equation (16). To find out the number of vehicles advancing internally between \( n_i - 1 \) and \( n_i \), from the total number of vehicles advancing out of \( n_i - 1 \), represented by \( \min \{ x_{n_i-1}, Q_j \tau, x_{n_i}^r \} \), one has to subtract \( (y_{ij} - \psi_{ij}) \).
3.2.1.2 Special Treatment of the Last Subcell in Diverging Cells

Similarly, updating internally the diverging cells on the last internal connector has to account for a possible shorter last subcell. The number of vehicles advancing between can be calculated with the equation (19)

\[ y_{n-1 \rightarrow n,j} := \min(x_{n-1,j}, Q_i \tau, x'_{n,j}) - \sum_{j \in \Gamma(i)} (y_{ij} - \psi_{ij}) \]  

(19)

\[ x'_{ij} = \delta_{n-1 \rightarrow n,i} (X_{n,j} - x_{n,j}) + \left(1 - \frac{\tau_i}{\tau}\right) \sum_{j \in \Gamma(i)} \min\left[\tau Q_j \delta_{ij} (X_{1,j} - x_{1,j})\right] \]

(20)

Where,

\[ \psi_{ij} = \min(y_{ij}, x_{n,j}) \quad \forall j \in \Gamma(i) \]  

(21)

\[ \delta_{ij} = \begin{cases} 1 & x'_{i \rightarrow j} \leq \tau \min(Q_i, Q_j) \\ \frac{w_j}{V_j} & Otherwise \end{cases} \quad \forall j \in \Gamma(i), \]  

(22)

\[ x'_{i \rightarrow j} = \text{the number of vehicles eligible to advance from cell } i \text{ to cell } j, \]

\[ x'_{n,i} = \text{the number of vehicles spaces available in the last subcell in cell } i \text{ and the first subcell in each cell } j, \text{ usable by the vehicles advancing from subcell } n_j. \]

Equation (19) is similar with equation (16) with the exception of the second term. The second term contains the total number of vehicles advancing out of the next to last subcell into each of the downstream successor cells of cell i.

3.2.2 External Update for Cells

The flow advancing equation of CTM controls the number of vehicles advancing externally between any two interconnected cells, i and j, at each simulation update, as shown in equation.
\[
y_0 = \min \left[ x'_i, \tau Q_i, \tau Q_j, \delta_j x'_j \right] \tag{23}
\]

Where,

\(x'_i\) = the number of vehicles eligible to advance out of cell \(i\),

\(\tau\) = the simulation update interval,

\(Q_i, Q_j\) = flow capacity of cells \(i\) and \(j\), respectively

\(x'_j = (X_j - x_j)\) = the number of vehicle spaces available in cell \(j\)

\(X_j\) = the maximum space capacity of cell \(j\), at jam density conditions,

\(x_j\) = the number of vehicles occupying cell \(j\),

\(\delta_j\) = a congestion indicator for cell \(j\),

\[
\delta_j = \begin{cases} 
1 & x'_i \leq \tau Q_i \\
\frac{W_i}{V_i} & Otherwise 
\end{cases}
\]

\(V_i, W_i\) = the free-flow and backward moving wave speeds in cell \(i\)

Assuming the length of the last subcell in cell \(i\) equal to the typical subcell, the first term in the flow advancing equation, \(x'_i\), can be estimated by the number of vehicles occupying the last subcell in cell \(i\). However, short subcells may occur in any cell and this estimate of \(x'_i\) cannot be used in equation (23) because of the lack of accuracy. Therefore, a better solution was developed to ensure accuracy of the simulation results regardless of the last subcell length.
3.2.3 Vehicle Wait Times

In CTM vehicles are allowed to advance out of cell \( i \) if they occupied that cell, or waited inside the cell, for at least \( \tau_i \) time units, where \( \tau_i \) is the cell’s free-flow travel time. To facilitate the calculation of the vehicles eligible to advance out of cell \( i \), one needs to estimate the wait times of the vehicle platoons occupying the cell. In Figure 7, it is shown an example of wait time function for a cell \( i \), where \( \phi_{m,i}^i \) represents the total number of vehicles that, at simulation time \( t \), already spent inside the cell \( i \) a time greater than or equal to \( \tau_i' + (m-1)\tau \), where \( \tau_i' \) represents the free-flow travel time of the last subcell, and \( \tau \) is the simulation update interval. Hence, the number of vehicles eligible to advance out of cell \( i \) can be determined by \( \phi_{1,i}^i \), where \( \phi_{n-1,i}^i \) represents the total number of vehicles with wait time in cell \( i \) greater than or equal to \( \tau_i - \tau \) at the beginning of any update. For each cell, the computation of the wait time function depends on the structure and the number of the predecessor and successor cells.

3.2.3.1 Computation of Vehicle Wait Times for Ordinary Cells

At the end of each simulation update, the wait time function is adjusted based on the current information about the traffic flows between the cells. Therefore, a recursive method is applied to update \( \phi_{m,i}^i \), for ordinary cell \( i \), in a decreasing order of \( m \) (the index of subcells in cell \( i \)).

\[
\phi_{m,i}^i = \phi_{m-1,i}^{i} - y_{ji}^i \quad \forall m = n_i - 1, ..., 2
\]

(24)

\( \phi_{i,i}^i \) represents the number of vehicles that have a wait time in cell \( i \) greater or at least equal to \( \tau_i' \) and estimating its correct value has to account for the length of the last
Consider the example shown in Figure 6, where cell \( i \) is connected with an upstream cell, \( k \), and a downstream cell, \( j \). For this example, \( \phi_{i,j}' \) is calculated by equation (25).

\[
\phi_{i,j}' = x_{i,t}^t - y_{ij}^t + \alpha_{ki}
\]  

(25)

Where

\[
\alpha_{ki} = \begin{cases} 
0 & \text{Case I} \\
\left(1 - \frac{\tau_i'}{\tau_k} \left(1 - \frac{y_{ki}'}{y_{ki}}\right)\right) y_{ki}' & \text{Case II}' \end{cases}
\]  

(26)

\( n_i \) = the number of subcells in cell \( i \),

\( y_{ki}^t \) = the number of vehicles advancing from \( k \) into \( i \) at the end of update \( t \),

\( y_{ij}^t \) = the number of vehicles advancing from \( i \) into \( j \) at the end of update \( t \),

\( y_{ki}' = \min(y_{ki}^t, x_{n_i-k}^{t-2}) \) = the number of vehicles advancing from the last subcell in cell \( k \) into cell \( i \) (\( y_{ki}' \leq y_{ki}' \)).

Figure 6: External update for ordinary cells
If the last subcell in cell $k$ is shorter than the typical subcell length, each platoon that advances out of the next to last subcell may split into two parts: A front part that advances into the first subcell in cell $i$ and a rear part that advances into the last subcell in cell $k$. The update of the wait time function after external movement is graphically illustrated in Figure 7.

![Figure 7: Update of wait time function after external movements into an ordinary cell](image.png)

The head of the front platoon advancing from cell $k$ can be located along the horizontal line while its tail is always anchored to the zero wait time. Depending on the number of vehicles that advance into cell $i$ from the last subcell in $k$, $\psi_{ki}$, two cases may arise as shown. Given the size of the last subcell in $i$ the number of vehicles with wait time greater than or equal to $\tau_i'$ can be estimated. Note that the two cases are dynamically
changing because of their dependence on $\psi_{ki}$ and $y_{ki}$, therefore, a correction may or may not be needed in every simulation update. Note that $\psi_{ki}$ is bounded by 0 and $y_{ki}$. If zero, no vehicles advance out of cell $k$; if $y_{ki}$ then all advancing vehicles are sent from the last subcell. An example of a possible wait time function is depicted in Figure 8.

![Figure 8: Example of a wait time functions for cell $i$](image)

### 3.2.3.2 Computation of Vehicle Wait Times for Merging Cells

Similarly, the same method can be applied to compute $\phi_{i,j}'$ for a merging cell, $i$. In particular, $\phi_{i,j}'$ accounts for the total effect of the variable last subcell in each predecessor cell, $k$, as shown by the last term in equation (27):

$$
\phi_{i,j}' = x_i^{j-i} - y_{j}^{'} + \sum_{k \in \Gamma^{-}(i)} \alpha_{ki}
$$

(27)
3.2.3.3 Computation of Vehicle Wait Times for Diverging Cells

For diverging cells, the wait time function is defined by target connectors such that \( \phi_{m,i \rightarrow j}^{t} \) represents the total number of vehicles destined for the successor cell \( j \) with wait time greater than or equal to \( \tau_{i}^{t} + (m-1)\tau \) at time \( t \). The following recursive functions are used to update the wait time function, \( \phi_{m,i \rightarrow j}^{t} \), for cell \( i \) \( \forall j \in \Gamma(i) \).

\[
\phi_{m,i \rightarrow j}^{t} = \phi_{m-1,i \rightarrow j}^{t} - y_{ij}^{t} \quad \forall m = n_{i} - 1, \ldots, 2 \tag{28}
\]

\[
\phi_{1,i \rightarrow j}^{t} = x_{i \rightarrow j}^{t} - y_{ij}^{t} + \alpha_{ki \rightarrow j} \tag{29}
\]

Where

\[
\alpha_{ki \rightarrow j} = \begin{cases} 
0 & \text{Case I} \\
(1 - \frac{\tau_{i}^{t}}{\tau - \tau_{i}^{t}(1 - \psi_{ki \rightarrow j}^{t})}) y_{ki \rightarrow j}^{t} & \text{Case II} 
\end{cases} \tag{30}
\]

\( x_{i \rightarrow j}^{t} \) = the vehicles occupying cell \( i \) with destination cell \( j \),

\( y_{ki \rightarrow j}^{t} \) = the vehicles advancing into cell \( i \) with destination cell \( j \),

\( \psi_{ki \rightarrow j}^{t} = \min(y_{ki \rightarrow j}^{t}, x_{k \rightarrow i}^{t-\tau}) \) = the number of vehicles advancing from the last subcell in cell \( k \) into cell \( i \), with destination \( j \).

It can be seen that calculating \( \phi_{i,j}^{t} \) for a diverging necessitates an estimate of diverging flows inside cell \( i \). It is assumed that for each vehicle entering a diverging cell the next target cell is known. Hence, the flows by target cells can be estimated. An example of the wait time function for a diverging cell \( i \) and a target cell \( j \) is depicted in Figure 9 below.
3.2.4 Determination of $x_j'$

The number of available spaces in cell $j$ during one simulation update is denoted by $x_j'$. The magnitude of this term bounded by the storage capacity of a cell under jam density conditions, $X_j$ since $\tau \leq \tau_j$. Because the network update time is equal to the free-flow travel time of the base subcell, the number of available vehicle spaces is given by the difference between the maximum vehicle storage of the last subcell (jam density conditions) and the number of vehicles occupying cell $j$, as described in equation (31).

$$x_j' = X_j - x_j$$  (31)

This simple approach can be applied regardless of the cell length as well as for any cell type (e.g. ordinary, merging, or diverging).
3.2.5 Determination of $\delta_j$

Cell length variability also requires adjustment to the original definition of $\delta_j$, used in equation (5). Recall that the congestion indicator $\delta_j$ determines the rate at which available spaces, $x^j_t$, in cell $j$, can be used by the advancing vehicles. This factor is dependent on the traffic conditions at the entrance of cell $j$, which can be estimated by comparing the number of vehicles eligible to advance from the sending cell $i$ to the flow capacity of the two interconnecting cells, $i$ and $j$. If the flow capacity is exceeded by the vehicles demand, then the traffic is in forced-flow conditions. Thus, $\delta_j$ can be estimated as follows:

$$
\delta_j = \begin{cases} 
1 & \text{if } x^j_t \leq \tau \min(Q_i, Q_j) \\
\frac{W_j}{V_j} & \text{otherwise}
\end{cases}, \quad (32)
$$

Which is mathematically equivalent to:

$$
\delta_j = \begin{cases} 
1 & \text{if } \phi^j_{i_{n-1,j}} \leq \tau \min(Q_i, Q_j) \\
\frac{W_j}{V_j} & \text{otherwise}
\end{cases}. \quad (33)
$$

The definition in equation (33) applies to ordinary and diverging receiving cells. For merging cells, vehicles may advance from multiple merging connectors and the congestion indicator has to be adjusted as follows:

$$
\tilde{\delta}_j = \begin{cases} 
1 & \text{if } \sum_{i \in \Gamma^{r(j)}} x^i_t(t) \leq \min \left[ \sum_{i \in \Gamma^{r(j)}} Q_i, Q_j \right] \tau \\
\frac{W_j}{V_j} & \text{otherwise}
\end{cases}, \quad (34)
$$
which is mathematically equivalent to:

\[
\delta_j = \begin{cases} 
1 & \text{if } \sum_{i \in \Gamma^{-1}(j)} \phi_{i,j}^0 \leq \min \left[ \sum_{i \in \Gamma^{-1}(j)} Q_i \right] \\
\frac{W_j}{V_j} & \text{otherwise}
\end{cases}
\]  (35)

3.3 Experimental Analysis

This section presents testing and validation results for the aforementioned topological modifications.

3.3.1 Effect of Variable Cell Length on Simulation Results

As outlined earlier, specific corrections had to be made to the flow advancing equations in order to allow topologies with variable cell sizes in CTM. To examine the effect of such corrections on the simulation performance a simple network composed of a few ordinary cells was constructed, as shown in Figure 10. Different topologies were used to simulate the same traffic conditions on this network.

A base case scenario was created by dividing the network into 5 cells: a source cell, a sink cell, an upstream cell (2) (two miles long), a downstream bottleneck cell (RC) (0.2 miles long), and an intermediate cell (1) (0.4 miles long). Each cell was further
divided into subcells of equal size (264 ft long) to eliminate the effect of cell size variations and provide a basis for comparisons with other scenarios. The network properties are shown in Table 2. Assuming a constant free-flow speed of 60 mph for all cells, the free-flow travel time to traverse any subcell was estimated from $264/(60*1.47) = 3$ seconds. Other network parameters were assumed based on a triangular flow-density with the following characteristics: a jam density of 200 pcpm, a flow capacity of 2200 pcphpl, and a backward moving wave speed of 13.5 mph.

Table 2: Network properties (Base Case)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>1</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>2</td>
<td>10560</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
</tbody>
</table>

Two other scenarios were created by changing the network topology and varying the cell sizes. In scenario one, the intermediate cell was split into two unequal cells (1a and 1b), as shown in Figure 11 with lengths 924 ft and 1188 ft, respectively. Cell 1a was divided into four subcells, three 264-ft long subcells followed by one 132-ft long subcell. Cell 1b was divided into five subcells, four 264-ft long subcells followed by one 132-ft long subcell (see Table 3).

![Figure 11: Detailed topology of study section (Case 1)](image)
Table 3: Network properties (Case 1)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>1a</td>
<td>924</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>1b</td>
<td>1188</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>2</td>
<td>10560</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
</tbody>
</table>

Similarly, in case 2 the intermediate cell was split into two unequal cells (1a and 1b), as shown in Figure 12 and Table 3, with lengths 1294 ft and 818 ft, respectively. Cell 1a was divided into five subcells, four 264-ft long subcells followed by one 238-ft long subcell. Cell 1b was divided into four subcells, three 264-ft long subcells followed by one 26-ft long subcell.

Table 4: Network properties (Case 2)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>1a</td>
<td>1294</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>1b</td>
<td>818</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>2</td>
<td>10560</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
</tbody>
</table>

Figure 12: Detailed topology of study section (Case 2)
To examine the effect of cell size variations on the simulation performance under extreme queuing conditions, heavy traffic congestion was set out in the simulated network by reducing the number of lanes from two to one to restrict the flow capacity into the downstream cell (RC). This geometric bottleneck caused traffic to spill back inside the intermediate cells. All three scenarios were simulated for one hour under the same demand shown in Table 5. Since the demand flow rate (3600 pcph) during the first simulation period exceeds the flow capacity of the downstream cell, RC (2200 pcph), the geometric bottleneck at the entrance of RC was active and queuing conditions propagated into upstream cells. During the second simulation period, the demand flow rate was reduced to allow the queues to dissipate before the end of the simulation period.

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source</td>
<td>08:00:00</td>
<td>20</td>
<td>3600</td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>08:20:00</td>
<td>30</td>
<td>600</td>
</tr>
</tbody>
</table>

The simulation results for all case scenarios were expressed in terms of cell occupancies. In this study, the occupancy of a cell is defined as the total number of vehicles physically present in that cell. Figure 13 shows the occupancy profile over time for the intermediate cell in the base case scenario.

Figure 13 clearly shows how the occupancy increases gradually at the beginning of simulation to a maximum of nearly 90 vehicles because of the high demand. During the second period of simulation, the occupancy drops to nearly five vehicles and then diminishes as vehicles clear out of the network. The difference between the cell
occurrences of the base case scenario and each of the other two scenarios was used to quantify the error caused by cell size variations, as shown in Figure 14.

Figure 13 Occupancy of the study section (base case)

Figure 14 Absolute error in occupancy for comparative Cases 1 and 2
It can be seen that for the congested period the maximum absolute error in occupancy did not exceed nearly 0.07 vehicles. For both scenarios, the errors increase during queue formation and dissipation and drop to zero during the steady-flow traffic conditions. However, the magnitude of the errors is very small and it can be attributed to the mathematical rounding errors. Moreover, under free-flowing conditions, the errors are zero for the second case but for the first comparative scenario, a maximum of 0.08 is observed for the whole duration of the light traffic flow. This may be explained by the forced lane-changing effect between cells 1b and RC. The lane changing impact is investigated in Chapter 5 of the dissertation.

In addition to the discrepancy in cell occupancies, the overall network travel time and average delay per vehicle were also used to estimate the effect of cell size variations. It can be seen from Figure 15 that both scenarios produce total network travel times within 0.05% absolute relative error of the nearly 70 vehicle-hours of total travel time in the base case. In addition, at an average free-flow travel time of 2 min and 36 sec, an average delay per vehicle of about 72 seconds is observed in each of the three networks, regardless of the topology used.

3.3.2 Non-Discrete Vehicle Movements

The effect of allowing continuous movement of vehicles between cells was examined using the test network shown in Figure 10. The movement precision (p), representing the number of decimal places used to advance quantities of vehicles, was varied from 0 to 5, with 0 representing the base case of moving whole vehicles only. Figure 16 shows the total network travel times from average runs with different movement precisions.
The figure shows that for $p = 0$ (discrete or integer vehicle movements) the total network travel time is 175 vehicle hours and drops to around 70 vehicle hours for all cases where $p > 0$. Since the partial advancing of vehicles is not allowed for $p = 0$, vehicles may be forced to stay longer in cells as a result of rounding down the advancing quantities. For relatively short simulation time steps, the instantaneous cell flow capacity is typically small, causing high rounding-off errors. Since the flow advancing equation in CTM is derived from the minimum of four terms, the actual number of advancing vehicles will always be rounded down to the nearest integer when $p = 0$. This may cause the flow and

Figure 15 Effect of cell size variations on network travel time and average vehicular delay

The figure shows that for $p = 0$ (discrete or integer vehicle movements) the total network travel time is 175 vehicle hours and drops to around 70 vehicle hours for all cases where $p > 0$. Since the partial advancing of vehicles is not allowed for $p = 0$, vehicles may be forced to stay longer in cells as a result of rounding down the advancing quantities. For relatively short simulation time steps, the instantaneous cell flow capacity is typically small, causing high rounding-off errors. Since the flow advancing equation in CTM is derived from the minimum of four terms, the actual number of advancing vehicles will always be rounded down to the nearest integer when $p = 0$. This may cause the flow and
space capacities to be underutilized, and therefore, lead to higher travel times. As $p$ increases to values greater than one, the variation in travel times appears insignificant, and consequently $p = 2$ provides sufficient approximation for the 3-second time step used.

Figure 16 Effect of Movement Precision on Network Travel Time

In this example, it is demonstrated that using non-discrete vehicle movements is vital to recognizing the true approximation of the network performance with respect to the traffic flow representation. By varying the level of precision in representing vehicle quantities, from zero (integer movements) to five decimals, a decreasing value for total network travel time was recorded. The large difference in travel time of more than 137% absolute relative error, when using discrete movements vs. one decimal precision to representing the progression of vehicles, demonstrates the need for non-discrete vehicle
movements. However, as the precision increases the variation in travel time is less significant, due to the decreasing cumulating errors from the decimal truncation.

3.4 Summary

This chapter introduced a modification of the topology of the original CTM that allows for an arbitrary selection of the cell’s length in a network. This modification is based on representing the cells with equal-length interconnecting subcells, with the exception that only the last subcell is allowed to have smaller lengths. This modification is very important because allows for the flexibility in using any simulation time step regardless of the variations in cell length. Specific changes in the flow advancing equation are deemed necessary to account for this topological relaxation. Two stages are identified in applying the flow advancing formulation. At the cell level, the flow advancing equation dictates how vehicles propagate between the subcells, and a special treatment of the vehicle movements between the last two subcells of a cell is developed. While at the network level, the vehicle movements between the cells are treated using a modified version of the flow advancing equation. These changes are tested with a hypothetical network and comparisons between a base case and two other cases are performed. It is shown that the differences in cell occupancy, total network travel-time and average delay between the compared scenarios are marginal, and are most likely attributed to the rounding errors. In addition, the effect of using discrete vs. non-discrete representation of vehicle flows is investigated. It was concluded that it is essential for the accuracy of simulation to represent non-discrete vehicle movements in the system, and implicitly in the equations of the model.
CHAPTER 4 LANE-CHANGING LOGIC

4.1 Introduction

It has been reported by Li et al. (1999) that the original CTM may generate pathological cases that will lead to violation of FIFO rule. The FIFO principle maintains the order in which vehicles enter and exit the cells. In CTM, the FIFO principle could be violated in diverging cells, because the model is not able to capture non-homogeneity in lane occupancy inside the cells. In this chapter, a lane-changing approach is introduced. To account for lane-changing maneuvers in CTM the traffic flow inside the cells has to be represented by lane. Specific lane-changing algorithms are developed to implement the lane-changing maneuvers. In the second part of the chapter, an experimental analysis provides more insights about the realism of the lane changing algorithms in CTM.

4.2 Accounting for Lane-Changing Behavior in CTM

The proposed mesoscopic approach models the vehicles propagation through the network by assigning individual travel lanes in each cell. This is the first step in implementing a more intuitive representation of the traffic flow in traffic networks. For example, when two unbalanced traffic streams are merging (e.g. two merging freeways, with heavy and light traffic, respectively) it is commonly observed that after the merge the occupancy across all lanes evens out, as a result of driving behavior (i.e. drivers select travel lanes with the least occupancy). Other situation that may create unbalanced distribution of flows across lanes can be observed in congested diverging sections. For
example, a congested freeway off-ramp that backs-up traffic into the mainstream leads to uneven distribution of flows upstream of the diverging junction. To model these traffic conditions it is necessary to represent lane-by-lane traffic flows in CTM. In addition, a lane-changing type of behavior has to be accounted for.

In the proposed lane-changing algorithms, as vehicles transfer from one cell to another and inside the cell from one subcell to another, they have the option to select a different travel lane in two situations. The first assumes a discretionary selection of the travel lane, and it is based on the prevailing traffic conditions for the available lanes that lead to the desired destination. The second situation considers a mandatory lane selection based on the assigned successor cell out of the diverging cell. The next sections describe the modeling of the discretionary and mandatory lane selection.

4.3 **Discretionary Lane Selection Logic**

For discretionary lane changing maneuvers the driver’s decision of the target travel lane is assumed to depend on the perceived utility of each lane in the set of target lanes, $\Gamma(r)$, available to lane $r$. The utility of a lane is an indicator of how appealing a specific lane is to a specific driver. This factor is assumed to take any value between 0 and 1, with large values indicating a higher probability to travel onto that lane. The perceived lane utility by a driver is assumed to be directly related to the level of congestion or relative occupancy of that lane $\frac{x_i(m,s)}{X_i(m,s)}$, where $x_i(m,s)$ represents the number of vehicles occupying lane $s$ in subcell $m$ in cell $i$, and $X_i(m,s)$ represents the maximum space capacity of lane $s$ of subcell $m$ in cell $i$.  


The set of target lanes is assumed to consist of two subsets: $\Gamma(r \Downarrow)$, the set of lanes requiring lane changing maneuvers from $r$, and $\Gamma(r \leftrightarrow)$, the set containing the lane that does not require lane changing maneuvers, such that $\Gamma(r) = \Gamma(r \leftrightarrow) \cup \Gamma(r \Downarrow)$. Typically, $|\Gamma(r \Downarrow)| \leq 2$ and $|\Gamma(r \leftrightarrow)| \leq 1$. A disutility function for lane $s$ can be determined by:

$$D_s = -\alpha_s \frac{x_i(m,s)}{X_i(m,s)} + \beta_s$$

(36)

**Figure 17: Illustrative target lane disutility function**

Where,

$D_s =$ the overall perceived disutility of target lane $s$
\[ \alpha_s = \begin{cases} 1 & \text{if } s \in \Gamma(r \leftrightarrow) \\ a & \text{if } s \in \Gamma(r \downarrow) \end{cases} \]

\[ \beta_s = \begin{cases} 0 & \text{if } s \in \Gamma(r \leftrightarrow) \\ b & \text{if } s \in \Gamma(r \downarrow) \end{cases} \]

\(a = \) the perceived lane-changing cost associated with movement from lane \(r\) to lane \(s\) for each unit change of lane occupancy, \(a\) is assumed to be \(\geq 1\)

\(b = \) the perceived lane-changing cost associated with movement from lane \(r\) to lane \(s\), regardless of lane occupancy, \(b\) is assumed to be \(\geq 0\)

Hence, given the disutility function of each lane in the set \(\Gamma(r)\), the probability of selecting a destination lane \(s\) can be estimated by a logit model in the form:

\[ p(s) = \frac{e^{D_s}}{\sum_{\forall k \in \Gamma(r)} e^{D_k}} \] (37)

The following is a simple algorithm used to implement the discretionary lane selection:

1. Select a vehicle from lane \(r\)

2. Determine the set of lanes \(\Gamma(r)\) available to this vehicle

3. If \(|\Gamma(r)| = 1\), vehicle must choose the only lane available in \(\Gamma(r)\), terminate;

   Else at least two lanes are available to choose from, proceed to next step.

4. For each lane \(s\) in the set \(\Gamma(r)\), calculate the lane disutility \(D_s\)

5. Calculate the probability of NOT changing lane, \(p(s \mid s \in \Gamma(r \leftrightarrow)) = \frac{e^{D_{sr}}}{T_r}\), where \(T_r = \sum_{\forall s \in \Gamma(r)} e^{D_s}\)
6. Generate a random number $R$ from 0 to 1

   If $R \leq p\{s \mid s \in \Gamma(r \leftrightarrow)\}$, this vehicle will not change lanes, i.e. the target lane is set to $s \mid s \in \Gamma(r \leftrightarrow)$, **terminate**;

   Else vehicle will change lanes from the set $\Gamma(r \rhd)$.

7. If $|\Gamma(r \rhd)| = 1$, vehicle must choose the only lane in $\Gamma(r \rhd)$, **terminate**;

   Else there are two lanes to choose from $\Gamma(r \rhd) = \{s_1, s_2\}$.

8. Adjust $T_r = T_r(1 - p\{s \mid s \in \Gamma(r \leftrightarrow)\})$

9. Calculate the probability of choosing lane $s_1$, $P\{s = s_1\} = \frac{e^{U_{S_1}}}{T_r}$

10. Generate a random number $R$ from 0 to 1

    If $R \leq P\{s = s_1\}$, this vehicle will choose lane $s_1$, **terminate**;

    Else vehicle will choose lane $s_2$, **terminate**.

4.3.1 **Effect of $\alpha$ and $\beta$ on Lane Selection Probability**

The behavior of a driver that uses discretionary lane selection logic is modeled by two parameters, $\alpha$ and $\beta$. These parameters represent the perceived costs associated with lane-changing maneuvers. Arbitrary selected values have been assigned to these parameters to evaluate their effect on the probability of selection of a specific target lane. Figure 18 shows how the probability of selection a target lane varies for different $\alpha$ values when no default cost associated with lane changing maneuvers is perceived by the driver, $\beta = 0$. It can be seen that for the same relative occupancy of the target lane, the probability decreased with increase in $\alpha$. In addition, the probability of selection shows a higher rate of change for small values in relative occupancies, than for higher occupancies.
This implies that a driver is more likely to consider lane changing maneuvers under light traffic conditions (i.e. low lane occupancy), than under congested conditions (i.e. high lane occupancy).

![Figure 18: Probability to select a target lane for $\beta = 0$](image)

The second parameter of the discretionary lane-changing algorithm, $\beta$, may also have strictly positive values if the drivers perceive a default penalty for executing a lane-changing maneuvers, regardless of the relative occupancy of the target lanes. This parameter is associated with the level of discomfort for a driver that executes a lane changing (e.g. estimating the available gap, checking the mirrors for the presence of vehicles in the adjacent lanes, etc.). Positive values of $\beta$ reduce the overall utility of the target lane, regardless of the traffic conditions. Figure 19 shows that for the same values in
relative occupancy and the same values of $\alpha$ the probability of selection is lower than the case with $\beta$ zero, shown in Figure 18. However, similar trend of probability variations is observed. This implies that a driver that perceives a default cost associated with lane-changing maneuvers, $\beta > 0$, is less likely to change lanes than a driver with $\beta$ zero. For example, if the occupancy of the target lane is zero, the probability of selection is reduced from 1 to 0.6, for $\beta$ changing from 0 to 0.5, respectively.

![Figure 19: Probability to select a target lane for $\beta = 0.5$](image)

Figure 20 shows that the higher the default cost associated with lane-selection, $\beta$ the higher the reduction in probability of selection of a target lane. It can be seen that for $\beta = 1.5$ leads to a decrease in probability of selection to about 0.22 for zero occupancy of...
the target lane. Moreover, it can be seen that for higher values of $\beta$ the variation in probability is less sensitive to the relative occupancy (i.e. the probability values range between near 0 to about 0.22).

![Probability to select a target lane for $\beta = 1.5$](image)

Figure 20: Probability to select a target lane for $\beta = 1.5$

### 4.4 Mandatory Lane Selection Logic

Mandatory lane changing maneuvers are executed by vehicles in diverging cells if the current lane does not lead to the destination cell. This maneuver is assumed to take place at the entrance of and inside diverging cells. Once a target diverging cell is selected at the entrance of a diverging cell, the vehicle will attempt to move from its current lane to the closest lane leading to the target cell. Lane changing opportunities are only available
when vehicles first advance into the diverging cell and subsequently advance from one subcell to another inside the diverging cell. Therefore, as the vehicle gets closer to the exit of the diverging cell, the utility of the lanes leading to the target cell, is assumed to increase.

Given a diverging cell with \( n \) subcells, there exist \( n \) lane-changing opportunities from the entry point to the exit point of the diverging cell. Let \( n' \) be the minimum number of lane changing maneuvers required to reach the nearest lane leading to the target cell, and \( n'' \) be the number of lane changing opportunities remaining to the end of the diverging cell. The ratio \( \lambda = n'/n'' \) can be used as an urgency measure for lane changing at any specific location. \( \lambda \) ranges from zero (least critical) to one (most critical).

The probability of making a lane changing maneuver is assumed as follows:

\[
P(s \mid s \in \Gamma(r \downarrow)) = \lambda^b
\]  (38)

Where,

\( b \) = is a measure of driver aggressiveness when executing lane-changing maneuvers in diverging cells.

An illustration of the probability of making a lane changing maneuver is depicted in Figure 21. It can be seen that for \( b = 2 \), the probability of making a lane changing maneuver is less than for \( b = 1/2 \). Therefore it can be inferred that \( b \) is an indicator of driver aggressiveness. A conservative driving behavior leads to early execution of lane-changing maneuvers (higher values of \( P(s \mid s \in \Gamma(r \downarrow)) \) for the same \( \lambda \)). A more aggressive driving behavior is characterized by late early execution of lane-changing maneuvers (lower values of \( P(s \mid s \in \Gamma(r \downarrow)) \) for the same \( \lambda \)).
A systematic implementation of the mandatory lane-changing maneuver algorithm is described next.

1. Assign a random number $R$ and use it to estimate $b$ for a particular vehicle; this step is made once for every vehicle, at initialization stage, since the characteristics of the driver will not change during the simulation.

2. Construct the set of lanes available given source lane $r$; $\Gamma(r \rightarrow)$ and $\Gamma(r \downarrow)$

3. Calculate $\lambda = \frac{\eta''}{\eta''}$

4. If $\lambda = 1$, then lane changing is critical, choose $s \in \Gamma(r \downarrow)$, terminate;

   Else calculate $P(s \mid s \in \Gamma(r \rightarrow)) = 1 - \lambda^b$
5. Assign a random number $R$

If $R \leq P(s \mid s \in \Gamma(r \rightarrow))$, then vehicle will not change lanes, $s \in \Gamma(r \rightarrow)$;

Else $s \in \Gamma(r \downarrow)$.

4.5 Experimental Analysis

A sensitivity analysis was conducted to test the two types of lane-changing (LC) algorithms. For the discretionary lane-changing algorithm, the effect of different combinations of values for parameters $\alpha$ and $\beta$ on the network performance was investigated. The mandatory lane-changing algorithm was studied by changing the driver aggressiveness factor from low to high in a congested network environment.

4.5.1 Effect of Discretionary Lane-Changing Maneuvers

An arbitrary network was used to demonstrate the effect of the discretionary lane-changing behavior on the simulation results. The network topology from Figure 22 shows a possible geometric constraint that may be caused by lane-blocking incidents, work-zone, etc., and leads to a reduction in the flow capacity of the freeway segment. Therefore, congestion can be observed upstream of these temporary bottleneck sections. In this example network, cell 2 has two lanes and is connected upstream to cell 3 with three lanes and downstream to cell 1 with three lanes. The network parameters are shown in Table 6. The free-flow and backward moving wave speeds assumed for all cells were 60 mph and 13.5 mph, respectively. An arbitrary simulation time scan of 12 seconds was used, which led to splitting cells 2 and 3 into 8 subcells, and cell 1 into 4 subcells. The traffic demand is shown in Table 7. The source cell releases 70% of the traffic onto lane 1, 20% onto lane 2, and 10% onto lane 3. This unbalanced demand in traffic networks may have various
reasons. For instance, lanes 2 and 3 may be part of a free-flowing freeway section, while the lane 1 is a congested merging on-ramp. To evaluate the discretionary lane-changing effect on traffic flow representation a base case, where lane changing was banned, was used as basis for comparison with other scenarios, where discretionary lane changing was enabled.

Figure 22: Network layout and topology used to examine discretionary LC behavior

Table 6: Network characteristics used to examine discretionary LC behavior

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4224</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>2</td>
<td>8448</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>1300</td>
</tr>
<tr>
<td>3</td>
<td>8448</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
</tbody>
</table>

Table 7: Traffic demand used to examine discretionary LC behavior

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source</td>
<td>8:00:00</td>
<td>45</td>
<td>3000</td>
</tr>
</tbody>
</table>

4.5.1.1 Base Case: No Lane-Changing Maneuvers

The results in terms of lane-by-lane occupancy distribution for the section upstream the bottleneck are plotted in Figure 23. It can be seen that under the no lane-
changing assumption (i.e. original CTM), the uneven lane utilization prevails throughout the simulation period, which is a misrepresentation of the actual behavior of traffic flow. In reality, the drivers execute lane-changing maneuvers to advance to a lower density lane, which eventually leads to nearly even distribution of traffic flows across the lanes.

Figure 23: Lane occupancy distribution in cell 3 (base case)

Cell 2 shows also unbalanced occupancy of the two lanes, as depicted in Figure 24. The occupancy ratio in the two lanes is around 70/30. This is explained by the merge of traffic from lane 3 of the cell 3 into lane 2 of cell2.
Figure 24: Lane occupancy distribution in cell 2 (base case)

The lane occupancy distribution in cell 1 is shown in Figure 25. It can be seen that because no lane-changing is allowed the occupancy of lane 3 is zero. In reality, drivers driving out of a congested bottleneck into an open-up area change lanes such that additional lanes are utilized and occupancies become more evenly distributed.

4.5.1.2 Case 1: Discretionary Lane-Changing Maneuvers ($\alpha = 1; \beta = 0$)

This case assumes a discretionary lane-changing algorithm that has no default cost associated with the lane-changing maneuver, $\beta = 0$, and a minimum penalty, $\alpha = 1$, for the utility of the target lanes. The dotted plot in Figure 18 shows the probability of lane selection used in case 1.
The distribution of lane-by-lane occupancy inside cell 3 is shown in Figure 26. When compared with the base case in Figure 23, a distinct pattern in lane occupancy can be observed. In this case, though the source cell releases vehicles with the same distribution 70/20/10, the effect of the discretionary lane-changing mechanism can be observed. The occupancies of the three lanes vary about the same values. Similar results are expected in real-life networks, where drivers are likely to move from the high-density lanes to low density lanes. Small variations are observed in the occupancy of each lane over time, but this is due to the randomness in drivers’ behavior.
Figure 27 shows the distribution of occupancy by lane inside cell 2 for case 1. The comparison with the base case provides more insights about the effect of the discretionary lane-changing algorithm. The entrance to this cell acts as a bottleneck since demand exceeds capacity. It can be seen that the occupancy of the two lanes in cell 2 is nearly the same, with the exception of small random variations towards the end of the simulation period.
The distribution of the occupancy inside cell 1 is plotted in Figure 28. The major difference between case 1 and the base case is that in case 1 the vehicles released into cell 1 from cell 2, allocate themselves into the three lanes of cell 1, following the discretionary lane-changing rule. Some variations in each lane occupancy are observed over time, but this is due to the randomness in driving behavior under light traffic conditions.

Overall, case 1 shows a more realistic representation of traffic flows than the base case. All three cells show lane that are in agreement with the expected driving behavior when the effect of lane-changing maneuvers is accounted for. These results support the finding that a lane-changing algorithm is necessary to avoid misrepresentation of traffic flows inside the cells.
4.5.1.3 Case 2: Discretionary Lane-Changing Maneuvers \( (\alpha = 5; \beta = 1.5) \)

The effect of the lane changing parameters was investigated using a second comparative scenario that assumes different values for \( \alpha \) and \( \beta \). The settings of the second case, represented by the dotted plot in Figure 20, assume a higher cost \( (\alpha = 5) \) for the utility per unit change in relative occupancy of the target cell, and a higher default cost \( (\beta = 1.5) \) perceived by the driver for executing lane changing maneuvers.

The distribution of vehicle flows by lane inside cell 3 is shown in Figure 29. It can be seen that similar to case 1 the occupancies of all three lanes are nearly equal. However, lane 2 shows a relatively higher occupancy than it did in case 1. This can be
explained by the tardiness in lane-changing maneuvers. A higher value of $\beta$, 1.5 in this case versus 0.5 in case 1, implies smaller probabilities for lane selection, which lead to delayed lane-changing maneuvers.

![Lane occupancy distribution in cell 3 (case 2)](image)

Figure 29: Lane occupancy distribution in cell 3 (case 2)

Figure 30 shows the flow distribution inside cell 2. It can be seen that the occupancies of the two lanes are identical for most of the simulation duration. Similar to cell 2 in case 1, there is a small difference in the occupancy of the two lanes after simulation time 8:45. This can be explained by the randomness in allocating travel lanes for the last vehicles released in the network, which leads to this small difference in lane occupancy when no more vehicles are released into the network.
The occupancy distribution of the downstream cell 1 is shown in Figure 31. It can be seen that lane 3, the rightmost, maintains a lower density, when compared to lanes 1 and 2. This can be explained by the relatively high cost for lane-changing maneuvers under light traffic conditions. The higher values for $\beta$ lead to a significant reduction in the number of lane-changing maneuvers, when compared with case 1. Consequently, lane 3 is less preferred by travelers from lanes 1 and 2 than in case 1, which leads to an unbalanced distribution of vehicles across the three lanes in cell 1.
4.5.2 Mandatory Lane-Changing Logic

In this study, modeling the flow entering diverging cells is modified from the original model in order to account for the effects of lane-changing maneuvers. To demonstrate the improvements of this approach, a base case was constructed in which the diverging cells are modeled similar to the original CTM. A sketch of the testing network is shown in Figure 32. In this network diverging cell 4 was selected to include eight subcells of equal length (1056 ft), as detailed in Figure 33 and Table 8.

The incoming traffic upstream the diverging cell flows on three lanes. The three lanes entering the diverging cell 4 split in two directions: lane 1 and 2 lead to cell 8 and...
lane 3 leads to cell 5 (e.g. a two-lane freeway and an exit lane). In this network, the vehicles are released from the source cell under moderate demand, shown in Table 9, with evenly split flows among the three lanes. A predetermined split ratio is applied to the vehicles entering the diverging cell, two thirds of the vehicles continue on the main freeway, towards cell 8, and one third of the vehicle exit towards cell 5. In addition, to create congested conditions cell 6 has a flow capacity of 900 pcp/hpl, reduced from the freeway capacity of 2200 pcp/hpl. This reduction in flow capacity can be assumed the effect of a downstream traffic signal in real-life networks.

Figure 32: Network layout and topology

Figure 33: Detailed structure of the diverging cell 4
Table 8: Network parameters

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>3</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>4</td>
<td>8448</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>5</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>6</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>900</td>
</tr>
<tr>
<td>8</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>9</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
</tbody>
</table>

Table 9: Traffic demand

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source</td>
<td>8:00:00</td>
<td>45</td>
<td>3600</td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>8:45:00</td>
<td>30</td>
<td>600</td>
</tr>
</tbody>
</table>

During the first 45 minutes of the 2-hour simulation period, vehicles are released from the source cell at a flow rate of 3600 pcph. Consequently, the 1200 pcph demand for the off-ramp link exceeds the 900 pcph flow capacity of the downstream cell 6. Therefore, vehicles queue up at the entrance of cell 6 and about 15 minutes after the simulation begins (at simulation time 8:14:36) they spill back into the diverging cell. It can be observed that the occupancy off all three lanes in the diverging cell increase simultaneously during the
congestion build up. This implies that congested traffic conditions from the downstream cell 5 lead to queues formation on all three lanes of the diverging cell 4. This effect is rather unrealistic, because it suggests that when vehicles reach the end of the diverging cell they will jump across lanes to get to their target cell. However, a more realistic representation of the traffic flow for diverging cells is more complex, and therefore, difficult to capture accurately. In this study, the diverging cells are treated based on the implementation of a mandatory lane changing logic in order to represent more realistically various features of the traffic streams under different traffic conditions.

![Figure 34: Lane occupancy in the diverging cell 4 (without lane-changing)](image)

The proposed approach allows the drivers to choose their travel path as they travel through a diverging cell based on a selected destination assigned at the entrance of the
diverging cell. This is called mandatory lane-changing behavior, because the vehicles have to change lanes to reach certain lane that leads to their destination. Under this logic vehicles have the opportunity to change lanes as they enter the diverging cell, and subsequently at every subcell inside.

For comparison, the same network shown in Figure 32 was simulated and the proposed mandatory lane-changing algorithm was enabled. The average value of parameter $b$ was set to 1, which characterizes a moderate driving behavior. Figure 35 shows the lane-by-lane distribution of occupancy inside the diverging cell.

![Figure 35: Lane occupancy in the diverging cell 4 (with lane-changing)](image)

Using a predetermined turn ratio vehicles are randomly assigned to their destination as they enter the diverging cell. For example, at random, vehicles targeting cell
5 may enter on lane 1, and will have to change lanes in order to reach lane 3 that leads to their destination. Similarly, vehicles selecting cell 8 as their next cell may enter in lane 3 and will need to change lanes to lane 1 or 2 in order to reach their destination. It can be seen that for the first 15 minutes all three lanes in the diverging cell have nearly the same occupancy, with small variations due to lane-changing maneuvers. The nearly evenly distributed flow inside the diverging cell across the three lanes is in agreement with the results shown in Figure 34, which plots the occupancy of the diverging cell under no lane-changing assumption. In addition, except for the random variation this behavior is close to the expected real-life traffic patterns under free-flow conditions.

Under congested conditions, in traffic networks interactions that are more complex take place between the vehicles in the diverging sections. It can be seen from Figure 35 that about 15 minutes after the simulation begins, the occupancy of the exit lane 3 starts to increase, while the occupancy of through lanes 1 and 2 is maintained nearly the same. This effect is intuitive and is in agreement with what one would expect to happen on a congested off-ramp and a free-flowing freeway. It can also be seen that lane 2, adjacent to the congested lane 3 shows a somewhat higher occupancy than lane 1. This can be explained by the late lane-changing maneuvers. For example, some more aggressive drivers that are destined for the off-ramp and traveling in the less congested lane 2 may defer the lane-changing maneuvers until they encounter their last opportunity to change lanes. Hence, those drivers will hold traffic behind them in the same travel lane, and this inherently leads to increase in lane occupancy. In the second part of the simulation (8:45-9:15), after the excess demand for the ramp reduces to 200 pcp/h the traffic conditions
become free flowing again. It can be seen that after simulation time 09:03:12, all three lanes are back to free-flow conditions and exhibit nearly the same occupancy.

The detailed occupancy by subcell of each lane inside the diverging cell 4 was investigated. As detailed in Figure 33 cell 4 contains eight subcells. Figure 36 represents the occupancy profile of lane 1 assuming that mandatory lane changing logic is disabled. It can be seen that all subcells, from SC1 to SC8, have equal occupancies. At every simulation update each subcell is occupied by about four vehicles, for the first nearly 15 minutes, during the free-flowing conditions. Nevertheless, after the congestion on the off-ramp reaches the diverging cell, the occupancy of the last subcell, SC8, starts to increase, while traffic in the remaining subcells is still free flowing. It can be seen that the congestion propagates upstream and reaches subcells SC7, SC6, SC5 and SC4 at simulation times 8:31:12, 8:36:12, 8:41:12, and 8:46:12 respectively. In this particular example, congestion takes 5 minutes to propagate within each subcell.

By comparison Figure 37 shows the detailed occupancy in lane 1 when mandatory lane changing logic is enabled. During free-flow conditions all subcells have randomly variable occupancy, with average values of about 4 vehicles. As opposed to the case with no mandatory lane-changing maneuvers, an increase in the occupancy of last subcell, SC8, does not occurs immediately after the congestion spills back into the diverging cell, but at simulation time 8:30. Moreover, in this case only subcells SC8, SC7 and SC6 show congested conditions. This can be explained by the fact that vehicles queuing up on the off-ramp do not have a direct impact on the traffic streams of the adjacent lanes. These differences in occupancy across lanes and between subcells show
that non-homogeneous traffic conditions inside long diverging cells can be captured if a mandatory lane-changing logic is used.

Figure 36: Occupancy profile by subcells for lane 1 (no lane-changing)

Figure 38 plots the occupancy of lane 2 for each subcell when mandatory lane-changing maneuvers are not permitted. It can be seen that congestion reaches each subcell at the same simulation times as the subcells of lane 1. Some variations in the actual lane occupancy for individual subcells are identified, but they can be explained by the randomness in driver behavior.
Figure 37: Occupancy profile by subcells for lane 1 (with lane-changing)

Figure 38: Occupancy profile by subcells for lane 2 (no lane-changing)
The detailed occupancy profile of the same lane, 2, with mandatory lane-changing logic enabled is represented in Figure 39. It can be seen that congestion occurs in the last subcell, SC8 at simulation time 8:30. Moreover, only two subcells appear to have congested conditions, SC8 and SC7. This means that the spill-back of congested traffic conditions from the off-ramp into the diverging cell at about 8:15 does not affect directly lane 2. The effect of congestion is seen indirectly some time later and is due to some late lane-changing drivers that may hold traffic behind them in lane 2. These results demonstrate that non-homogeneous traffic conditions inside long diverging cells can be captured if a mandatory lane-changing logic is used.

![Figure 39: Occupancy profile by subcells for lane 2 (with lane-changing)](image-url)
Figure 40 shows the occupancy profile of lane 3 assuming no mandatory lane-changing logic is used. It can be seen for the first 15 minutes all subcells are in free-flow conditions. Then congestion propagates from the off-ramp and reaches subcells SC8 – SC4 at the same simulation times as lanes 1 and 2. These results demonstrate that when no lane-changing logic is used diverging cells cannot capture non-homogeneous traffic flows across lanes. All three lanes exhibit simultaneous congested conditions in all subcells.

![Figure 40: Occupancy profile by subcells for lane 3 (no lane-changing)](image)

The detailed occupancy profile of lane 3 with mandatory lane-changing logic enabled is represented in Figure 41. This occupancy profile shows how the effect of the congestion generated by the reduced capacity on the off-ramp impacts one by one the subcells in lane 3. For example, nearly 15 minutes after the simulation began the last subcell, SC8, shows an increase in occupancy. About five minutes later, subcell SC7 is
affected by the congestion and its occupancy starts increasing as well. After another 10 minutes, the occupancy of subcell SC6 starts increasing. Similar congested conditions are observed about 8 minutes later in subcell SC5. By the time the congestion propagates into the next subcell, SC4, the high demand from the first simulation period terminates and this subcell does not show very high occupancy. The last four subcells, SC5, SC6, SC7, and SC8, reach a maximum occupancy of 40 vehicles, which is nearly equal to the jam density. The high occupancy of lane 3 can be explained by the mandatory lane-changing maneuvers and the spill back of vehicle queues into the diverging cell 4 from the downstream cell 5, more precisely into lane 3 that is geometrically aligned with the congested off-ramp. Recall that vehicles advancing inside the diverging cell between subcells have the opportunity of changing lanes based on their pre-assigned destination at the entrance. Depending on the individual drivers’ behavior, lane-changing maneuvers will be executed sooner by the more conservative drivers, or later by the more aggressive ones.

4.6 Summary

This chapter provides a new approach to represent the traffic flows in CTM. Mainly, a separation of flows by lanes is implemented in CTM. This approach is justified because it helps capturing possible non-homogeneous traffic flows inside the cells. Different reasons may create highly non-homogeneous flows across lanes in traffic networks. Consequently, lane-changing algorithms are deemed necessary to account for a better representation of traffic flow. Two lane-changing algorithms are developed in tested in this chapter.
First a discretionary lane-changing algorithm was implemented and tested. A 3-lane simple freeway network is simulated assuming that vehicles are released from the source cell in an unbalanced fashion, 70%, 20%, and 10% are released into lane 1, lane 2, and lane 3, respectively. The algorithm includes two parameters, $\alpha$ and $\beta$, that account for different levels of costs perceived by the drivers when engaging in a lane-changing maneuver. By varying the two parameters, $\alpha$ and $\beta$, from low to high, the effect in the distribution of vehicles across the lanes inside each cell was investigated. It was shown that the discretionary lane-changing algorithm plays a very important role in removing the misrepresented separation of traffic streams of the vehicles observed when no lane-changing behavior is modeled.
Second, the mandatory lane-changing logic was tested with a 3-lane freeway network that splits into a 2-lane freeway and an off-ramp. It was shown that the use of a lane-changing logic controls the vehicles separation at the diverging cell in a more realistic fashion, as opposed to the original CTM that cannot correctly simulate the non-homogeneity of flow across lanes inside the diverging cells. In conclusion, it was proven in this chapter that if a lane-base representation of the traffic flows inside the cells is used in CTM, also some lane-changing logic has to be in place to represent more realistically the traffic flow evolution under various traffic conditions.
CHAPTER 5  MERGING LOGIC

5.1  Introduction

Modeling traffic streams in a mesoscopic environment, under the homogeneity assumption of the original CTM, necessitates the implementation of special algorithms to treat traffic streams at merging and diverging junctions. In this chapter a merging logic is developed and tested. The proposed merging algorithm also accounts for lane-changing maneuvers at the merging junction. The next section of the chapter presents the mechanism of the merging logic that advances vehicles from the merging streams in the FIFO order. The following section demonstrates the usage of the developed algorithm by testing its effect on two sample networks.

5.2  Improvements to Merging Logic

The original CTM has limited representation of merging traffic flows. For operational analysis a more realistic modeling of the merging behavior is deemed necessary. Recall that a merging junction is represented in CTM through corresponding merging cells as shown in Figure 42.

![Figure 42 An example of a merging cell i](image)

Figure 42  An example of a merging cell i
According to the formulation developed by Daganzo the merging streams are treated with a simple linear program as follows.

\[
\text{max} \sum_{k \in \Gamma^{-1}(i)} y_{ki} \quad (39)
\]

Subject to:

\[
y_{ki} \leq \min \left\{ \rho_k x_k, \rho_i q_i, q_i, \delta_i \left( X_i - x_i \right) \right\}, \quad \forall k \in \Gamma^{-1}(i) \quad (40)
\]

Where,

\( i = \) a merging cell in the set of merging cells \( (E_M) \),

\( \Gamma^{-1}(i) = \) the set of predecessor cells of cell \( i \),

\( y_{ki} = \) the flow on the merging connector from cell \( k \) to cell \( i \),

\( x_k, x_i = \) the vehicles occupying cells \( k \) and \( i \), respectively,

\( Q_k, Q_i = \) the flow capacity of cells \( k \) and \( i \), respectively,

\( X_i = \) the maximum space capacity of the merging cell \( i \).

The equations above account for specific priorities on merging connectors, \( \rho_k \), and thus, establish quantitative rules for assigning right of way to merging vehicles from roads with different hierarchies, as follows:

\[
\rho_k \leq 1, \quad \forall k \in \Gamma^{-1}(i) \quad \sum_{k \in \Gamma^{-1}(i)} \rho_k = 1 \quad (41)
\]

The proposed merging logic is illustrated in the flowchart shown in Figure 43.

The merging algorithm is applied to each merging cell to determine sequentially how vehicles merge during every simulation update interval \( \tau \). First, the algorithm estimates
the maximum number of vehicles to advance from all predecessor cells of the merging cell $i$, $S_{\text{max}} = \sum_{k \in \Gamma^{-1}(i)} \min\{x^i_k, rQ_k\}$, as well as the maximum that can be received by the merging cell $i$, $R_{\text{max}} = \min\{rQ_i, \delta x_i\}$, during each simulation update interval $\tau$.

A systematic method is used to process vehicles from the merging streams. The vehicles eligible to advance into the merging cell are processed based on the FIFO principle. The first step in this process is to select the next cell, $k$, sending vehicles into the merging cell. Next, if the cell is open, then the next vehicle quantity, $v$, is processed into the merging cell. The vehicle quantity, $v$, may represent a whole vehicle or a fraction of a vehicle. If the whole is processed completely into cell $i$, then the next is to select another sending cell, $k$. Otherwise, the lane occupied by $v$ in cell $k$ is declared blocked. If all lanes is sending cell $k$ are blocked then the sending cell itself becomes blocked and is removed from the set of predecessors of the merging cell $i$. Also, a sending cell is declared blocked if there are no more vehicles eligible to advance out of it. When the set of predecessors of cell $i$, $\Gamma^{-1}(i)$, becomes empty the algorithm terminates.

Block A of the flowchart in Figure 43 represents the selection method for the next sending cell to be processed in the merging algorithm. The original CTM uses a random selection based on the merging priorities, $\rho_k$. However, it is believed that to obtain a better representation of the merging flows one should account for dynamic merging ratios derived from the volumes or densities of the merging streams, similar to real-life networks.
Find the maximum to be sent by Cell $i$

$$y_i := \min \{x'_i, \tau Q_i\} \forall k \in \Gamma^{-1}(i)$$

Find the maximum to be sent to Cell $i$

$$S_{max} := \sum_{k \in \Gamma^{-1}(i)} y_{ki}$$

Select next cell, $k$ to process

$$k \in \Gamma^{-1}(i)$$

Cell $k$ Open?

Select vehicle quantity, $\nu$ (FIFO ordered)

Is $\nu$ able to advance?

Yes

Advance $\nu$ in Cell $i$

$\nu$ advances completely?

Yes

Block source lane in cell $k$

No

$$\Gamma^{-1}(i) = \Gamma^{-1}(i) \setminus k$$

$\Gamma^{-1}(i) \equiv \emptyset$?

Yes

End

No

Figure 43: Merging Algorithm
Consider the merging junction illustrated in Figure 44, which shows a one-lane on-ramp merging into a 2-lane freeway. Denote by $N_k$ the total number of lanes each of the predecessor cells $k$, and by $N^s_k$, the number of shared lanes from cell $k$. This junction has no additional or auxiliary lane and it can be seen that lane 2 in the merging cell is shared by vehicle sent from both predecessor cells, $k_1$ and $k_2$, respectively. In addition, $S_k$ represents the remaining flow to be processed from cell $k$ during each simulation update, and $w_k$ is the weight factor for merging priority of cell $k$.

![Figure 44: A freeway merging junction](image)

The logic represented by block A, is executed using the following pseudo-code.

1. For each $k \in \Gamma^{-1}(i)$ compute the merging factor, $f_k$ using:

$$f_k = \sum_{k \in \Gamma^{-1}(i)} S_k \left(2 - \frac{N^s_k}{N_k}\right)$$

2. For each $k \in \Gamma^{-1}(i)$ compute the merging ratios, $r_k$:

$$r_k = \frac{f_k}{\sum_{k \in \Gamma^{-1}(i)} f_k}$$

3. Set $P_0 = 0$

For each $k \in \Gamma^{-1}(i)$ the cumulative merging probabilities, $P_k$:
4. Select cell $k$, such that:

$$P_{j-1} \leq \varepsilon < P_j \land j = \arg\max_{k \in \Gamma^{-1}(i)} j$$

Where,

$$\varepsilon \text{ = an independent random number from a [0,1] uniform distribution}$$

This algorithm is used when the merging cell operates under free flow conditions. If congestion from the merging cell propagates towards the one of the sending cells, k, then a change is necessary in the merging procedure to account for a more realistic representation of the traffic flows in the congested junction. Therefore, the last two steps of the above procedure were modified to account for this behavior as follows:

Set $P_0 = 0; \; P_{\min} = 1$;

For each $k \in \Gamma^{-1}(i)$ the cumulative merging probabilities, $P_k$:

$$P_j = P_{j-1} + r_j, \; \forall j = 1..\max_{k \in \Gamma^{-1}(i)} j$$

if $r_j > 0 \text{ then } P_{\min} = \min(P_j, P_i), \; \forall j = 1..\max_{k \in \Gamma^{-1}(i)} j$

Set $\varepsilon_k = \begin{cases} 
\varepsilon_k + P_{\min}, & \text{if } \varepsilon_k + P_{\min} < 1 \\
\varepsilon_k + P_{\min} - 1, & \text{otherwise}
\end{cases}$

Select cell $k$, such that:

$$P_{j-1} \leq \varepsilon_k < P_j \land j = \arg\max_{k \in \Gamma^{-1}(i)} j$$

Where,

$$\varepsilon_k \text{ = a random number from a [0,1] uniform distribution associated with cell k}$$
Note that for a freeway the merging priority factor, $w_k$, can be assumed 1 for all merging streams. However, in a ramp meter configuration or for streets with signalized or unsignalized intersections, $w_k$ values should be selected to reflect actual merging priorities given by the traffic control devices.

5.3 Experimental Analysis

The effectiveness of the merging algorithm to model the representation of traffic flow is tested with a hypothetical network. This section demonstrates that employing the developed merging logic will lead to a representation of traffic flow in agreement with real-life networks. First, a detail analysis of a new merging algorithm is presented. Second, a freeway network is simulated and the effect of the merging algorithm in the simulation results is investigated.

5.3.1 Effect of the Merging Logic on Merging Ratios

Two merging scenarios were tested in this study. One merging scenario is drawn in Figure 45 and it shows a one-lane on-ramp that merges into a two-lane freeway, which continues on two lanes after the merge (i.e. no auxiliary or additional lane is considered in this case). A second merging scenario, shown in Figure 46, considers a one-lane on-ramp that merges into a two-lane freeway, and continues on a separate auxiliary or additional lane. The progression of the traffic flow during one simulation update, in each of the two merging cases is analyzed using the new merging algorithm.
The traffic demand at the beginning of one simulation update is shown in Table 10. It can be seen that the 20-vehicle demand from cell 1 competes with the 15-vehicle demand from cell 2. However, the total merging demand, 35 vehicles is less than the flow capacity (50 vehicles) of the merging cell M. In addition, this total demand does not exceed the available space capacity (60 vehicles) in the merging cell. Therefore, the merging junction operates under free-flow conditions.

Under this demand, the merging algorithm was employed for both merging cases. The results plotted in Figure 47 show the vehicles progression as they merge into cell M for the no auxiliary lane scenario. It can be seen that even though the two streams are
relatively unbalanced (25 vs. 15 vehicles) the vehicles advance into cell M in an alternating fashion, such that both traffic streams are processed simultaneously within the update.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell 1</td>
<td>25</td>
<td>n/a</td>
</tr>
<tr>
<td>Cell 2</td>
<td>15</td>
<td>n/a</td>
</tr>
<tr>
<td>Cell M</td>
<td>n/a</td>
<td>60</td>
</tr>
</tbody>
</table>

The merging flow from cell 2 is completely processed after 34 steps, while the flow from cell 1 is advanced in 35 steps. This is explained by the merging ratios, \( f_k \) of the streams from the two cells. The two merging ratios, \( f_1 \) and \( f_2 \), are proportional with the demand from each cell, \( S_1 \) and \( S_2 \), weighted by 1.5 and 0.5, respectively. This merging behavior resembles the traffic behavior in similar real-life merging configurations, where under light traffic conditions limited interaction between the vehicles is expected. Therefore, vehicles simply advance into the merging cell at random, based on their arrival times.

The effect of the merging algorithm on the traffic streams for the one auxiliary lane scenario was investigated using the same traffic demand in Table 10. The results plotted in Figure 48 show the vehicles progression as they merge into cell M. It can be seen that the demand from cell 2 is processed after 27 steps, while the vehicles from cell 1 are completely advanced into the merging cell after 35 steps. This is explained by the fact that due to the geometric configuration that does not restrict the flow capacity of the
sending cells. In other words, the vehicles are processed in the order of their arrival and no interaction affects the merging process.

![Graph depicting flows on merging connectors](image)

**Figure 47** Merging streams under free-flow conditions (no auxiliary lane case)

Another set of initial conditions at the merging junction presented in Table 11. In this case, a total demand of 65 vehicles (40 vehicles from cell 1 and 25 vehicles from cell 2) is attempts to advance into merging cell within one update interval. However, even though the merging cell capacity (75 vehicles) does not limit this demand, the available space capacity in the merging cell is only 50 vehicles. Hence, the merging junction operates under congested conditions.
Figure 48  Merging streams under free-flow conditions (auxiliary lane case)

Table 11: Merging traffic demand (congestion)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell 1</td>
<td>40</td>
<td>n/a</td>
<td>50</td>
</tr>
<tr>
<td>Cell 2</td>
<td>25</td>
<td>n/a</td>
<td>25</td>
</tr>
<tr>
<td>Cell M</td>
<td>n/a</td>
<td>60</td>
<td>75</td>
</tr>
</tbody>
</table>

The results plotted in Figure 49 and Figure 50 show the flow advancing into the merging cell for the no auxiliary lane case. It can be seen that under congested conditions at the entrance of the merging cell M, both streams advance fewer vehicles than the existing demand. Similar to the free-flow conditions, the merging ratios, \( f_k \), are proportional with the demand, \( S_k \), weighted by the same factors derived from the
geometric configuration. Therefore, the ratio of the vehicles that were able to advance into the merging cell is 34:16, higher than the demand ratio from the two cells, 40:25, respectively. On the other hand, the second merging case shows a merging ratio in agreement with the demand ratio. The merging algorithm advances 31 vehicles from cell 1, and 19 vehicles from cell 2, which is equal to a final merging ratio of 1.63 nearly the same as 40/25=1.6. These results seem to conform with the what one would expect to observe in a real-life merging junction. The first case indicates that under congested conditions, the main stream of a merging junction dominates the secondary stream if the total flow capacity of the upstream links is higher than the flow capacity of the merging junction. On the other hand, congested traffic flows at merging junctions with auxiliary lanes are likely to reflect the same merging ratio as the demand ratio, because the total flow capacity of the upstream links is maintained at the merging junction level.

![Figure 49 Merging streams under congested conditions (no auxiliary lane case)](image_url)

Figure 49 Merging streams under congested conditions (no auxiliary lane case)
5.3.2 Effect of the Merging Algorithm on Simulation Results

To demonstrate the employment of the merging algorithm a small network that includes a two-lane main freeway section merging with a one-lane on-ramp was analyzed. Two practical geometric configurations were investigated in this sample network. The first scenario assumes a merging junction with one auxiliary lane, as shown in Figure 51. This configuration allows for the vehicle flows from cell 3 and 5 to continue on separate lanes after the merge. The second scenario assumes no auxiliary lane in the merging junction, as shown in Figure 52. Under this assumption vehicles sent from cell 5 into cell 6 will share lane 2 with the vehicles sent from cell 3. In both scenarios, cell 7 has 2 lanes and its connecting successor cell, 8, restricts the traffic to one lane.

Figure 50 Merging streams under congested conditions (auxiliary lane case)
Other assumed network parameters are a free-flow and wave backward moving speeds of 60 and 13.5 mph, respectively, a flow capacity of 2200 pcphpl and a jam density of 200 pcpm for all cells (see Table 12 and Table 13). In addition, the network update time was derived from the free-flow travel time of the shortest cell and was set to $1056/60 \times 1.47 = 12$ seconds.

Table 12: Network parameters – (auxiliary lane case)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>3</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>4</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>5</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>6</td>
<td>4224</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>7</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>8</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
</tbody>
</table>
Table 13: Network parameters – (no auxiliary lane case)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>3</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>4</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>5</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>6</td>
<td>4224</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>8</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>9</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
</tbody>
</table>

The simulation duration is two hours, between 8:00 and 10:00, and the traffic demand for the two scenarios is presented in Table 14. It can be seen that vehicles are released in two periods. During the first period, between 8:00 and 8:45, both source cells 1 and 2, release vehicles in two 15-minutes intervals. Source 1 releases vehicles at a rate of 3800 pcph, while source 2 is set to release vehicles at a flow rate of 400 pcph. Between 8:15 and 8:30 a demand of 1200 pcph is sent from each source cell. Between 8:30 and 8:45, no vehicles are released from any of the two sources, in order to allow the existing vehicles to clear-up through the merging junction.

Table 14: Traffic demand

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source 1</td>
<td>8:00:00</td>
<td>15</td>
<td>3800</td>
</tr>
<tr>
<td>1</td>
<td>Source 2</td>
<td>8:00:00</td>
<td>15</td>
<td>400</td>
</tr>
<tr>
<td>2</td>
<td>Source 1</td>
<td>8:15:00</td>
<td>15</td>
<td>1200</td>
</tr>
<tr>
<td>2</td>
<td>Source 2</td>
<td>8:15:00</td>
<td>15</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>Source 1</td>
<td>8:45:00</td>
<td>15</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>Source 2</td>
<td>8:45:00</td>
<td>15</td>
<td>1600</td>
</tr>
<tr>
<td>4</td>
<td>Source 1</td>
<td>9:00:00</td>
<td>15</td>
<td>400</td>
</tr>
<tr>
<td>4</td>
<td>Source 2</td>
<td>9:00:00</td>
<td>15</td>
<td>400</td>
</tr>
<tr>
<td>5</td>
<td>Source 1</td>
<td>9:15:00</td>
<td>15</td>
<td>2000</td>
</tr>
<tr>
<td>5</td>
<td>Source 2</td>
<td>9:15:00</td>
<td>15</td>
<td>200</td>
</tr>
</tbody>
</table>
In the second stage, a reverse demand pattern is used. Between 8:45 and 9:00, source 1 releases vehicles on the main stream at a very low rate, 200pcph, while source 2 sends vehicles in the network at a flow rate of 1600 pcph. The second period, between 9:00 and 9:15, is characterized by low demand on both streams (400 pcph), and the third period, 9:15 to 9:30, has moderate demand on the main stream (1800 pcph) and light traffic on minor stream (200 pcph). Note that for both geometric configurations congestion in the merging junction is generated during the first 15-minutes of the simulation, due to excess demand versus the flow capacity of the downstream cell 8. Hence, the queue that builds-up at the entrance of this bottleneck propagates through the merging junction and spills back onto cells 3 and 5. The remaining demand periods generate vehicles at flow rates less the capacity of one lane (2200 pcph), and no other congested conditions are created.

The plots in Figure 53 and Figure 54 show the variations in the merging flows during the two-hour simulation period for both scenarios under the assume traffic demand. It can be seen that there is a similar pattern in the variation of the two merging flows for each the simulation interval. However, a few differences are noticed and explained next. For example, the non-auxiliary lane case shows congested conditions at the entrance of the merging cell at simulation time 8:08. This leads to increase in the flow from cell 3 to 6 that has higher demand, while the flow from cell 5 to 6 decreases. On the other hand, the one-auxiliary lane case exhibits congested conditions at the merging section later, simulation time 8:12:24. This dissimilarity between the two cases can be explained by the difference in space capacity of the merging cell. The 2-lane merging cell in the first scenario leads to a faster propagation of the queues in the merging cell when compared with the 3-lane merging cell of the second scenario. Nevertheless, it can be seen that for both cases the
congestion in the merging cell, leads to the same change in the merging ratios, from 90/10 to nearly 98/2. This is explained by the effect of the dynamic merging ratios that are proportional with the demands from the merging streams.

In addition, a difference between the two cases is observed with respect to the random variations. It can be seen that the auxiliary-lane scenario has very few local variations in the merging ratios under both congested and free-flow traffic conditions, which are explained by randomness of the lane-changing maneuvers. Conversely, the no-auxiliary lane case has some local variations in the merging ratios, with a higher magnitude during the free-flow periods, explained by random variations in driving behavior.

Figure 53: Dynamic distribution of the merging flow rates (no auxiliary lane)
During the second demand stage, both merging scenarios show identical merging ratios. It can be seen that the demand ratios follow the ratios of the flows running onto the two merging streams. However, high local variations are observed for the non-auxiliary lane case. The cause of these variations is the randomness in lane changing maneuvers as the vehicles merge into cell 6. This effect is not observed for the auxiliary lane case because no lane-changing maneuvers are necessary.

5.4 Summary

In this chapter, a new merging logic was developed and tested. It was shown that this merging logic is able to model dynamic merging ratios derived from the traffic demand. In addition, the algorithm contains a special treatment of the junction that has
auxiliary lanes. The effect of the algorithm on the merging ratios was tested with two merging scenarios. One scenario assumes an auxiliary lane exists at the merging junction, while the second scenario assumes no auxiliary lane. It was shown that under free flow conditions merging ratios are equal to the traffic streams demand ratios, for both scenarios. Moreover, under congested conditions, the non-auxiliary lane junction showed a merging ratio that favors the traffic stream with higher demand, as opposed to the auxiliary lane junction that shows a merging ratio equal to the traffic streams demand ratio.

The effect of the merging algorithm on the simulation results was tested with two similar freeway networks. One network contains a merging junction with an auxiliary lane and the second network assumes a merging junction without an auxiliary lane. Both networks were tested under free-flow and congested traffic conditions assuming variable demand ratios. The variation in the merging flows at the merging junctions of both networks was investigated. It was shown that for both networks the merging flows are sensitive to the traffic demand. Different variations in the magnitude of the merging flows between the two simulations were observed, but they are mostly due to randomness in driving behavior.

In conclusion, when used for congested merging junctions with reduction in capacity, this algorithm allows the higher demand flows to prevail over the lower demand flows. For all other traffic and geometric conditions the merging ratio is equal to the traffic demand ratio. These results are more intuitive rather than assuming fixed merging ratios. Therefore, this merging logic improves CTM by providing a more realistic representation of the merging flows.
CHAPTER 6  DIVERGING LOGIC

6.1   Introduction

The development of a systematic procedure that models vehicles interaction at diverging junctions is necessary to improve the realism of traffic flow representation in CTM. In this chapter a diverging algorithm is presented. In addition, an illustrative example shows the utilization of the diverging algorithm with a sample freeway network. The results of the tested network demonstrate that real-life behavior of the traffic streams at diverging junctions can be approximated with the proposed diverging algorithm.

6.2   Improvements to Diverging Algorithm

Diverging cells are particularly important because vehicles approaching a diverging junction must decide which path to follow in order to reach their final destination. The selection of a target cell may be determined by any appropriate route choice model (e.g. shortest path) or using simple diverging ratios that are assumed to be known a priori. In traffic networks non-homogeneous flows may be observed across lanes in a diverging area. For example, on urban freeways during the morning and afternoon peaks, the traffic on heavily congested off-ramps spills back into the main stream. This behavior may create local variations in the traffic flow propagation, such that vehicles are temporarily prevented from advancing through the network.
Recall that for each diverging cell $i$, as shown in Figure 55, in the set of diverging cells ($E_{d}$), the outflows on diverging connectors are determined by the simple linear program.

\[
\text{max } \sum_{j \in \Gamma(i)} y_{ij}
\]

Subject to:

\[
y_{ij} \leq \min\{r_jx_i, r_jQ_i, Q_j, \delta_j(X_j - x_j)\}, \forall j \in \Gamma(i)
\]

\[
\sum_{j \in \Gamma(i)} r_j = 1
\]

Where,

$\Gamma(i)$ = the set of successors cells of cell $i$,

$y_{ij}$ = the flow on the merging connector from cell $i$ to cell $j$,

$x_i, x_j$ = the vehicles occupying cells $i$ and $j$, respectively,

$Q_i, Q_j$ = the flow capacity of cells $i$ and $j$, respectively,

$X_j$ = the maximum space capacity of cell $j$.

$r_j$ = the turn ratio from cell $i$ to cell $j$
The diverging logic, shown in Figure 56, was improved to account for possible blockage on one or more of successor cells of the diverging cell. Also, the algorithm accounts for restraint of flow in the diverging cell lanes that lead to the blocked cell(s).

\[
y_i' := \min \{ x_{i+1}^*, Q_i, Q_j, \delta_i x_j' \}; \quad S_{\text{max}} := \sum_{j \in \Gamma(i)} x_{i+1}^*
\]

\[
Y := 0; N := N_i; M := |\Gamma(i)|; k := 1
\]

**Figure 56: Diverging Algorithm**
Each vehicle entering a diverge cell, is randomly assigned to advance on one of the successor cells, using assumed turning ratios. During each simulation update interval, the vehicle flows are processed using the modified flow advancing equation as follows.

\[
y_{ij} = \min \{ x_{i \rightarrow j}^\prime, Q_i, Q_j, \delta_j x_j^\prime \}
\]

The first step of the algorithm estimates the number of vehicles eligible to advance out of the diverging cell into each successor cell, \( x_{i \rightarrow j}^\prime \). According to the flow advancing principle, the sum of the diverging flows, \( \sum_{j \in \Gamma(i)} y_{ij} \), does not exceed \( S_{\text{max}} = \sum_{j \in \Gamma(i)} x_{i \rightarrow j}^\prime \). In addition, individual flows, \( y_{ij} \), cannot exceed the flow and space capacities of the target cell. As long as all lanes on the downstream cells are open, vehicles advance out the diverging cell following the order in which they entered the diverging cell. Therefore, the default behavior is to maintain the FIFO principle. However, if a vehicle, \( v \), cannot completely advance out of a diverging cell, the partially occupied lane in the target cell is declared blocked (i.e. not available for other vehicles to advance onto). Next, the number of open lanes, \( N \), is reduced by 1. Then, if no other lanes are available in the target cell, the number of open cells, \( M \), is decreased by 1. The algorithm terminates if the check for the termination criteria holds true. The termination criterion is met if the total number of vehicles advancing out of cell \( i \), \( Y \), reaches \( S_{\text{max}} \), or if all target cells become blocked (\( M = 0 \)).

Otherwise, if \( v \) advances completely into target cell \( j \), then the same termination criteria are used. If the termination criteria are not met, then the next vehicle eligible to advance out of the diverging cell is processed. The diverging algorithm ends either when
one of the termination criteria holds true, or when all vehicles eligible to advance have been processed.

Note that when vehicles are blocked at the exit of the diverging cell, the mandatory lane-changing logic may advance vehicles from behind the blocked ones, which leads to violations of the FIFO rule. Similarly, in real-life networks, drivers may execute lane-changing maneuvers to avoid congestion. Therefore, controlled FIFO violations can account for this realistic driving behavior, and this diverging logic accounts for such situations.

6.3 Experimental Analysis

In this section a numerical example is used to illustrate the employment of the diverging algorithm in CTM. For this purpose a sample network described in Figure 57 was used. The network consists of a source cell that releases vehicles into a three-lane freeway. Lanes 3 of the diverging cell, 4, leads to destination sink 2, through cells 5 and 6. Lanes 1 and 2 continue on the main stream towards sink 1 through cells 8 and 9. Cell 9 on the freeway restricts the traffic from two lanes to one lane.

Figure 57: Network layout and topology
The network parameters in terms of cell length, speed, capacity, and number of lanes are listed in Table 15. The free-flow and backward moving wave speeds were set to 60 and 13.5 mph, respectively, for all cells. Other assumed parameters are a flow capacity of 2200 pcphpl and a jam density of 200 pcpm. The simulation update time was derived from the free-flow travel time of the shortest cell and was set to 1056'/(60mph*1.47) = 12 seconds.

Table 15: Network details

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>3</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>4</td>
<td>4224</td>
<td>60</td>
<td>13.5</td>
<td>3</td>
<td>2200</td>
</tr>
<tr>
<td>5</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>6</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>8</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>9</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
</tbody>
</table>

The vehicles are systematically released from the source cell onto all three lanes of the freeway in two stages, as detailed in Table 16. The simulation begins at 8:00 and in the first stage, the vehicles are released at a flow rate of 3600 pcph during a 30-minute period. During the second period, from 8:45 to 9:15, the vehicles are released at a much lighter flow rate of 600 pcph. For both simulation periods vehicles are sent from the source cell in a systematic fashion into all three lanes to create an even distribution in lane occupancy. The even distribution in lane occupancy is necessary in order to test the ability of the diverging model to capture non-homogeneous traffic flows across diverging cell lanes under various traffic conditions.
Table 16: Traffic demand

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source</td>
<td>8:00:00</td>
<td>30</td>
<td>3600</td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>8:45:00</td>
<td>30</td>
<td>600</td>
</tr>
</tbody>
</table>

The network shown in Figure 57 is simulated under three scenarios that use the same demand from Table 16, but assume different turning ratios. The first scenario assumes an 85/15 turning ratio, on the main freeway and off-ramp, respectively. The second scenario assumes a 50/50 turning ratio and the third scenario assumes a 15/85 turning ratio. For each scenario, the effect of the diverging algorithm on the lane-by-lane distribution of flows inside cell 4 is next.

6.3.1 Case 1: Diverging Turning Ratio 85/15

The lane-by-lane occupancy distribution for case 1 is plotted in Figure 58. It can be seen that for nearly the first 10 minutes of simulation the lane occupancy for each of the three lanes remains relatively constant with small random variations. Even though vehicles are released from the source cell with even distribution across all three lanes, in the diverging cell 4, lane 3 has less occupancy than lanes 1 and 2. This difference can be explained by the 85/15 turning ratio. During the first simulation period this turning ratio leads to a demand of $3600 \times \frac{85}{100} = 3060$ pcpd targeting cells 8 and cell 9, but cell 9 has a flow capacity of 2200 pcpd. Consequently, the entrance of cell 9 acts like a bottleneck that generates queues propagating upstream into the diverging cell. This explains the higher discrepancy in the occupancy of the three lanes stating with simulation time 8:15, when congestion reaches the diverging cell.
Figure 58: Lane occupancy in a diverging cell with 85/15 split ratio

Note from Figure 58 that the occupancy of the diverging cell drops to zero at simulation time 8:42:12. The vehicles released in the second demand period enter diverging cell at simulation time 8:47:24, when the congested traffic from the previous demand has already cleared the cell. Figure 58 shows in the second part of the simulation a similar trend in the distribution of flows across the lanes. The magnitudes of the lanes’ occupancies are smaller because of the reduced demand (600 pcph). In addition, some random variations can be seen, but they can be explained by the randomness in lane changing maneuvers.
6.3.2 Case 2: Diverging Turning Ratio 50/50

The lane-by-lane occupancy of the diverging cell in the second scenario is shown in Figure 59. It can be seen that during the first demand period (from simulation time 8:00 to 8:30) lane 3 has a somewhat higher occupancy than lanes 2 and 1. This is explained by the split ratio used. In this scenario half of the vehicles that enter the diverging cell continue on the main freeway, and consequently these vehicles target to travel in lanes 1 and 2. The other half of the vehicles need to select lane 3 to travel onto, because they have to exit toward cell 5. On both diverging streams the flow rate is \( \frac{3600 \text{pcph}}{2} = 1800 \text{ pcph} \), and it does not exceed the flow capacity of any downstream cell. In other words, in the second scenario, the combination of demand and split ratio does not create congested conditions. A somewhat similar pattern can be observed in the lane-by-lane occupancy of the diverging cell during the second demand period, between simulation time 8:45 and 9:15. However, the difference in the occupancies of the three lanes is less than that in the first period because of the very light demand. Under light demand and 50/50 split ratio, the lane-changing maneuvers have rather a visible impact on the random variations than on lane occupancies.

6.3.3 Case 3: Diverging Turning Ratio 15/85

In this scenario, on random basis, 85% of the vehicles that enter the diverging cell select cell 5 as the next cell to advance onto, and the remaining 15% select cell 8.
During the first demand period (8:00-8:30) the vehicles attempt to exit from the diverging cell on the off-ramp (cell 5), at a flow rate of 3600pcph*85/100=3060 pcph. This flow rate exceeds the 2200 pcpf flow capacity of the one-lane cell 5. Therefore, it is expected that a queue builds on lane 3 of the diverging cell, because it leads to cell 5. The effect of the excess demand for the capacity of the off-ramp can be seen in Figure 60.
Figure 60: Lane occupancy in a diverging cell with 15/85 split ratio

It is shown in Figure 60 that the occupancy of lane 3 rapidly increases until it reaches nearly 1/3 of the space capacity of the diverging cell ($\frac{4224'}{5280'}*200\text{pcpm} = 160\text{pc}$). Similar increasing trends in occupancy of lanes 1 and 2 are shown in Figure 60, but the middle lane 2 has a significantly higher occupancy than the most left lane 1. This increase in occupancy is explained by the mandatory lane-changing effect. For example, some more aggressive drivers, targeting cell 5 and traveling in lanes 1 and 2 may choose to defer the necessary lane-changing maneuvers that would bring them into lane 3. Those drivers may hold traffic behind them in the same travel lane, which leads to an increase in lane occupancy.
In addition, the plots in Figure 60 show that during the second demand interval all three lanes have nearly constant occupancies, with lane 3 slightly higher than lane 1 and 2. This is explained by the 600 pcp/h light demand released from the source cell under the 15/85 assumed turn ratio. Some random variations can be seen in the occupancy of each lane, inside the diverging cell, but they are simply the effect of random lane-changing behavior.

6.4 Summary

This chapter presents an enhanced diverging algorithm for CTM and the testing of this algorithm with a sample network is demonstrated. The developed logic has special treatments for the vehicle streams that split at a diverging junction under various traffic and geometric conditions. For example, it accounts for lane blockage in the downstream cell due to non-discrete vehicle movements and it allows for non-homogeneous distribution of flows across lanes.

The analysis of the diverging algorithm was performed with a sample network in which deterministic a priori turn ratios are used to construct three diverging scenarios. It is shown that various non-homogeneous conditions can be captured with the implemented algorithm. For example, it was shown that if one of the diverging traffic streams operates under congested conditions, the congestion is reflected in higher occupancy inside the diverging cell for the lanes leading to the congested stream. This was explained by the queues that spill back into the diverging cell. The behavior is believed to be intuitive for the congested real-life freeways, where often times during the morning and afternoon peak periods the traffic on the off-ramps may spill back into the freeway, and cause similar
effects. Therefore, if employed in a CT framework, the diverging algorithm is expected to contribute to the overall realistic representation of traffic flow.
CHAPTER 7  RANDOM DRIVING BEHAVIOR

7.1  Introduction

Random variations in traffic networks may play a significant role in determining the representation accuracy when modeling traffic networks. The original form of CTM, however, is presented in a deterministic context. This is because the parameters controlling the flow advancing equations, even when time-dependent, are assumed deterministic throughout the simulation period. This may be a reasonable approximation by CTM in planning analysis. However, for operational analysis, this may compromise the model’s accuracy. In this chapter a possible alternative to modeling CTM in a stochastic fashion is presented. First, several sources of stochastic variation in driver behavior are identified. Next, methodologies that account for each source of randomness are presented. Lastly, a numerical example illustrates how real-life microscopic data can be used in deriving the stochastic variations of one of the parameters used in CTM.

7.2  Accounting for Random Variation in CTM

While random characteristics in traffic networks may be best described by microscopic simulation models using car-following and lane-changing behavioral models, it may be argued that random variations can also be modeled in macroscopic/mesoscopic simulation environments. This is because the random behavior of individual drivers at the microscopic level can be captured in the observed randomness of the aggregate behavior of traffic in the network at the macroscopic level. For instance, random variations in the
minimum headways essentially lead to random variations in the observed short-term flow capacity. Similarly, the random variations in the minimum space headway lead to random variations in the observed local jam densities.

In general, the aggregate macroscopic characteristics such as density and flow inherit randomness from their respective disaggregate microscopic characteristics such as spacing and headway. This implies that the probability distribution of a macroscopic parameter such as flow capacity can be derived from the probability distribution of the minimum headways, if known. Consequently, some parameters in the modified flow advancing equation may be replaced with random variables that are generated from an assumed probability distribution as shown in Eq. (46).

\[
\tilde{y}_{ij} = \min \left\{ \tilde{x}_i', \tau \tilde{Q}_i, \tau \tilde{Q}_j, \tilde{\delta}_j (\tilde{X}_j - x_j) \right\} \tag{46}
\]

\[
\tilde{\delta}_j = \begin{cases} 
1 & \text{if } \tilde{x}_i' \leq \tau \min \left( \tilde{Q}_i, \tilde{Q}_j \right) \text{(free-flow conditions)} \\
\frac{\tilde{W}_j}{v_j^\text{max}} & \text{otherwise (forced-flow conditions)}
\end{cases} \tag{47}
\]

Where,
\[
\tilde{y}_{ij} = \text{a realization of a random variable representing the number of vehicles advancing from cell } i \text{ to cell } j,
\]
\[
\tilde{x}_i' = \text{a realization of a random variable representing the number of vehicles eligible to advance out of cell } i,
\]
\[
x_j = \text{the number of vehicles occupying cell } j,
\]
\[
\tilde{X}_j = \text{a realization of the random variable representing the maximum space capacity of cell } j,
\]
\( \tilde{Q}_i, \tilde{Q}_j \) = realizations of random variables representing the flow capacities of cells \( i \) and \( j \), respectively,

\( \tau \) = the simulation scan time

\( \tilde{\delta}_j \) = a realization of the random variable representing the rate at which vehicle spaces in cell \( j \) can be utilized by advancing vehicles.

\( \tilde{W}_j \) = a realization of the random variable representing the speed of the backward moving wave under congested conditions in cell \( j \).

\( V_{j}^{\text{max}} \) = the maximum free-flow speed in cell \( j \).

All the above realizations of random variables can be generated from some probability distribution with known parameters (e.g. mean and variance). To account for stochastic variations in driving behavior in CTM, four sources of randomness are identified: free-flow speed, minimum headway, minimum spacing, and wave speed.

### 7.2.1 Accounting for Randomness in Free-Flow Speed

In CTM free-flow speeds are assumed to be constant for each cell and are consequently used to determine the free-flow travel time of that cell. In reality, drivers choose their own free-flow speed under light traffic conditions, which may be slightly higher or lower than the cell free-flow speed. Variations in free-flow speeds essentially lead to variations in the cell free-flow travel times for individual vehicles. To prevent vehicles with relatively high free-flow speeds from skipping cells, the cell free-flow speed must be set to the absolute maximum free-flow speed of any individual vehicle in the stream, rather than the mean free-flow speed.
The free-flow speed $V$ of some vehicle can be approximated by a triangular distribution with a maximum speed $V^{\text{max}}$, a minimum speed $V^{\text{max}} - \varepsilon$, and a mode $V^{\text{max}} - \varsigma$. An example of a triangular distribution that has a probability density function, $f(V)$, and a cumulative distribution function, $F(V)$, is shown in Figure 61.

![Figure 61: A hypothetical triangular distribution](image-url)
To introduce random variation in free-flow speeds, two additional properties must be defined for any vehicle, \( v \): free-flow speed (\( u^v \)) and wait (\( w^v \)). The free-flow speed property value is assigned to each vehicle at the time of its release into the network using an assumed distribution of free-flow speeds (e.g. normal or triangular). The wait property, \( w^v \), is reset to zero every time a vehicle enters a subcell inside cell \( i \). Recall that long cells are split into equal length subcells that have a free-flow travel time, \( \tau \), equal to the simulation time scan. Because \( u^v \leq V^\text{max}_i \), where \( V^\text{max}_i \) is the maximum free-flow speed in cell \( i \), vehicle \( v \) may need more than \( \tau \) time units to advance between two consecutive subcells. Another property for vehicle \( v \) is needed, \( \tau^v \) - the vehicle minimum wait time and represents the vehicle’s free-flow travel time inside a subcell defined as follows:

\[
\tau^v = \tau \frac{V^\text{max}_i}{u^v}
\]  

(48)

During each simulation update, the vehicle’s wait time, \( w^v \), is incremented with \( \tau \) to reflect the total time spent in the current subcell. It follows that \( v \) is allowed to advance between two consecutive subcells, if \( w^v \geq \tau - \tau^v \).

The random variation in the vehicles’ free-flow speeds, \( u^v \), leads to random variation in the number of vehicles eligible to advance out of cell \( i \), \( x^i' \). Therefore, a random realization of \( x^i' \), say \( \tilde{x}^i' \), is defined as:

\[
\tilde{x}^i' = x^i' \frac{\tilde{V}_i}{V^\text{max}_i}
\]  

(49)

Where,

\( \tilde{V}_i \) = the average free-flow speed of vehicles in cell \( i \),
It should be noted that \( \frac{\tilde{V}_i}{V_i^{\max}} \leq 1 \). An illustration of how the average free-flow speed of vehicles, \( \tilde{V}_i \), can be selected such that the ratio \( \frac{\tilde{V}_i}{V_i^{\max}} \) is always less than one is shown in Figure 62. In Figure 62 \( P(V_i) \) represents the cumulative distribution of the mean free-flow speed, bounded by \( V_i^{\min} \) and \( V_i^{\max} \). A random number uniformly distributed from \([0,1]\) can be used to find out the probability that \( P\{V_i \geq \tilde{V}_i\} \).

![Figure 62: Random sampling of mean free-flow speed of vehicles in cell \( i \)](image)

### 7.2.2 Accounting for Randomness in Flow Capacity

Considering the random behavior of drivers in maintaining a minimum headway at capacity conditions, the variations in flow capacity of a cell \( i \), \( Q_i \), can be derived as follows:
\[ \hat{Q}_i = \frac{N_i}{\bar{h}_i} \]  

(50)

Where,

\( \bar{h}_i \) = the realization of the mean minimum headway of vehicles in cell \( i \),

\( N_i \) = the number of lanes in cell \( i \).

The cumulative distribution function of the mean minimum headway, \( P(h_i) \), bounded by \( h_i^{\text{min}} \) and \( h_i^{\text{max}} \), can be used to randomly generate \( \bar{h}_i \) as shown in Figure 63.

![Figure 63: Random sampling of mean minimum headway of vehicles in cell \( i \)](image)

7.2.3 Accounting for Randomness in Space Capacity

Similar assumptions can be made to introduce random behavior in space capacity of cell \( j \) at jam density conditions. Individual driver behavior is characterized by certain aggressiveness in maintaining a minimum spacing between vehicles under stopped traffic conditions. The minimum spacing is measured between the rear bumpers of two
consecutive vehicles. Different drivers maintain different minimum spacing between vehicles under congested traffic conditions. Therefore, variations in the minimum spacing in cell \( j \) lead to variations in jam density values for that cell: 

\[ \tilde{X}_j = \frac{N_j}{\tilde{s}_j} \]

where, \( \tilde{s}_j \) = the average minimum spacing between vehicles in cell \( j \), and \( N_j \) = the number of lanes in cell \( j \). The cumulative distribution function of the mean minimum spacing, \( P(s_j) \), bounded by \( s_j^{\text{min}} \) and \( s_j^{\text{max}} \), can be used to randomly generate \( \tilde{s}_j \) as show in Figure 64.

![Figure 64: Random sampling of mean minimum spacing of vehicles in cell \( i \)](image)

7.2.4 Accounting for Randomness in \( \delta_j \)

Random variation in the rate at which vehicle spaces in the receiving cell \( j \) can be utilized is accounted for using \( \delta_j \).
\begin{equation}
\delta_j = \frac{\tilde{W}_j}{V_{j_{\text{max}}}}
\end{equation}

The realization of backward wave speed, \( \tilde{W}_j \), can be obtained from Figure 65.

Figure 65 shows how a random realization the wave speed can be obtained based on the cumulative distribution of the mean backward moving speed, \( P(W_j) \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure65}
\caption{Random sampling of wave speeds in cell \( j \)}
\end{figure}

Thus, all the random variations in driving behavior can be included in the flow advancing equation that is re-written as follows:

\begin{equation}
\tilde{y}_{ij} = \min \left\{ x_i', \frac{\tilde{V}_i}{\tilde{V}_{i_{\text{max}}}}, \min \left[ \frac{N_i}{h_i}, \frac{N_j}{\tilde{h}_j} \right], \tilde{W}_j, \frac{N_j}{\tilde{s}_j} \right\}
\end{equation}

\( (52) \)
7.3 Experimental Analysis

To test the effect of the stochastic component CTM it is necessary to make certain assumptions about the type and parameters of the probability distribution functions outlined in section 7.2. Determining the distributions of the individual vehicle speeds, minimum headway, and minimum spacing is beyond the scope of this study. The Central Limit Theorem can be applied to assume that the means of free-flow speed, minimum headway, and minimum spacing of vehicles inside the cells are normally distributed.

7.3.1 Modeling Variations in $\delta_j$

The objective of this study is to test the assumption that random variations in $\delta_j$ can be attributed to variations in traffic conditions. This parameter was selected because is the most important parameter to study as it affects the behavior during congestion. This task was achieved with an empirical approach using the publicly available vehicle trajectories data from the Next Generation Simulation (NGSIM) website. NGSIM is a group of public and private organizations that focus on developing open behavioral algorithms in support of traffic simulation with a primary focus on microscopic modeling. However, for this particular application the microscopic data was selected for calibration/validation of the randomness in wave speed of the proposed mesoscopic traffic simulator. Several datasets are available at http://www.ngsim.fhwa.dot.gov. The selected dataset for this study was collected on April 13, 2005 during the afternoon peak period (i.e. between 4:00 and 4:15 pm). This dataset contains 15 minutes of vehicle trajectories from the 1650 ft long segment of Interstate 80 in Emeryville (San Francisco), California. Figure 66 shows the layout of the data-collection site, and Table 17 presents the dataset structure
Figure 66: NGSIM study area
Table 17: NGSIM data structure

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Following Vehicle</td>
<td>Vehicle Id of the vehicle following the subject vehicle in the same lane. A value of '0' represents no following vehicle - occurs at the beginning of the study section and on-ramp due to the fact that only complete trajectories were recorded by this data collection effort (vehicle that did not traverse the downstream boundaries of the section by the end of the study period were not recorded).</td>
<td>Number</td>
</tr>
<tr>
<td>2</td>
<td>Frame ID</td>
<td>Frame Identification number (ascending by start time)</td>
<td>1/10 of a second</td>
</tr>
<tr>
<td>3</td>
<td>Global Time (Epoch Time)</td>
<td>Elapsed time since Jan 1, 1970.</td>
<td>Milliseconds</td>
</tr>
<tr>
<td>4</td>
<td>Global X</td>
<td>X Coordinate of the front center of the vehicle based on CA State Plane III in NAD83.</td>
<td>Feet</td>
</tr>
<tr>
<td>5</td>
<td>Global Y</td>
<td>Y Coordinate of the front center of the vehicle based on CA State Plane III in NAD83.</td>
<td>Feet</td>
</tr>
<tr>
<td>6</td>
<td>Headway (Time Headway)</td>
<td>Headway provides the time to travel from the front-center of a vehicle (at the speed of the vehicle) to the front-center of the preceding vehicle. A headway value of 9999.99 means that the vehicle is traveling at zero speed (congested conditions).</td>
<td>Seconds</td>
</tr>
<tr>
<td>7</td>
<td>Lane Identification</td>
<td>Current lane position of vehicle. Lane 1 is farthest left lane; lane 6 is farthest right lane. Lane 7 is the on-ramp at Powell Street, and Lane 9 is the shoulder on the right-side.</td>
<td>Number</td>
</tr>
<tr>
<td>8</td>
<td>Local X</td>
<td>Lateral (X) coordinate of the front center of the vehicle with respect to the left-most edge of the section in the direction of travel.</td>
<td>Feet</td>
</tr>
<tr>
<td>9</td>
<td>Local Y</td>
<td>Longitudinal (Y) coordinate of the front center of the vehicle with respect to the entry edge of the section in the direction of travel.</td>
<td>Feet</td>
</tr>
<tr>
<td>10</td>
<td>Preceding Vehicle</td>
<td>Vehicle Id of the lead vehicle in the same lane. A value of '0' represents no preceding vehicle - occurs at the end of the study section and off-ramp due to the fact that only complete trajectories were recorded by this data collection effort (vehicles already in the section at the start of the study period were not recorded).</td>
<td>Number</td>
</tr>
<tr>
<td>11</td>
<td>Spacing (Space Headway)</td>
<td>Spacing provides the distance between the front-center of a vehicle to the front-center of the preceding vehicle.</td>
<td>Feet</td>
</tr>
<tr>
<td>12</td>
<td>Total Frames</td>
<td>Total number of frames in which the vehicle appears in this data set.</td>
<td>1/10 of a second</td>
</tr>
<tr>
<td>13</td>
<td>Vehicle Acceleration</td>
<td>Instantaneous acceleration of vehicle</td>
<td>Feet/Second Square</td>
</tr>
<tr>
<td>14</td>
<td>Vehicle Class</td>
<td>Vehicle type: 1 - motorcycle, 2 - auto, 3 - truck</td>
<td>Text</td>
</tr>
<tr>
<td>15</td>
<td>Vehicle ID</td>
<td>Vehicle identification number (ascending by time of entry into section)</td>
<td>Number</td>
</tr>
<tr>
<td>16</td>
<td>Vehicle Length</td>
<td>Length of vehicle</td>
<td>Feet</td>
</tr>
<tr>
<td>17</td>
<td>Vehicle Velocity</td>
<td>Instantaneous velocity of vehicle</td>
<td>Feet/Second</td>
</tr>
<tr>
<td>18</td>
<td>Vehicle Width</td>
<td>Width of vehicle</td>
<td>Feet</td>
</tr>
</tbody>
</table>
Each row contains information on one vehicle about its location, speed, headway, spacing, vehicle length, traveling lane, immediate predecessor and successor vehicle, and others, sampled at 0.1 seconds. The raw data was imported in a database format to facilitate data pre-processing. First, data from lane 1, HOV lane, and lane 6, auxiliary lane was discarded in order to remove highly non-homogeneous traffic conditions usually expected to be seen in those types of lanes. Each lane of the freeway segment was split into cells of equal length to facilitate modeling of traffic flow in CTM.

\[
y_{ij} = \min \left\{ x_i', Q_j, \tau Q_j, \delta_j (X_j - x_j) \right\}
\]

(53)

\[
\delta_j = \begin{cases} 
1 & \text{if } x_i \leq \tau Q_j \text{ (free-flow conditions)} \\
\frac{W_j}{V_j} & \text{otherwise (forced-flow conditions)}
\end{cases}
\]

(54)

One can use Eq. (53) to estimate \( \delta_j \) assuming that under congested traffic conditions the dominant term in the minimum function is \( \delta_j (X_j - x_j) \). Consequently, \( \delta_j \) can be estimated as follows:

\[
\delta_j = \frac{y_{ij}}{X_j - x_j}
\]

(55)

Several steps were executed to filter the records that correspond to congested traffic conditions. First, the occupancy of each cell \( i \), \( x_i \) was computed for all the cells at every \( \tau \) seconds within the 15-minute period. The occupancy of each cell depends on the mix of vehicle types (i.e. passenger cars, buses, trucks, etc.) residing in that cell. Next, to account for the effect of vehicle types on the cell occupancy, \( x_i \) was calculated as the total length of the vehicles occupying the cell. Consequently, the jam density, \( X_j \), is set to the cell length. The maximum speed observed in the dataset was nearly 98ft/sec. Therefore,
for each cell length the free-flow travel time, $\tau$, was calculated assuming a maximum free-flow speed of 100ft/sec (68 mph), to ensure that no vehicle skips cells. In order to capture various levels of traffic conditions in traffic flow three scenarios, each with different cell length were considered. The three scenarios assume cells lengths of 100 ft, 500 ft, and 1000 ft, and free-flow travel times of 1 sec, 5 sec and 10 sec, respectively. An illustrative example that shows how $\delta_j$ can be calculated is depicted in Figure 67.

![Figure 67: Illustrative example of $\delta_j$ computation](image)

In this example $n(t,x)$ represents the cumulative effective length of vehicles passing a certain location $x$ at a certain time $t$. The effective length of each vehicle is the sum of the physical length of the vehicle, from the dataset, and the spacing between its front bumper and the rear bumper of the leading vehicle. $n(t,x)$ values were calculated
assuming a discretization of the time and space of 0.1 seconds and 1 foot, respectively. The example in Figure 67 shows how $x_i$, $x_j$ and $y_{ij}$ were calculated for two interconnected cells, $i$ and $j$. For all cells representing the studied section the maximum observed value of $y_{ij}$ was assumed as the flow capacity value for the given dataset. Consequently, the records for which $x_i > y_{ij}$ were selected to represent congested traffic conditions. For all these records equation (55) was used to compute $\delta_j$. Additionally, the occupancy of each cell was divided by $X_{\text{max}}$ and 5% increments were used to generate twenty bins, from zero to one. The final dataset contains nearly 4 million records, and each holds a combination of categorical values, $x_i$, and $x_j$, and corresponding $\delta_j$ value. The mean, standard deviation, and 95% confidence intervals were calculated to determine the variation in $\delta_j$ for each $x_i$ and $x_j$ pair. A power analysis with a tolerance of 1% was conducted to eliminate the $x_i$ and $x_j$ combinations that yield a high variation in $\delta_j$, and thus have less confidence.

Figure 68 shows how the distribution of means for $\delta_j$ varies with the relative occupancy in the receiving cell, $x_j$, if the occupancy of the sending cell, $x_i$, is from the 10-15% bin. It can be seen that $\delta_j$ varies in an ascending trend with the increase in $x_j$, approximately between 0.025 and 0.2. In addition, it can be seen that the variance of the mean increases with $x_j$, but there are no observations for $x_j > 70\%$. The missing data can be explained by the fact that for some $x_i$, $x_j$ combinations $\delta_j$ has high variance and the 1% tolerance criterion is not met. However, missing data points can be approximated by nonlinear high order polynomials interpolation.
A data fitting exercise was conducted for all distribution of means for $\delta_j$ that yielded from the analyzed I-80 dataset. Several polynomials were tested and the third order polynomial was selected to approximate $\delta_j$ because it provided the best fit for most distributions. Table 18 provides the final form of all the polynomials fitted. It can be seen that the goodness ranges between 0.95 and 0.994.

Figure 68: Distribution of means for $\delta_j$ ($x_i = 10-15\%$)
Table 18: Third order polynomials for fitting $\delta_j$

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>$\delta_j = a_3x_j^3 + a_2x_j^2 + a_1x_j + a_0$</th>
<th>R-square</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5%</td>
<td>$0.0002x_j^3 - 0.0005x_j^2 + 0.0162x_j + 0.2287$</td>
<td>0.992</td>
</tr>
<tr>
<td>10-15%</td>
<td>$0.0002x_j^3 - 0.0025x_j^2 + 0.0178x_j + 0.1194$</td>
<td>0.994</td>
</tr>
<tr>
<td>15-20%</td>
<td>$0.001x_j^3 - 0.0022x_j^2 + 0.0158x_j + 0.1082$</td>
<td>0.994</td>
</tr>
<tr>
<td>20-25%</td>
<td>$0.004x_j^3 + 0.0009x_j^2 + 0.0025x_j + 0.1557$</td>
<td>0.973</td>
</tr>
<tr>
<td>25-30%</td>
<td>$-0.00007x_j^3 + 0.0023x_j^2 - 0.0095x_j + 0.1833$</td>
<td>0.958</td>
</tr>
<tr>
<td>30-35%</td>
<td>$0.00009x_j^3 + 0.0013x_j^2 - 0.0061x_j + 0.1893$</td>
<td>0.983</td>
</tr>
<tr>
<td>35-40%</td>
<td>$-0.00008x_j^3 + 0.0038x_j^2 - 0.0151x_j + 0.2103$</td>
<td>0.982</td>
</tr>
<tr>
<td>40-45%</td>
<td>$-0.00003x_j^3 + 0.0017x_j^2 - 0.0039x_j + 0.1935$</td>
<td>0.980</td>
</tr>
<tr>
<td>45-50%</td>
<td>$-0.0001x_j^3 + 0.0041x_j^2 - 0.0198x_j + 0.2389$</td>
<td>0.978</td>
</tr>
<tr>
<td>50-55%</td>
<td>$-0.00002x_j^3 + 0.0017x_j^2 - 0.0076x_j + 0.2351$</td>
<td>0.994</td>
</tr>
<tr>
<td>55-60%</td>
<td>$0.0002x_j^3 - 0.0031x_j^2 + 0.0293x_j + 0.159$</td>
<td>0.977</td>
</tr>
<tr>
<td>70-75%</td>
<td>$-0.00009x_j^3 + 0.0036x_j^2 - 0.0202x_j + 0.2689$</td>
<td>0.950</td>
</tr>
</tbody>
</table>

Figure 69 plots the polynomial fitting for $\delta_j$, superimposed on the derived distributions of the means for four $x_i$ bins covering values between 0-5% and 10-25%. All four polynomials have high order coefficients $a_3$, and $a_2$ that are considerably smaller than the low order coefficients, $a_1$, and $a_0$. However, the former cannot be ignored because their contribution in the overall goodness fit is necessary to capture non-linear variations of $\delta_j$.

Two distinct patterns can be identified in Figure 69. The fitted polynomial for $x_i = 0-5\%$ and the other three polynomials for $x_i = 10-25\%$. The former covers the high values of $x_j$, between 55-100%, while the latter are applicable for $x_j$ ranging between 0 and 70%. This difference can be explained by the fact that when congestion triggers in the
receiving cell $j$ the occupancy of the sending cell $i$ is much lower (i.e. congested vs. free-flow). Moments after the congestion propagates backward cell $i$ increases its occupancy getting more congested, while cell $j$ reduces its occupancy because of reduced inflow from cell $i$. It is believed that the extrapolated areas of the polynomials may not have a practical meaning. If more data is available for the missing data points then a new fitting should be done.

![Figure 69: Polynomial fitting for $\delta_j$ and $x_i \in [0.1-0.25]$](image)

Similarly, Figure 70 shows five fitted polynomials for $\delta_j$, one for each 5% bin of $x_i$ between 25% and 50%. The dataset plots show that $\delta_j$ varies approximately between 0.08 and 0.26. For similar reasons the extrapolated values of $\delta_j$ outside the range of $x_j$
from the dataset, should be carefully used. If available, more data should be considered to validate the $\delta_j$ values resulted from extrapolation. Figure 71 plots last three fitted polynomials for $\delta_j$, for $x_i = 50\%-55\%$, $x_i = 50\%-55\%$, and $x_i = 70\%-75\%$. The dataset plots show that $\delta_j$ varies approximately between 0.09 and 0.23.

![Figure 70: Polynomial fitting for $\delta_j$ and $x_i \in [0.25 - 0.50]$](image_url)
7.3.2 **The Effect of Dynamic $\delta_j$ in CTM**

The effect of using a dynamic $\delta_j$ was investigated by simulating a freeway segment. The network topology is detailed in Figure 72. As shown in Table 19 the 2-lane freeway has a downstream one-lane bottleneck capacity of 2200 pcph. The simulation update time was 3 seconds. Other parameters used are free-flow and backward moving wave speeds of 60 mph and 13.5 mph and a jam density of 200 pcph. Vehicles are released into cell 2 using the demand from Table 20. For the first 20 minutes of the simulation (8:00 – 8:20) they are released at a flow rate that exceeds the flow capacity of the one lane segment (2700 pcph > 2200 pcph). Therefore it is expected that forced-flow conditions are created at the entrance of cell RC, and propagate backwards through the network.
Figure 72: Testing dynamic variations in $\delta_j$ (network topology)

Table 19: Testing dynamic variations in $\delta_j$ (network description)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>1100</td>
</tr>
<tr>
<td>1</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>2</td>
<td>10560</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
</tbody>
</table>

Table 20: Testing dynamic variations in $\delta_j$ (traffic demand)

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source</td>
<td>08:00:00</td>
<td>20</td>
<td>2700</td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>08:20:00</td>
<td>30</td>
<td>600</td>
</tr>
</tbody>
</table>

A base case scenario is generated using the a fixed value for $\delta_j$ from Eq. (54), which implies a constant wave speed. A comparative case is created using the fitted polynomials from Table 18 to account for dynamic variations in $\delta_j$, and implicitly in the wave speed. Two performance measures are evaluated, the occupancy of cell 1 over time and the total network travel-time.

In Figure 74, the variations over time of the occupancy of cell 1 are plotted. It can be seen that there is a similar pattern in the evolution of the traffic flow inside the cell for the one-hour simulated period. For example, in both scenarios, the onset of congestion
is triggered at the same time (08:02:00) and the demand is cleared at the same time (08:52:00). However, there is a difference in the transitional period during the queue propagation through the congested cell 1. It appears that the congestion takes longer to propagate in the stochastic case (i.e. cell 1 reaches a higher occupancy nearly 110 vehicles, vs. about 100 vehicles in the deterministic case). In addition, after the queue reaches the entrance of cell 1, marked by the drop in occupancy, the second case shows less stable densities inside cell 1. This can be explained by the variable $\delta_j$ values that are dependent of the dynamically changing traffic conditions within the eight subcells of cell 1. Interestingly, the total network travel-time in both scenarios is very similar, the absolute relative error is 0.025% for a magnitude of 88.84 vehicle-hours.

![Graph](image)

Figure 73: Effect of constant $\delta_j$ on the occupancy of cell 1
7.4 Summary

This chapter introduces a possible methodology to account for random driving behavior in CTM. It is suggested that the model’s parameters should be replaced with stochastic variables, and random realizations of these variables should be used when modeling the network. The macroscopic parameters in CTM can be modeled as stochastic variables through their microscopic counterparts. For example, the mean free-flow speed of individual drivers residing in a cell $i$ may lead to variations in the cell occupancy, $x_i$. Other stochastic parameters are the flow rate dependent of the mean minimum headway, the flow capacity dependent of the mean spacing capacity, and $\delta_j$ dependent of the variations in the wave speed, $W_j$. 

Figure 74: Effect of dynamic variations in $\delta_j$ on the occupancy of cell 1
An empirical method is presented to account for a dynamic change in $\delta_j$ based on the prevailing traffic conditions using vehicle trajectories. This method uses vehicle trajectories from a section of Interstate-80, near San Francisco. The freeway section was split into imaginary cells and the raw trajectory data was processed to retain only forced-flow traffic conditions. The flow advancing equation was used to extract a relationship between $\delta_j$ and a set of discretized values of occupancies in the upstream and downstream cells, $x_i$ and $x_j$, respectively. A family of third order polynomials were fitted to facilitate the estimation of $\delta_j$ for a continuous range of values for $x_i$ and $x_j$.

A sample freeway network was simulated assuming a constant value for $\delta_j$ versus a dynamically changing value derived from the fitted polynomials. The variation of a cell that captures both congested and free-flow traffic was compared. It was shown that during congested conditions there is some differences in the variation of cell occupancy over time. However, at the macroscopic level with a relative absolute error of 0.025%, the total network travel time in both cases was nearly the same.
CHAPTER 8  MODELING MULTI-MODAL FLOWS

8.1  Introduction

Another source of variation in the flow advancing equation may be attributed to the variation in vehicle types in the traffic stream. The current form of CTM applies to single-mode traffic flow (e.g., passenger cars only). This chapter presents a solution to account for the effect of traffic mix on the flows modeled with CTM. A hypothetical network is simulated to demonstrate the proposed treatment of multi-modal flows in CTM.

8.2  Modifications to the Flow Advancing Equation

Multiple vehicle types in a traffic stream can often be easily converted into passenger car units using the common procedure described in the highway capacity manual (2000). However, it may be difficult to ensure that the final destination and path of one large vehicle are the same as its passenger car equivalents (PCEs), especially at diverging junctions. Moreover, the effect of traffic mix on the flow and space capacity of the cells cannot be captured by this approach. This would cause misrepresentation of the actual network flow. For example, the 1000 vph demand for a network that has 5% trucks in the traffic mix, and a truck PCE of 2.5, is converted to a demand of 1075 peph. Given the modified flow advancing equation:

\[ y_{ij} = \min \{ x'_i, \tau Q_i, \tau Q_j, \delta_j (X_j - x_j) \} \]  \hspace{1cm} (56)

Since \( y_{ij} \) represents the number of passenger cars advancing from \( i \) to \( j \), equation (56) is equivalent to:
\[ y_{ij}' = \sum_{k=1}^{z} E_k \]  

(57)

Where \( E_k \) is the passenger-car equivalents for vehicle \( k \) in a stream of vehicles with size \( y_{ij}' \), which is unknown but can be determined by solving equation (57). To adjust the right-side of equation (56) the following substitutions are made. First, \( x_i' \) is replaced with \( E_i' \), where \( E_i' \) represents the total passenger car equivalents for all vehicles eligible to advance out of cell \( i \). Second \( x_j \) is replaced with \( E_j \), where \( E_j \) is the total passenger car equivalents for all vehicles occupying cell \( j \).

Equation (56) can be re-written as follows:

\[ \sum_{k=1}^{z} E_k = \min \left\{ E_i', \tau Q_i, \tau Q_j, \delta_j (X_j - E_j) \right\} \]  

(58)

And by solving equation (58), \( y_{ij}' \) can be determined as follows:

\[ y_{ij}' = \arg \min_z \left[ \sum_{k=1}^{z} E_k - \min \left\{ E_i', \tau Q_i, \tau Q_j, \delta_j (X_j - E_j) \right\} = 0 \right] \]  

(59)

The effect of random variations in driving behavior can also be accounted for in equation (59) to obtain the more generalized form of the flow advancing equation:

\[ y_{ij}' = \arg \min_z \left[ \sum_{k=1}^{z} E_k - \min \left( \frac{\tilde{V}_i}{V_{\max}} \sum_{k=1}^{z} E_{ik}, \frac{\tau}{h} \min \left( N_j, N_{ij} \right), \frac{\tilde{W}_j}{V_{\max}} \min \left( \frac{N_j}{\frac{s_j}{N_{ij} - \sum_{k=1}^{z} E_{ik}} \left( \frac{N_j}{\frac{s_j}{N_{ij} - \sum_{k=1}^{z} E_{ik}}} \right) \right) = 0 \right] \]  

(60)

8.3 Experimental Analysis

A simple freeway segment, shown in Figure 75, was used to demonstrate how the multimodal traffic flow is treated in CTM. The network parameters are described in Table 21. A bottleneck capacity is created at the entrance of the downstream one-lane cell (RC)
to trigger congested conditions. Note that the 2400 pcph demand from the source cell (Table 22) activates the bottleneck of cell RC.

![Network layout and topology](image)

Figure 75: Network layout and topology

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>1056</td>
<td>60</td>
<td>13.5</td>
<td>1</td>
<td>2200</td>
</tr>
<tr>
<td>1</td>
<td>2112</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
<tr>
<td>2</td>
<td>10000</td>
<td>60</td>
<td>13.5</td>
<td>2</td>
<td>2200</td>
</tr>
</tbody>
</table>

Table 22: Traffic demand

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source</td>
<td>08:00:00</td>
<td>60</td>
<td>2400</td>
</tr>
</tbody>
</table>

A base case scenario assumes traffic demand in passenger cars only. Other scenarios included several levels of traffic mix in the traffic stream (5%, 10%, 15%, 20%, and 25% trucks) and the same network was simulated for each level. For all scenarios the same number of vehicles was simulated and the passenger car equivalent for trucks, $E_r$, would
was set to 2.5 assuming the Highway Capacity Manual 2000 recommendation for a rolling terrain freeway. The comparison between the base case scenario and the other five scenarios is shown in Figure 76. It can be seen a significant increase in the total network travel time with the increase in the percentage of heavy vehicles (i.e. from nearly 90 vehicle-hours to about 140 vehicle-hours). This difference in travel time suggests that the effect of traffic mix can be captured in CTM with the proposed modification to the flow advancing equation.

Figure 76. Effect of traffic mix on the network travel time
8.4 Summary

This chapter presented a methodology for modeling multi-modal flows in CTM. The flow advancing equation was modified to account for variations in the traffic streams vehicles’ type by introducing a passenger-car equivalent (PCE). It was shown that various vehicle types could be associated with different PCE values such that the resulting flow advancing equation has PCE units. A freeway segment was simulated under congested conditions and assuming various percentages of trucks in the traffic mix, from no trucks present to 25%. The simulation results show an increase in network travel time with increasing percentage of trucks in the traffic stream. This demonstrates that a multi-modal behavior can be explicitly accounted for in CTM using the proposed methodology.
CHAPTER 9    CTM MODULE DEVELOPMENT

9.1 Introduction

In this chapter the implementation of a CTM simulation environment is presented. A real-life network is simulated with the developed CTM simulator and with a microscopic model, namely CORSIM. The results of the two simulations are compared using macroscopic performance measures. A series of advantages of using CTM versus CORSIM / microscopic models is presented at the end.

9.2 Software Development

All model improvements and algorithmic changes were implemented into a specially developed computer simulation module. This provides a necessary platform for testing and validation of the model. Evidently, this type of application appears is fit for development with an object-oriented programming paradigm, where cells and connectors can be structured in a hierarchy of class objects with common properties and methods. This ensures exploitation of both encapsulation and inheritance concepts in object oriented programming. Encapsulation is the process of combining elements to create a new entity in programming, while inheritance in programming allows new classes to be derived from a base class.

For this study, the computer simulation module was developed using the latest version of object-oriented Microsoft Visual Basic .NET 2003 development environment. This environment appears to provide all functional capabilities required, although other
environments may be just as suitable. Furthermore, for efficient memory use and data access speed the network data was compiled into a SQL database server at both pre-analysis and post-analysis stages. In the future, a special module will be developed to convert the user input into a set of database tables for easy and fast access by the simulation module during runtime, as well as to convert the simulation output into a user-friendly report. The modular structure of the program is detailed in APPENDIX B and a selective collection of source code to illustrate the software implementation of the developed model is provided in APPENDIX C.

9.3 Comparative Evaluation of CTM

In order to evaluate the improvements introduced in CTM a freeway network was selected from the west section of the I-10 corridor in Baton Rouge. The CTM simulation results were compared with CORSIM, a microscopic simulator frequently used by traffic researchers and practitioners as well.

9.3.1 Study Section

The selected network includes five off-ramps and four on-ramps, as depicted in Figure 78. The figure shows the network topology with a total of 15 ordinary cells, 4 merging cells, and 5 diverging cells (Table 23). The network also has 4 source cells, where traffic enters, and 6 sink cells, where traffic exits the network. The cell length was chosen to retain geometric homogeneity while minimizing the number of cells.

The 2-hour simulation period (12:00-14:00) was divided into three time periods. The first and second periods lasted for 30 minutes each with the traffic demand shown in Table 24. The third period (60 minutes) was used to clear all queues and vehicles in the
network. Congestion was induced in the network by reducing the speed limit in cell 18 to 10 mph throughout the entire simulation period. The demand flows were selected such that queues do not spill back to block the source cells. Merging priorities were set to 50% on all merging connectors. Diverging ratios for off-ramps were assumed 10% of the mainline flow. The assumed free-flow speed for main freeway cells and for on-/off-ramp cells were assumed 60 mph and 40 mph, respectively. The cell flow capacity was assumed 2250 pcp/hpl and the jam density 214 pcp/mpl. The jam density and flow capacity were derived from simulation of sample networks with CORSIM to facilitate comparisons. Based on the minimum cell length (700 ft) in the network, the simulation time step was set to nearly 8 seconds.

The same network was constructed in CORSIM by mapping each cell in CTM network to a link in CORSIM network. Traffic demand and geometric configurations were kept the same. No special calibration was performed for the microscopic parameters of CORSIM (i.e. the default values were used).
Figure 77: A segment of I-10 freeway (west section)
Figure 78: A segment of I-10 freeway (east section)
Table 23: I-10 freeway (network details)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4650</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>11a</td>
<td>1050</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>900</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>13a</td>
<td>1050</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>2100</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>15b</td>
<td>800</td>
<td>60</td>
<td>1</td>
</tr>
<tr>
<td>16a</td>
<td>700</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>16b</td>
<td>800</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>17a</td>
<td>800</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>2550</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>19b</td>
<td>1500</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>20a</td>
<td>700</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>20b</td>
<td>3350</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>4b</td>
<td>2150</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>4a</td>
<td>1050</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>5a</td>
<td>1050</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1800</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>7b</td>
<td>1350</td>
<td>60</td>
<td>1</td>
</tr>
<tr>
<td>8a</td>
<td>700</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>8b</td>
<td>700</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>9a</td>
<td>1050</td>
<td>40</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 24: I-10 freeway (traffic demand)

<table>
<thead>
<tr>
<th>Period</th>
<th>Origin</th>
<th>Start Time</th>
<th>Duration [min]</th>
<th>Flow Rate [pcph]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SRC1</td>
<td>12:00</td>
<td>30</td>
<td>3000</td>
</tr>
<tr>
<td>1</td>
<td>SRC7a</td>
<td>12:00</td>
<td>30</td>
<td>600</td>
</tr>
<tr>
<td>1</td>
<td>SRC15a</td>
<td>12:00</td>
<td>30</td>
<td>200</td>
</tr>
<tr>
<td>1</td>
<td>SRC19a</td>
<td>12:00</td>
<td>30</td>
<td>2000</td>
</tr>
<tr>
<td>2</td>
<td>SRC1</td>
<td>12:30</td>
<td>30</td>
<td>1500</td>
</tr>
<tr>
<td>2</td>
<td>SRC7a</td>
<td>12:30</td>
<td>30</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>SRC15a</td>
<td>12:30</td>
<td>30</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>SRC19a</td>
<td>12:30</td>
<td>30</td>
<td>1000</td>
</tr>
</tbody>
</table>
9.3.2 Simulation Results

To account for the stochastic behavior embedded in the microscopic parameters the simulation of each network was repeated 5 times with different random seeds for CTM and CORSIM and the average of the 5 runs was compared. The performance measures used for comparison are the cell (link) occupancy over time and the total network travel time by all vehicles. Several cells located upstream of the reduced-speed cell 18 were in analyzed in detail.

In Figure 79 the variation of the average occupancy of cell 12 and its corresponding link in CTM is plotted. The occupancy values represent 5 minute averages over the 5 runs in each of the simulated environment. It can be seen that the variation in cell occupancy follows closely the variation in the corresponding link in CORSIM.

![Figure 79: Comparison between CORSIM and CTM for cell 12](image-url)
Figure 80 plots the occupancy of the cell 14, closer to the bottleneck at the entrance of the reduced-speed cell 18. It appears that some congestion is captured by CTM because of the slight increase in occupancy during the first simulation period (12:00-12:30). Even though the corresponding link occupancy from CORSIM does not show an similar trend, the absolute relative difference does not exceed 11%. Similar with cell 12, during the reduced flow demand from the second simulation period the two models show a better agreement. Although the absolute relative during the light demand period varies between 1.9% and 69%, it can be attributed to the random variations in driving behavior.

Figure 80: Comparison between CORSIM and CTM for cell 14
In Figure 81 the combined occupancy of cells 16a and 16b is shown. These two cells represent a weaving area that includes an auxiliary lane. It can be seen that both CTM and CORSIM agree in showing some congested conditions. The relative absolute difference in occupancy over time ranges from 2% to nearly 33% and it is attributed to the random variations in driving behavior embedded in car-following and lane-changing models used in CORSIM.

Figure 81: Comparison between CORSIM and CTM for combined cells 16a and 16b
The average total network travel times for CTM and CORSIM were 319.74 and 299.86 vehicle hours, respectively. The 6.62% difference in travel time may be attributed to the random variations and lack of calibration of both models. Given a total of 4350 simulated vehicles, all passenger cars, the average error in travel time is nearly 16 seconds per vehicle for an average trip time of 248 seconds per vehicle. This clearly shows that, despite its mesoscopic nature, CTM provides very good representation of traffic flow when compared to microscopic simulation.

9.4 Summary

In this chapter, using a segment of I-10 freeway network, the performance of the modified CTM simulator was compared to that of CORSIM, a microscopic simulation platform. Link-by-link comparisons showed very similar results for both platforms over time, despite the difference in level of simulation and sources of stochastic variations. While microscopic in nature, variations in CORSIM results may be mostly attributed to random variations in driving behavior, as described by car-following and lane-changing models. On the other hand, variations in CTM results are attributed to random variations in the lane-changing and driving behavior, as well as the variation in the backward moving wave speeds during congestion. Regardless of source of variations, both platforms produced very comparable results, which reflect the potential of CTM for development and implementation as an operations model for large-scale networks.

Several strengths recommend using CTM instead of CORSIM or any other microscopic platform. For example, as a mesoscopic model CTM offers a less computational intensive environment, with virtually no restrictions on the numbers of
simulated entities. Therefore, it can be used in planning analysis and is more appropriate for area-wide networks.

Several studies demonstrated the potential of CTM in dynamic traffic assignment (DTA) applications. On the other hand, the existing DTA module in CORSIM is less flexible and the access to the embedded parameters is limited. Moreover, because of licensing issues, many microscopic model do not provide direct access to any embedded model (e.g. car-following, lane-changing, gap-acceptance, etc.), but only through the limited set of API functions.

In addition, CTM provides a more versatile simulation environment than microscopic models. It allows modeling at various fidelity levels through the change in cell size (i.e. small cells provide high fidelity and large cells low fidelity). Moreover, there is a great potential for using CTM in hybrid environments when freeways and intersections are modeled in one network, so that simulation updates may be allowed to vary.

Another advantage in CTM it comes from the demonstrated capabilities of modeling randomness in driving behavior. For a more realistic representation of driving behavior, microscopic models need complex calibration of car-following and lane-changing algorithms to account for various levels of congestion, safety, different driver types, effect of mixed vehicles. On the other hand, CTM can capture randomness in driving behavior at an aggregated level through dynamic variations in wave speeds.
CHAPTER 10  CONCLUSIONS AND FUTURE WORK

10.1 Summary

The cell transmission model (CTM), developed by Daganzo in 1994, provides a simple and physically relevant numerical solution to the LWR kinematic wave theory. Using finite difference approximation to solve the LWR differential equation of state, CTM predicts the temporal and spatial evolution of traffic flows by discretizing the network into cells and connectors. Movement of traffic streams between cells can be executed at either macroscopic or mesoscopic levels. While CTM is essentially derived from macroscopic flow-density models, it can be readily transformed into a mesoscopic model to recognize destinations of vehicles and provide better traffic flow representation. The open literature shows several research studies that deployed CTM to solve various traffic network problems such as dynamic traffic assignment, network design problems, travel time predictions, and others. Preliminary calibration efforts, reported by a few studies, indicated that CTM offers traffic flow representation that is sufficiently accurate for planning analysis purposes. Moreover, the macroscopic / mesoscopic characteristics of CTM offer computational and calibration advantages over microscopic traffic simulation models. Such advantages render CTM more appropriate for modeling large-scale traffic networks. However, in order to meet certain operational analysis needs of large-scale traffic networks, additional improvements and modifications are still required, and therefore, are the focus of this research study.
The primary goal of this study was to convert CTM from a planning model to an operations model, while retaining its macroscopic / mesoscopic features. To achieve this goal specific topological enhancements and operational improvements were introduced and examined through numerical analysis of network components. The topological enhancements included: (1) relaxation of the constant cell size requirements imposed by CTM and the subsequent adjustments to the flow advancing equations of ordinary, merging, and diverging cells; and (2) use of subcells to capture non-homogeneity of traffic conditions inside long cells. Several operational improvements included: (1) explicit treatment of traffic flows by lanes; (2) discretionary and mandatory lane changing maneuvers inside cells; (3) non-discrete movements of vehicles between cells; (4) more realistic representation of merging and diverging flows; (5) integrating random variations in driving behavior and non-linear dynamic changes in speeds of backward moving waves into the flow advancing equations; and (6) adjustments for modeling multi-modal traffic flows. The theoretical foundation for each improvement was presented and followed by experimental analysis to demonstrate and validate the effectiveness of such improvement. The study also conducted a comparative analysis between the improved version of CTM and another commercially available microscopic simulation model (CORSIM). All CTM improvements were implemented into a specially developed object-oriented simulation module, which also served as a platform for all computational work presented in this study.

10.2 Conclusions

This research study introduced significant topological and algorithmic improvements to the cell transmission model to address critical operational analysis needs of large-scale traffic networks. One of the topological constraints imposed by the original
form of CTM is the required use of constant cell size throughout the entire traffic network. Moreover, long, geometrically homogeneous cells may not be able to capture accurately the propagation of traffic queues within the cell. Consequently, the study introduced two topological improvements. The first improvement discretizes the cells into subcells in order to account for possible non-homogeneity of traffic conditions within long cells. This treatment is exclusively carried out by the simulation module at the pre-processing stage, and therefore, does not require the modeler’s intervention. In addition, this treatment does not lead to the same computational and memory overhead that would otherwise be caused by converting subcells into an equivalent number of cells. The second improvement includes mathematical modifications to the original flow advancing equations of CTM to permit selection of variable cell sizes.

Both improvements required separate processes to advance vehicles internally (within cells) and externally (between cells), as well as construction of a wait time function to track the time spent by vehicles in each cell. Experimental analysis showed negligible differences (mostly attributed to rounding errors) in simulation results, in terms of network travel time, cell occupancy, and average vehicular delay, using network topologies with variable vis-à-vis constant cell size. Testing and validating the topological improvements demonstrate that the constant cell size constraint can now be relaxed in order to increase the flexibility of CTM in operational analysis applications.

In addition to the topological improvements, the study targeted specific algorithmic improvements to the logic used to advance vehicles in ordinary, merging, and diverging segments of a traffic network. An intuitively essential modification to the vehicle advancing algorithms is to allow non-discrete movements of vehicles within and
between cells for more realistic and accurate representation of traffic flows. While advancing vehicles in real-number quantities is obviously a trivial task in macroscopic simulation, the implementation at the mesoscopic level is yet more challenging because of the special treatment required to ensure that all parts of the same vehicle follow the same path to the final destination. To overcome this implementation challenge, algorithmic modifications to the developed simulation module were made to ensure that all route choice decisions are exclusively made by the vehicle driver, who is always assumed to occupy the front part of the vehicle. All subsequent parts of the same vehicle must follow the path of the front part.

To examine the effect of vehicle movement precision on simulation results, experimental analysis was conducted on a simple freeway segment. The results showed a substantial difference in the total network travel time between discrete movements and the one-tenth movement precision. Differences attributed to higher precisions were small or negligible. This is because the size of subcells in the sample network was relatively larger than the length of a vehicle. However, for networks with much smaller subcell size, higher precisions may be deemed necessary to achieve the same level of accuracy. Such cases, however, are least common in practical applications. Therefore, the recommended movement precision in this study was set to one-tenth of a vehicle.

The study also made a significant improvement to CTM by introducing lane-changing capabilities to account for possible non-uniformity of lane use at various locations in the network. Examples of such locations can be found near or at merging and diverging junctions due to possible variations in traffic conditions along the merging or diverging streams. Two types of lane-changing maneuvers were identified and modeled in
CTM using a logit model. The first type represents discretionary lane-changing maneuvers, where drivers are assumed to choose the lane with the least disutility (measured by the relative lane occupancy). As such, this model allows vehicles to utilize the available travel lanes based on the prevailing traffic conditions in each lane, and therefore, leads to better and more uniform utilization of lanes.

The second type of lane-changing behavior involves mandatory maneuvers that are required to place vehicles on the lane leading to their destination. This is often seen near or at diverging junctions. Both lane-changing models were tested independently to examine their effect on the simulation results. The experimental analysis showed that disabling lane-changing maneuvers in CTM could adversely affect the utilization of lanes and lead to unrealistic representation of flows. For instance, lanes added at road widening locations are underutilized, while lanes within a diverging junction may be over-utilized. The effect of aggressiveness in driving behavior on lane-changing activities was also examined by adjusting the parameters used in the assumed disutility function. Such parameters are essentially probabilistic in nature, and therefore, could be derived from an assumed or observed probability distribution function. This, however, is beyond the scope of this study, and will be addressed in future research.

In this study, other algorithmic modifications to CTM were necessary to improve the merging and diverging logic. At merging junctions, the mechanism used to process vehicles from competing merging streams was revised to account for the presence of auxiliary lanes and various traffic conditions. Lane-changing behavior was also modeled at merging junctions to allow for more realistic representation of merging streams with different traffic conditions. Experimental analysis showed simulation results were
consistent with the expected real-life behavior. Nevertheless, calibration with real-world observations remains crucial to confirm adequacy of the proposed algorithm.

Treatment of flows at diverging junctions was also improved in this study to prevent some of the pathological cases of FIFO violation, reported earlier by other researchers. Such violation may occur when one of the diverging connectors gets blocked by congestion downstream the diverging cell. In such cases, the macroscopic treatment of traffic flow as a hydrodynamic fluid fails to capture the resulting non-homogeneous traffic conditions across the lanes in the diverging cell. To address this problem, this research study improved the diverging logic to account for lane alignment, possible partial blockage, and mandatory lane-changing behavior inside diverging cells. Experimental analysis was also conducted to show the behavior of vehicles inside diverging cells before and after the improvements were made. The simulation results showed consistency with what would be expected from real-life behavior. When one of the diverging connectors is blocked, only the lanes leading to that connector are affected, while the remaining lanes remain open for traffic to flow onto the other diverging connectors. Moreover, the results show that CTM no longer produces pathological cases of FIFO violation. Nevertheless, permissible cases of FIFO violation can still be observed as a result of modeling the lane-changing behavior.

Since the original form of CTM was presented in a deterministic context, the study attempted to account for the random driving behavior in the flow advancing equation of CTM in order to produce more accurate representation of traffic flows. Sources of randomness in the flow advancing equation may exist in one of its four degrees of freedom: free-flow speed, flow capacity, backward moving wave speed, and space capacity. Each of the four parameters can be assumed to follow some probability distribution function that
reflects the randomness in driving behavior. Moreover, the original equation of CTM assumes the backward-moving wave speed independent of traffic conditions. This assumption was validated in this study using vehicle trajectory data collected from I-80 and publicly available on the Next-Generation Simulation website. The calibration efforts clearly indicated that the backward-moving waves tend to travel faster as density increases in the receiving and sending cells. This suggests that the rate at which spaces can be utilized in a receiving cell dynamically depends on the occupancy of both sending and receiving cells at any simulation clock tick. Such finding indicates that the relationship between flow and density under forced-flow conditions is better captured with a nonlinear model. Consequently, the flow advancing equation in this study was modified to account for random variations in desired free-flow speeds, random variations in minimum headway, random variations in minimum spacing, and dynamic variations in the backward moving wave speed. The I-80 data was used to fit a set of third order polynomials that capture the dynamic variation in wave speeds under congested conditions. Experimental analysis was used to demonstrate the effect dynamic modeling of wave speed has on the simulation results. The results showed that the variation in wave speed leads to oscillatory variation in cell occupancies over time. Clearly, this observation is more consistent with real-life behavior of traffic flows.

Another shortcoming in the original formulation of CTM is the inability to model multi-modal flows. In this study, this limitation was overcome by adjusting the flow advancing equation to treat traffic flows with mixed types of vehicles without the need to convert demand into passenger car equivalents. The multi-modal adjustment was applied to the modified equations for random driving behavior to yield a more generalized flow
advancing equation. Experimental analysis was conducted to demonstrate how the
generalized flow advancing equation is capable of treating composite traffic streams of
passenger cars and trucks.

The last part of the study investigated the combined effect of all topological and
algorithmic improvements using a freeway segment of I-10 in downtown Baton Rouge.
The simulation results obtained from CTM were compared with those produced by
CORSIM, a microscopic simulation model. In lieu of calibration with real-life
observations, which is beyond the scope of this study, the evaluation was made to
determine qualitatively how the results from both simulation environments compare. The
simulation was executed for a period of two hours with mixed light to medium traffic
demand. A work zone was simulated by reducing the speed limit to create a bottleneck and
congested conditions. The simulation results showed that under both free-flow and
congested conditions the cell occupancies exhibited very similar profile over time, with
6.62% absolute relative error in the total network travel time. Discrepancies may be
attributed to differences in the level of simulation, network parameters, and treatment of
random driving behavior in both environments. The observed similarity in simulation
results from both models implies that CTM is as capable of capturing the main
characteristics and behavior of traffic flows at the mesoscopic level as CORSIM is at the
microscopic level. However, in addition to the calibration advantages, CTM is clearly
more versatile and computationally efficient than microscopic models in operational
analysis of large-scale traffic networks.
10.3 Future Work

Additional modifications need to be included in CTM to create more robust and flexible simulation platform. For example, to allow more accurate representation of traffic flows on hybrid networks of both interrupted flows (e.g. signalized intersections) and uninterrupted flows (freeways). It is also necessary to perform a comprehensive calibration of the random parameters. Integration of ITS components is a must for this simulation platform to be fully functional in traffic operations context.

Research is currently underway to expand the applicability of CTM to modeling pre-timed and actuated signalized intersections, as well as other network components such as toll plazas and unsignalized intersections.
REFERENCES


APPENDIX A\textsuperscript{1}: NGSIM SAMPLE DATASET

<table>
<thead>
<tr>
<th>Col 1</th>
<th>Col 2</th>
<th>Col 3</th>
<th>Col 4</th>
<th>Col 5</th>
<th>Col 6</th>
<th>Col 7</th>
<th>Col 8</th>
<th>Col 9</th>
<th>Col 10</th>
<th>Col 11</th>
<th>Col 12</th>
<th>Col 13</th>
<th>Col 14</th>
<th>Col 15</th>
<th>Col 16</th>
<th>Col 17</th>
<th>Col 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.884</td>
<td>48.213</td>
<td>6042842</td>
<td>2133118</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.931</td>
<td>49.463</td>
<td>6042842</td>
<td>2133119</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.991</td>
<td>50.712</td>
<td>6042842</td>
<td>2133120</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.045</td>
<td>51.963</td>
<td>6042842</td>
<td>2133122</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.098</td>
<td>53.213</td>
<td>6042842</td>
<td>2133123</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>17</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.151</td>
<td>54.463</td>
<td>6042842</td>
<td>2133124</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.49</td>
<td>-9.00E-02</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.204</td>
<td>55.712</td>
<td>6042842</td>
<td>2133125</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.48</td>
<td>-8.00E-02</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.257</td>
<td>56.956</td>
<td>6042842</td>
<td>2133127</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.52</td>
<td>0.55</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.33</td>
<td>58.199</td>
<td>6042842</td>
<td>2133128</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.67</td>
<td>2.21</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>21</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.289</td>
<td>59.463</td>
<td>6042841</td>
<td>2133129</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13</td>
<td>4.43</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>22</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.197</td>
<td>60.776</td>
<td>6042841</td>
<td>2133130</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13.49</td>
<td>5.64</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.027</td>
<td>62.157</td>
<td>6042841</td>
<td>2133132</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13.98</td>
<td>4.77</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>24</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.863</td>
<td>63.592</td>
<td>6042840</td>
<td>2133133</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.36</td>
<td>2.84</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.752</td>
<td>65.054</td>
<td>6042840</td>
<td>2133134</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.58</td>
<td>1.17</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>26</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.731</td>
<td>66.526</td>
<td>6042840</td>
<td>2133136</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.64</td>
<td>8.00E-02</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>27</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.682</td>
<td>67.993</td>
<td>6042840</td>
<td>2133137</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.62</td>
<td>-0.4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.632</td>
<td>69.455</td>
<td>6042839</td>
<td>2133139</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.59</td>
<td>-0.54</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>29</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.583</td>
<td>70.913</td>
<td>6042839</td>
<td>2133140</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.52</td>
<td>-0.69</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.528</td>
<td>72.367</td>
<td>6042839</td>
<td>2133142</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.38</td>
<td>-1.71</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.478</td>
<td>73.809</td>
<td>6042839</td>
<td>2133143</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>14.13</td>
<td>-3.49</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.453</td>
<td>75.21</td>
<td>6042839</td>
<td>2133144</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13.75</td>
<td>-4.36</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>33</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.462</td>
<td>76.559</td>
<td>6042838</td>
<td>2133146</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13.37</td>
<td>-3.51</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>34</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.495</td>
<td>77.867</td>
<td>6042838</td>
<td>2133147</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13.11</td>
<td>-1.72</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>35</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.534</td>
<td>79.159</td>
<td>6042838</td>
<td>2133148</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13</td>
<td>-0.3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>36</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.567</td>
<td>80.454</td>
<td>6042838</td>
<td>2133150</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>12.98</td>
<td>0.16</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>37</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.601</td>
<td>81.754</td>
<td>6042838</td>
<td>2133151</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
<td>13</td>
<td>0.07</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\textsuperscript{1} Definitions of the columns are detailed in Table 1
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>70</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.312</td>
<td>117.17</td>
<td>6042834</td>
<td>2133186</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>71</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.296</td>
<td>117.537</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>72</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.28</td>
<td>117.903</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>73</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.263</td>
<td>118.27</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>74</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.247</td>
<td>118.637</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>75</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.231</td>
<td>119.004</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>76</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.214</td>
<td>119.37</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>77</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.199</td>
<td>119.737</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>78</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.182</td>
<td>120.104</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>79</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.165</td>
<td>120.47</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.15</td>
<td>120.837</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>81</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.133</td>
<td>121.204</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>82</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.117</td>
<td>121.57</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>83</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.101</td>
<td>121.937</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>84</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.085</td>
<td>122.304</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>85</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.068</td>
<td>122.67</td>
<td>6042834</td>
<td>2133187</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>86</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.052</td>
<td>123.037</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>87</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.036</td>
<td>123.403</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>88</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.019</td>
<td>123.772</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>89</td>
<td>884</td>
<td>1.11E+12</td>
<td>17.004</td>
<td>124.143</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>90</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.986</td>
<td>124.509</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>91</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.97</td>
<td>124.854</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>92</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.959</td>
<td>125.18</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>93</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.954</td>
<td>125.515</td>
<td>6042833</td>
<td>2133192</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>94</td>
<td>884</td>
<td>1.11E+12</td>
<td>16.954</td>
<td>125.893</td>
<td>6042833</td>
<td>2133195</td>
<td>14.3</td>
<td>6.4</td>
<td>2</td>
</tr>
</tbody>
</table>
APPENDIX B: MODULAR STRUCTURE OF CTM
APPENDIX C: SELECTED SOURCE CODE FOR CTM

'--------------------------------------------------------------------------------------------------------------------------
'Scope RUNTIME, called for each simulation update
'This subroutine applies the flow advancing equation for all links that do not
'belong to an intersection; the function builds a temporary collection to store the cells
'and connectors and it calls specific functions for each type of cell/connector to be
'processed
'--------------------------------------------------------------------------------------------------------------------------

Public Sub RunStageOneNonIntersections(ByVal TmpConnectors As Collection)
'flow advancing procedure
  Dim LocalTmpConnectors As New Collection
  Utils.CopyCollection(TmpConnectors, LocalTmpConnectors)

  Do While LocalTmpConnectors.Count > 0
    Dim crtConnector As cConnector = LocalTmpConnectors(1)
    Select Case CType(crtConnector, cConnector).Type
      Case ConnType.SOURCE, ConnType.SINK, ConnType.ORDINARY
        Me.Stage1(crtConnector, Me.Tau)
        LocalTmpConnectors.Remove(crtConnector.ID)
      Case ConnType.MERGE
        Me.Stage1(crtConnector, Me.Tau)
        For Each connector As cMConnector In CType(crtConnector, cMConnector).ToCell.Predecessors
          LocalTmpConnectors.Remove(connector.ID)
        Next
      Case ConnType.DIVERGE
        Me.Stage1(crtConnector, Me.Tau)
        For Each connector As cDConnector In CType(crtConnector, cDConnector).FromCell.Successors
          LocalTmpConnectors.Remove(connector.ID)
        Next
    End Select
  Loop
End Sub     'End RunStageOneNonIntersections
Public Sub RunStageTwoNonIntersections(ByVal TmpConnectors As Collection)

    'advancing vehicles from cells to connectors
    Dim LocalTmpConnectors As New Collection
    Utils.CopyCollection(TmpConnectors, LocalTmpConnectors)

    Do While LocalTmpConnectors.Count > 0
        Dim crtConnector As cConnector = LocalTmpConnectors(1)
        Select Case crtConnector.Type
            Case ConnType.SOURCE, ConnType.ORDINARY, ConnType.SINK
                Me.Stage2(crtConnector, Me.Tau)
                LocalTmpConnectors.Remove(crtConnector.ID)
            Case ConnType.MERGE
                Me.Stage2(crtConnector, Me.Tau)
                For Each connector As cMConnector In CType(crtConnector, cMConnector).ToCell.Predecessors
                    LocalTmpConnectors.Remove(connector.ID)
                Next
            Case ConnType.DIVERGE
                Me.Stage2(crtConnector, Me.Tau)
                For Each connector As cDConnector In CType(crtConnector, cDConnector).FromCell.Successors
                    LocalTmpConnectors.Remove(connector.ID)
                Next
        End Select
    Loop
End Sub 'End RunStageTwoNonIntersections
Public Sub RunStageThreeNonIntersections(ByVal TmpCells As Collection)
    'update wait time of vehicles in ordinary cells
    For i As Integer = 1 To TmpCells.Count
        Dim cell As cCell = TmpCells(i)
        Me.Stage3(cell, Me.Tau)
    Next i
End Sub 'End RunStageThree

Public Sub RunStageFourNonIntersections(ByVal TmpCells As Collection)
    'advancing flow from connectors into cells
    'source cells are excluded
    Dim t1 As DateTime
    Dim t2 As DateTime
    For Each Cell As cCell In TmpCells
        'Me.Stage4(Cell, Me.Tau)
        If Cell.Type <> CellType.SINK Then
            'MOEs
            Utils.CollecMOEs(Me, (MOETypes.NetworkTT And Me.MOEMap), Cell, , Me.Tau)
        End If
        'collect vehicle trajecotry, if requested
        Utils.CollecMOEs(Me, (MOETypes.VehiclesTraject And Me.MOEMap), Cell, , Me.Tau)
    Next Cell
    'MOEs
    Utils.CollecMOEs(Me, ((MOETypes.CellsTT Or MOETypes.CellsX Or MOETypes.CellsOccupancy) And Me.MOEMap))
End Sub 'End RunStageFour
Public Sub Stage1(ByRef crtConnector As cConnector, ByVal networkTau As Integer)', 'Optional ByVal SignalStatus As Boolean = True)', Optional ByVal NGaps As single = single.MaxValue)
Dim max2Send As Integer = 0
Dim max2Receive As Integer = 0
Dim Delta As Single = 1
Dim SourceCell As cCell = (crtConnector.FromCell)
Dim TargetCell As cCell = crtConnector.ToCell
If (TypeOf crtConnector Is cMConnector) Then 'TREAT MERGE
  Dim mCell As cMCell = crtConnector.ToCell
  mCell.ComputeAndSetRearXr(Me.Tau, Me)
  Dim mConnectors As New ArrayList
  mConnectors.AddRange(mCell.Predecessors)
  mConnectors.Sort() 'sort the predecesors ascending by priority
  "set the max Priority Connector for the cell
  "done at the begining of each update to allowe for priorities to be flexible during simulation
  mCell.MaxPriorityConnector = mConnectors(mConnectors.Count - 1)
  For Each mConnector As cMConnector In mConnectors
    mConnector.FromCell.ComputeAndSetXs(networkTau)
    mConnector.Y = Min(mConnector.FromCell.Xs, Int(mConnector.FromCell.Qmax * networkTau))
    "
    mConnector.YFront = 0
    max2Send = max2Send + MConnector.Y
  Next
ElseIf (TypeOf crtConnector Is cDConnector) Then 'TREAT DIVERGE
  Dim dCell As cDCell = crtConnector.FromCell
  dCell.ComputeAndSetXs(Me.Tau)
  For i As Integer = 0 To dCell.Successors.Count - 1
Dim dConnector As cDConnector = dCell.Successors(i + 1)
'Dim lanes As Integer = dConnector.IncomingLanes.Count
Dim destCell As cCell = dConnector.ToCell

destCell.ComputeAndSetRearXr(Me.Tau, Me)

'What is the maximum that can be received by the successor cell of this connector?
'Answer: the sum of the first subcells in each lane
max2Send = dCell.GetXsByConnector(networkTau, Me, i)
max2Send = Max(max2Send, dCell.Qmax * networkTau)

If max2Send > destCell.Qmax * networkTau Then
    'Delta = destCell.WaveSpeed / destCell.FFS
    Delta = destCell.ComputeAndSetDelta(dCell, networkTau)
End If

max2Receive = Int(Min(destCell.Qmax * networkTau, Delta * destCell.RearXr))
dConnector.Y = Int(Min(max2Send, max2Receive))
Next i
Else 'TREAT ORDINARY, SOURCE, or SINK
    'sets Xs, Xr
    SourceCell.ComputeAndSetXs(Me.Tau)
    If SourceCell.Xs > 0 Then
        TargetCell.ComputeAndSetRearXr(Me.Tau, Me)
        If crtConnector.Type <> ConnType.SINK And crtConnector.Type <> ConnType.SOURCE Then
            If SourceCell.Xs > Int(TargetCell.Qmax * networkTau) Then
                Delta = TargetCell.WaveSpeed / TargetCell.FFS
                'Delta = TargetCell.ComputeAndSetDelta(SourceCell, networkTau)
            End If
        End If
    End If
    crtConnector.Y = Min(Int(TargetCell.Qmax * networkTau), Min(SourceCell.Xs, Int(Delta * TargetCell.RearXr)))
Else
crtConnector.Y = 0
End If
crtConnector.YFront = 0

'below is stage 1 for internal connectors of the sending cell (cells if merging)
'if is not a source cell
For Each predConn As cConnector In TargetCell.Predecessors
    SourceCell = predConn.FromCell
    If SourceCell.Type = CellType.SOURCE Then
Exit Sub
End If

'Apply stage 1
'to the front subcell

Dim DeltaN_i As Single = 1
If SourceCell.SubX(SourceCell.NSubCells - 2) > Int(SourceCell.Qmax * networkTau) Then
End If

Dim Xr As Integer = Int(DeltaN_i * (SourceCell.FrontSubXmax - SourceCell.SubX(SourceCell.NSubCells - 1)) + (1 - SourceCell.FrontTau / networkTau) * Min(TargetCell.Qmax * networkTau, Delta * (TargetCell.SubXmax - TargetCell.SubX(0))))
SourceCell.SubY(SourceCell.NSubCells - 2) = Min(Min(SourceCell.SubX(SourceCell.NSubCells - 2), Int(SourceCell.Qmax * networkTau)), Xr)
SourceCell.SubY(SourceCell.NSubCells - 2) = Max(0, SourceCell.SubY(SourceCell.NSubCells - 2) - (predConn.Y - Min(predConn.Y, SourceCell.SubX(SourceCell.NSubCells - 1))))

'apply stage 1
'to the remaining subcells
For subcell As Integer = 0 To SourceCell.NSubCells - 3
    Dim Xs As Integer = SourceCell.SubX(subcell) '* (UpdateTime / Me.SubTau)
    If Xs > 0 Then
        Dim Qmax As Integer = Int(SourceCell.Qmax * networkTau)
        Delta = 1
        If SourceCell.SubX(subcell) > Qmax Then
            Delta = Utils.Poly3(Xs / SourceCell.SubXmax, SourceCell.SubX(subcell + 1) / SourceCell.SubXmax)
        End If
        Xr = SourceCell.SubXmax - SourceCell.SubX(subcell + 1)
        Xr = Int(Xr * (networkTau / Me.SubTau) * Delta)
        SourceCell.SubY(subcell) = Min(Min(Xs, Qmax), Xr)
    Else
        'nothing to advance from this subcell, reset subY to zero
        SourceCell.SubY(subcell) = 0
    End If
Next subcell
Next predConn
End Sub  'End Stage1(FlowAdvancingEquation)
Public Sub Stage2(ByRef crtConnector As cConnector, ByVal UpdateTime As Integer)
    Dim Cell As cCell = crtConnector.ToCell
    'SubXr - needed for external LC algorithm to store
    'the existing space capacity in the last subcell of the receiving cell
    'this is continuously updated as the vehicles proceed out of the sending cell

    If crtConnector.Type = ConnType.MERGE Then
        Stage2_Merge(crtConnector, UpdateTime)
    ElseIf crtConnector.Type = ConnType.DIVERGE Then
        Stage2_Diverge(crtConnector, UpdateTime)
    Else
        Stage2_Ordinary(crtConnector, UpdateTime)
    End If
End Sub   'End Stage2(AdvanceFromCell2Connector)
Sub Stage2_Ordinary(ByRef ijConnector As cConnector, ByVal NetworkTau As Integer)
    Utilst1 = DateTime.Now()
    Dim Max2Advance As Long = ijConnector.Y
    If Max2Advance = 0 Then Exit Sub
    'Monitor how many advanced out
    ijConnector.Y = 0
    'Sending cell is icell and receiving cell is jcell
    Dim CC As cCell = ijConnector.FromCell
    Dim NC As cCell = ijConnector.ToCell
    Dim NCNLanes As Integer = NC.NLanes
    Dim CCNLanes As Integer = CC.NLanes
    'Monitor how many have advanced in by lane
    Dim Yreceived As Long() = Array.CreateInstance(GetType(Long), NCNLanes)
    'Stochastic variations in free-flow speeds
    'Create collections of Platoon, PlatoonX, and PlatoonCS for each lane in the next subcell
    Dim Platoon As cPlatoon = New cPlatoon(NCNLanes)
    Dim NS As Integer = 0
    Dim CS As Integer = 0
    Dim idx As Integer = 1
    Dim Y2move As Long = 0
    Dim NL As Integer = -1
    Dim NCurrentLanesOpen As Integer = CC.NLanes
    Dim NNextLanesOpen As Integer = NC.NLanes
    Dim NLOpen As Boolean() = Array.CreateInstance(GetType(Boolean), NCNLanes)
    Dim CLOpen As Boolean() = Array.CreateInstance(GetType(Boolean), CCNLanes)
    Dim MinHout As Integer() = Array.CreateInstance(GetType(Integer), CCNLanes)
    Dim MinHin As Integer() = Array.CreateInstance(GetType(Integer), NCNLanes)
    Dim Q As New Collection
    Dim QX As New Collection
    idx = 1
    For CS = CC.NSubCells - 1 To CC.NSubCells - 2 Step -1
        While idx <= CC.VQ(CS).Count
            Dim Part As cXCSL = CC.VQ(CS)(idx)
            Try
                Q.Add(Part, Part.V.ID)
QX.Add(Part.X, Part.V.ID)
Catch ex As ArgumentException 'update the Qx
    Dim tX As Long = QX(Part.V.ID)
    QX.Remove(Part.V.ID)
    QX.Add(tX + Part.X, Part.V.ID)
End Try
idx += 1
End While
idx = 1
Next

For i As Integer = 0 To NCNLanes - 1
    NLOpen(i) = True
Next
For i As Integer = 0 To CCNLanes - 1
    CLOpen(i) = True
Next

'reset index to 1 for collection-type index
idx = 1
Dim NSidx As Integer = 1
Dim CSIdx As Integer() = Array.CreateInstance(GetType(Integer), 2)
CSIdx(0) = 1
CSIdx(1) = 1
Utils.t2 = DateTime.Now
'Utils.stage4RunTime += Utils.t2.Subtract(Utils.t1).TotalMilliseconds

Do While idx <= Q.Count And NCurrentLanesOpen > 0 And NNextLanesOpen > 0 And ijConnector.Y < Max2Advance
    Dim CPart As cXCSL = Q(idx)
    Dim NPart As cXCSL = Nothing
    Dim CPartID As Integer = CPart.ID
    Dim Vehicle As cVehicle = CPart.V
    Dim VehID As String = Vehicle.ID
    Dim Nparts As Integer = Vehicle.Parts.Count
    Dim VehX As Long = QX(VehID)
    Dim CL As Integer = CPart.L
    Dim CS = CPart.S
    Y2move = 0
    If CLOpen(CL) Then
        Y2move = Min(VehX, Max2Advance - ijConnector.Y)
        'is this vehicle eligible to advance basec in the minimum headway remaining in current lane?
Y2move = Min(Y2move, Int((CC.SubTau - MinHout(CL)) * Vehicle.Length / Vehicle.MinH))
End If
If Y2move > 0 Then
  'Is this vehicle eligible to advance based on its minimum free-flow travel time in
  the current subcell?
  If CPartID = 0 Then
    'This is a front part that we need to check its wait time against the minimum
    wait time
    VCheckMinWait(Vehicle, CC, CS, Y2move, "E")
    If Y2move > 0 Then
      'NL = use mandatory LC or discretionary LC based on cell type
      NL = ComputeNextLaneExternal(Y2move, Vehicle, CS, CL, CC, NC, ijConnector.ID, NetworkTau)
      If NL < 0 Then
        Y2move = 0
      ElseIf Not NLOpen(NL) Then
        Y2move = 0
      End If
      'Ensure that the next lane does not overfill in the LC algorithm
    End If
  Else
    'This is not a front part and is not subject to wait time check
    NPart = Vehicle.Parts(CPartID - 1)
    NL = NPart.L
  End If
If Y2move > 0 Then
  'Is this vehicle eligible to advance based on the minimum headway remaining in
  next lane?
  Y2move = Min(Y2move, Int((CC.SubTau - MinHin(NL)) * Vehicle.Length / Vehicle.MinH))
  'Is this vehicle eligible to advance based on the minimum spacing remaining in
  next lane?
  Y2move = Min(Y2move, NC.Xr(NS, NL) * Vehicle.Length / Vehicle.MinS)
End If
End If
If Y2move > 0 Then
  Dim PPart As cXCSL
  Dim CPartX As Long = 0
  Dim PPartX As Long = 0
  Dim PPartID As Integer = Min(CPartID + 1, Nparts - 1)
  Dim MPart As cXCSL
  Dim LastSubcellID As Integer = CC.NSubCells - 1
  If CS = LastSubcellID Then
    CPartX = Min(CPart.X, Y2move)
  End If
PPartX = Y2move - CPartX
'Remove the disutility from CS, CL
CC.Disutility(LastSubcellID, CL) -= CPartX * (1 - CPartID / Nparts)
'Remove the disutility from CS-1, CL
CC.Disutility(LastSubcellID - 1, CL) -= PPartX * (1 - PPartID / Nparts)

If CS = LastSubcellID - 1 Or CPartID = Nparts - 1 Then
'Vehicle part exists in one of the two subcells
If VehX = Vehicle.Length Then
'Vehicle is a whole vehicle in CS
If Y2move = VehX Then
'Advance the whole vehicle
'Remove it from VQ of CS
CC.VQ(CS).Remove(VehID)
'update vehicle part coordinates
CPart.C = NC
CPart.S = NS
CPart.L = NL
Else
'Advance a part of this whole vehicle
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS).Add(CPart, VehID, CSIdx(LastSubcellID - CS))
CSIdx(LastSubcellID - CS) += 1
idx += 1
If CS < LastSubcellID Then   'And CC.FrontTau > 0 And
CType(Vehicle.Parts(CPart.ID), cXCSL).C.ID <> CC.ID Then
'Create a Zero part in the last subcell to ensure processing the
remaining parts first
VSplit(0, CPart, CC, LastSubcellID, CL)
Dim ZPart As cXCSL = Vehicle.Parts(CPart.ID - 1)
CC.VQ(LastSubcellID).Add(ZPart, VehID, CSIdx(0))
CSIdx(0) += 1
End If
VBlockLanes(CC, NC, ijConnector, CL, NL, NLOpen, NNextLanesOpen)
End If
ElseIf CPart.ID < Nparts - 1 Then
'This is not the last part in CS; it must be in the next-to-last subcell
'We must split it anyway; a zero part may be created if Y2move = VehX
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS).Remove(VehID)
CC.VQ(CS).Add(CPart, VehID, CSIdx(LastSubcellID - CS))
CSIdx(1) += 1
If CS < LastSubcellID Then
'Create a Zero part in the last subcell to ensure processing the remaining parts first

VSplit(0, CPart, CC, LastSubcellID, CL)
Dim ZPart As cXCSL = Vehicle.Parts(CPart.ID - 1)
CC.VQ(LastSubcellID).Add(ZPart, VehID, CSIdx(0))
CSIdx(0) += 1
End If
idx += 1
VBlockLanes(CC, NC, ijConnector, CL, NL, NLOpen, NNExtLanesOpen)
Else
'This is a last part, it must be in the last subcell
'Split it only if we cannot advance the whole part
If Y2move < VehX Then
'Split Y2move out of VehX
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS).Remove(VehID)
CC.VQ(CS).Add(CPart, VehID, CSIdx(LastSubcellID - CS))
CSIdx(LastSubcellID - CS) += 1
idx += 1
VBlockLanes(CC, NC, ijConnector, CL, NL, NLOpen, NNExtLanesOpen)
Else
'The last part is advancing out in full
'Remove it from VQ of CS
CC.VQ(CS).Remove(VehID)
End If
End If
Else
'Vehicle part spans both subcells
PPart = Vehicle.Parts(PPartID)
If Y2move <= CPartX Then
'Advance out of the last subcell ONLY; we must split Y2move out of the part in last subcell
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS).Remove(VehID)
CC.VQ(CS).Add(CPart, VehID, CSIdx(0))
CSIdx(0) += 1
idx += 1
VBlockLanes(CC, NC, ijConnector, CL, NL, NLOpen, NNExtLanesOpen)
Else
'Advance out of both subcells
If Y2move = Vehicle.Length Then
'A whole vehicle is advancing out of both subcells; no need to split, just
join
VJoin(PPart, CPart)
'update vehicle part coordinates
CPart.C = NC
CPart.S = NS
CPart.L = NL
'Remove both parts from VQ
CC.VQ(CS).Remove(VehID)
CC.VQ(LastSubcellID - 1).Remove(VehID)
Else
'This is not a whole vehicle
'is either a last part but not all part is advancing(Y2move<Vehx)
'or a middle part advancing in full(need to leave something in place
If Y2move < VehX Or PPartID < Nparts - 1 Then
'First, split the part in next-to-last subcell
'Take ?Y2move - CPartX? out of PPartX
VSplit(PPartX, PPart, NC, NS, NL)
MPart = Vehicle.Parts(PPartID)
'Then, join
VJoin(MPart, CPart)
'Place the remaining part CPart in front of VQ of the next to last subcell
CC.VQ(LastSubcellID - 1).Remove(VehID)
CC.VQ(LastSubcellID - 1).Add(PPart, Vehicle.ID, CSIdx(1))
CSIdx(1) += 1
'Create a Zero part in the last subcell
VSplit(CPart.X, CPart, CC, LastSubcellID, CL)
idx += 1
If CPartID = 0 Then
'update vehicle part coordinates
MPart = Vehicle.Parts(CPartID)
MPart.C = NC
MPart.S = NS
MPart.L = NL
End If
VBlockLanes(CC, NC, ijConnector, CL, NL, NLOpen, NNExtLanesOpen)
Else
'The last part is advancing out in full
'Join PPart and CPart
VJoin(PPart, CPart)
'Remove this vehicle from last subcell
CC.VQ(CS).Remove(VehID)
'Remove it also from the previous subcell
CC.VQ(LastSubcellID - 1).Remove(VehID)
End If
End If
End If
End If
'update subX, Xr
CC.SubX(LastSubcellID) -= CPartX
CC.SubX(LastSubcellID - 1) -= PPartX
If NC.Type <> CellType.SINK Then
    NC.Xr(NS, NL) -= CLng(Y2move * Vehicle.MinS / Vehicle.Length)
End If
CC.Xr(LastSubcellID, CL) += CLng(CPartX * Vehicle.MinS / Vehicle.Length)
CC.Xr(LastSubcellID - 1, CL) += CLng(PPartX * Vehicle.MinS / Vehicle.Length)
ijConnector.Y += Y2move
ijConnector.YFront += Min(CPartX, Y2move)
'How much is remaining in the last subcell?
Dim FrontY2move As Long = CPartX 'store before you loose it
CPartX = Max((CPartX - Y2move), 0)
'How much is remaining in the previous subcell?
PPartX = Min(PPartX, VehX - Y2move)
CC.Disutility(LastSubcellID - 1, CL) += PPartX * (1 - PPartID / Vehicle.Parts.Count)
CC.Disutility(LastSubcellID, CL) += CPartX * (1 - CPartID / Vehicle.Parts.Count)
MPart = Vehicle.Parts(CPartID)
NC.Disutility(NS, NL) += Y2move * (1 - MPart.ID / Vehicle.Parts.Count)
'Place this vehicle in the buffer of next cell
NC.PartBuffer.Add(MPart)
'Remove this vehicle from Queue, Parts, and QueueX
Q.Remove(VehID)
QX.Remove(VehID)
NC.SubX(NS) += Y2move
Yreceived(NL) += Y2move
If CPartID = 0 Then
    'Update platoons for a front part
    Platoon.Add(NL, Vehicle, CC, CS, Yreceived(NL))
    Vehicle.EntryTick = Vehicle.ExitTick
    Vehicle.ExitTick = Me.Tick
End If
MinHout(CL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
'check if the current lane should be blocked based on flow capacity
If CC.SubTau = MinHout(CL) Then
    'block the current lane
    CLOpen(CL) = False
    NCurrentLanesOpen -= 1
End If
MinHin(NL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
'check if the next lane should be blocked based on flow capacity
If CC.SubTau = MinHin(NL) And NLOpen(NL) Then
  'block the next lane
  VBlockLanes(CC, NC, ijConnector, CL, NL, NLOpen, NNextLanesOpen)
End If
If NC.Type = CellType.DIVERGE Then
  Dim i As Integer = CType(Vehicle.TargetConnector(NC.ID), cConnector).orderID
  CType(NC, cDCell).yInByDest(i) += Y2move
  CType(NC, cDCell).yInFrontByDest(i) += FrontY2move
End If
Else
  'We could not advance this vehicle because of lane changing
  idx += 1
End If
Loop
'Adjust the vehicle wait times in the next subcell by lane
VAdjustNSWaitTimes(NC.NLanes, CC, NetworkTau, Platoon, Yreceived, "E")
'Adjust the vehicle wait times of front parts remaining in the last subcell of current cell
ONLY
  'The one next-to-last will be taken care of in internal update
  VAdjustCSWaitTimes(CC, CC.NSubCells - 1, NetworkTau)
End Sub
Sub Stage2_Merge(ByRef ijConnector As cMConnector, ByVal NetworkTau As Integer)

'Sending cell is icell and receiving cell is jcell
Dim CC As cCell
Dim NC As cCell = ijConnector.ToCell
Dim NCNLanes As Integer = NC.NLanes
Dim Nk As Integer = NC.Predecessors.Count

'Monitor how many have advanced in by lane
Dim Yreceived As Long() = Array.CreateInstance(GetType(Long), NCNLanes)
'Stochastic variations in free-flow speeds
'Create collections of Platoon, PlatoonX, and PlatoonCS for each lane in the next subcell
Dim Platoon As cPlatoon = New cPlatoon(NCNLanes)

Dim NS As Integer = 0
Dim Y2move As Long
Dim NL As Integer
Dim NOpenLanes As Integer
Dim mConnector As cMConnector
Dim Total2Send As Long = 0
Dim Xs2Send As Long = 0
Dim Q2Send As Single = 0
Dim TotalReceived As Long = 0
Dim NTargetLanesOpen As Integer = NC.NLanes
Dim Max2Send() As Long = Array.CreateInstance(GetType(Long), Nk)
Dim Q As Collection() = Array.CreateInstance(GetType(Collection), Nk)
Dim QX As Collection() = Array.CreateInstance(GetType(Collection), Nk)
Dim Y() As Long = Array.CreateInstance(GetType(Long), Nk)
Dim CS() As Long = Array.CreateInstance(GetType(Long), Nk)
Dim CCOpen() As Boolean = Array.CreateInstance(GetType(Boolean), Nk)
Dim idx As Integer() = Array.CreateInstance(GetType(Integer), Nk)
Dim CCells As cCell() = Array.CreateInstance(GetType(cCell), Nk)
Dim CSIdx As Integer(,) = Array.CreateInstance(GetType(Integer), Nk, 2)
Dim MinHout As Array() = Array.CreateInstance(GetType(Integer), Nk)
Dim MinHin As Integer() = Array.CreateInstance(GetType(Integer), NCNLanes)
Dim CLOpen As Array() = Array.CreateInstance(GetType(Array), Nk)
Dim NCurrentLanesOpen As Integer() = Array.CreateInstance(GetType(Integer), Nk)
Dim NNextLanesOpen As Integer = NCNLanes

Dim NLOpen As Boolean() = Array.CreateInstance(GetType(Boolean), NCNLanes)
Dim YreceivedByLane As Long() = Array.CreateInstance(GetType(Long), NCNLanes)

For i As Integer = 0 To Nk - 1
    Dim SendingCell As cCell = CType(NC.Predecessors(i + 1), cConnector).FromCell
    Dim CCNLanes As Integer = SendingCell.NLanes
    mConnector = CType(NC.Predecessors(i + 1), cMConnector)
    CCells(i) = SendingCell
    Max2Send(i) = mConnector.Y
    CCOpen(i) = True
    mConnector.Y = 0
    Total2Send += Max2Send(i)
    CS(i) = SendingCell.NSubCells - 1
    CSIdx(i, 0) = 1
    CSIdx(i, 1) = 1
    Xs2Send += SendingCell.Xs
    Q2Send += SendingCell.Qmax
    NOpenLanes += CCNLanes
    NShared(i) = SendingCell.NsharedLanes
    idx(i) = 1
    Q(i) = New Collection
    QX(i) = New Collection
    MinHout(i) = Array.CreateInstance(GetType(Integer), CCNLanes)
    CLOpen(i) = Array.CreateInstance(GetType(Integer), CCNLanes)
    NCurrentLanesOpen(i) = CCNLanes

    For S As Integer = SendingCell.NSubCells - 1 To SendingCell.NSubCells - 2 Step -1
        While idx(i) <= SendingCell.VQ(S).Count
            Dim Part As cXCSL = SendingCell.VQ(S)(idx(i))
            Try
                Q(i).Add(Part, Part.V.ID)
                QX(i).Add(Part.X, Part.V.ID)
                Catch ex As ArgumentException 'update the Qx
                    Dim tX As Long = QX(i)(Part.V.ID)
                    QX(i).Remove(Part.V.ID)
                    QX(i).Add(tX + Part.X, Part.V.ID)
            End Try
            idx(i) += 1
        End While
        idx(i) = 1
    End For
End For
186
Next S
Next i
If Total2Send <= 0 Then Exit Sub

Dim NOpenLanesMax As Integer = NOpenLanes

If Xs2Send > Min(Q2Send, NC.Qmax) * NetworkTau Then
    NC.Delta = NC.WaveSpeed / NC.FFS
Else
    NC.Delta = 1
End If

Dim Max2Receive As Long = Min(NetworkTau * NC.Qmax, NC.Delta * NC.RearXr)

Dim CongestedMerging As Boolean = False
If Total2Send > Max2Receive Then
    CongestedMerging = True
End If

'We move vehicles simultaneously from the predecessors into the first subcell in targetcell

For i As Integer = 0 To NCNLanes - 1
    NLOpen(i) = True
Next

'reset index to 1 for collection-type index
Dim NSidx As Integer = 1
Dim CCIdx As Integer = 0
Do 'While idx <= Q.Count And NOpenLanes > 0 And mConnector.Y < Max2Advance
    If CongestedMerging Then
        CCIdx = GetSourceCellCongested(NC, NShared, Max2Send, CCOpen, Nk)
    Else
        CCIdx = GetSourceCellUncongested(NC, NShared, Max2Send, CCOpen, Nk)
    End If

    If CCIdx = -1 Then
        'all source cells are closed
        Exit Do
    End If

    CC = CCells(CCIdx)
    mConnector = CType(CC.Successors(1), cMConnector)

    Dim CPart As cXCSL = Q(CCIdx)(idx(CCIdx))
    Dim NPart As cXCSL = Nothing
Dim CPartID As Integer = CPart.ID
Dim Vehicle As cVehicle = CPart.V
Dim VehID As String = Vehicle.ID
Dim Nparts As Integer = Vehicle.Parts.Count
Dim VehX As Long = QX(CCIdx)(VehID)
Dim CL As Integer = CPart.L
CS(CCIdx) = CPart.S
Y2move = Min(VehX, Max2Send(CCIdx) - mConnector.Y)
'Is this vehicle eligible to advance based on the minimum headway remaining in current lane?
Y2move = Min(Y2move, (CC.SubTau - MinHout(CCIdx)(CL)) * Vehicle.Length / Vehicle.MinH)
If Y2move > 0 Then
  'Is this vehicle eligible to advance based on its minimum free-flow travel time in the current subcell?
  If CPartID = 0 Then
    'This is a front part that we need to check its wait time against the minimum wait time
    VCheckMinWait(Vehicle, CC, CS(CCIdx), Y2move, "E")
    If Y2move > 0 Then
      'NL = use mandatory LC or discretionary LC based on cell type
      NL = ComputeNextLaneExternal(Y2move, Vehicle, CS(CCIdx), CL, CC, NC, mConnector.ID, NetworkTau)
      If NL < 0 Then
        Y2move = 0
      ElseIf Not NLOpen(NL) Then
        Y2move = 0
      End If
      'Ensure that the next lane does not overfill in the LC algorithm
      End If
  Else
    'This is not a front part and is not subject to wait time check
    NPart = Vehicle.Parts(CPartID - 1)
    NL = NPart.L
  End If
Else
  'Is this vehicle eligible to advance based on the minimum headway remaining in next lane?
  Y2move = Min(Y2move, (CC.SubTau - MinHin(NL)) * Vehicle.Length / Vehicle.MinH)
  'Is this vehicle eligible to advance based on the minimum spacing remaining in next lane?
  Y2move = Min(Y2move, NC.Xr(NS, NL) * Vehicle.Length / Vehicle.MinS)
End If
End If
If Y2move > 0 Then
  Dim PPart As cXCSL
  Dim CPartX As Long = 0
  Dim PPartX As Long = 0
  Dim PPartID As Integer = Min(CPartID + 1, Nparts - 1)
  Dim MPart As cXCSL
  Dim LastSubcellID As Integer = CC.NSubCells - 1
  If CS(CCIdx) = LastSubcellID Then
    CPartX = Min(CPart.X, Y2move)
    End If
  PPartX = Y2move - CPartX
  'Remove the disutility from CS, CL
  CC.Disutility(LastSubcellID, CL) -= CPartX * (1 - CPartID / Nparts)
  'Remove the disutility from CS-1, CL
  CC.Disutility(LastSubcellID - 1, CL) -= PPartX * (1 - PPartID / Nparts)
  If CS(CCIdx) = LastSubcellID - 1 Or CPartID = Nparts - 1 Then
    'Vehicle part exists in one of the two subcells
    If VehX = Vehicle.Length Then
      'This is a whole vehicle in CS
      If Y2move = VehX Then
        'Advance the whole vehicle
        'Remove it from VQ of CS
        CC.VQ(CS(CCIdx)).Remove(VehID)
        'update vehicle part coordinates
        CPart.C = NC
        CPart.S = NS
        CPart.L = NL
      Else
        'Advance a part of this whole vehicle
        VSplit(Y2move, CPart, NC, NS, NL)
        'Place the remaining part CPart in front of VQ of the current subcell
        CC.VQ(CS(CCIdx)).Remove(VehID)
        CC.VQ(CS(CCIdx)).Add(CPart, VehID, CSIdx(CCIdx, LastSubcellID - CS(CCIdx)))
      End If
    CIdx(CCIdx, LastSubcellID - CS(CCIdx)) += 1
    idx(CCIdx) += 1
    CPartX = Min(CPart.X, Y2move)
    CC.VQ(CS(CCIdx)).Add(CPart, VehID, CSIdx(CCIdx, LastSubcellID - CS(CCIdx)))
  End If
End If
VBlockLanes(CC, NC, mConnector, CL, NL, NLOpen, NOpenLanes)

End If

ElseIf CPartID < Nparts - 1 Then
' This is a first part in CS; it must be in the next-to-last subcell
'We must split it anyway; a zero part may be created if Y2move = VehX
VSplit(Y2move, CPart, NC, NS, NL)
' Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS(CCIdx)).Remove(VehID)
CC.VQ(CS(CCIdx)).Add(CPart, VehID, CSIdx(CCIdx, LastSubcellID - CS(CCIdx)))
CSIdx(CCIdx, 1) += 1

If CS(CCIdx) < LastSubcellID Then
' Create a Zero part in the last subcell to ensure processing the remaining parts first
VSplit(0, CPart, CC, LastSubcellID, CL)
Dim ZPart As cXCSL = Vehicle.Parts(CPart.ID - 1)
CC.VQ(LastSubcellID).Add(ZPart, VehID, CSIdx(CCIdx, 0))
CSIdx(CCIdx, 0) += 1
End If

idx(CCIdx) += 1
VBlockLanes(CC, NC, mConnector, CL, NL, NLOpen, NOpenLanes)

Else
' This is a last part, it must be in the last subcell
'Split it only if we cannot advance the whole part
If Y2move < VehX Then
' Split Y2move out of VehX
VSplit(Y2move, CPart, NC, NS, NL)
' Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS(CCIdx)).Remove(VehID)
CC.VQ(CS(CCIdx)).Add(CPart, VehID, CSIdx(CCIdx, LastSubcellID - CS(CCIdx)))
CSIdx(CCIdx, LastSubcellID - CS(CCIdx)) += 1
idx(CCIdx) += 1
VBlockLanes(CC, NC, mConnector, CL, NL, NLOpen, NOpenLanes)

Else
' The last part is advancing out in full
' Remove it from VQ of CS
CC.VQ(CS(CCIdx)).Remove(VehID)
End If
End If

Else
' Vehicle part spans both subcells
PPart = Vehicle.Parts(PPartID)
If Y2move <= CPartX Then
' Advance out of the last subcell ONLY; we must split Y2move out of the part in last subcell

190
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS(CCIdx)).Remove(VehID)
CC.VQ(CS(CCIdx)).Add(CPart, VehID, CSIdx(CCIdx, 0))
CSIdx(CCIdx, 0) += 1
idx(CCIdx) += 1
VBlockLanes(CC, NC, mConnector, CL, NL, NLOpen, NOpenLanes)

Else
'Advance out of both subcells
If Y2move = Vehicle.Length Then
 'A whole vehicle is advancing out of both subcells; no need to split, just
join
VJoin(PPart, CPart)
'update vehicle part coordinates
CPart.C = NC
CPart.S = NS
CPart.L = NL
'Remove both parts from VQ
CC.VQ(CS(CCIdx)).Remove(VehID)
CC.VQ(LastSubcellID - 1).Remove(VehID)

Else
'This is not a whole vehicle
'is either a last part but not all part is advancing(Y2move<VehX)
'or a middle part advancing in full(need to leave something in place
If Y2move < VehX Or PPartID < Nparts - 1 Then
'First, split the part in next-to-last subcell
'Take ?Y2move - CPartX? out of PPartX
VSplit(PPartX, PPart, NC, NS, NL)
MPart = Vehicle.Parts(PPartID)
'Then, join
VJoin(MPart, CPart)
'Place the remaining part CPart in front of VQ of the next to last
subcell
CC.VQ(LastSubcellID - 1).Remove(VehID)
CC.VQ(LastSubcellID - 1).Add(PPart, Vehicle.ID, CSIdx(CCIdx, 1))
CSIdx(CCIdx, 1) += 1
'Create a Zero part in the last subcell
VSplit(CPart.X, CPart, CC, LastSubcellID, CL)
idx(CCIdx) += 1
If CPartID = 0 Then
 'update vehicle part coordinates
MPart = Vehicle.Parts(CPartID)
MPart.C = NC
MPart.S = NS
MPart.L = NL
End If
End If
VBlockLanes(CC, NC, mConnector, CL, NL, NLOpen, NOpenLanes)

Else
    'The last part is advancing out in full
    'Join PPart and CPart
    VJoin(PPart, CPart)
    'Remove this vehicle from last subcell
    CC.VQ(CS(CCIdx)).Remove(VehID)
    'Remove it also from the previous subcell
    CC.VQ(LastSubcellID - 1).Remove(VehID)
End If
End If
End If
End If
End If
End If

'update subX
CC.SubX(LastSubcellID) -= CPartX
CC.SubX(LastSubcellID - 1) -= PPartX
NC.Xr(NS, NL) -= CLng(Y2move * Vehicle.MinS / Vehicle.Length)
CC.Xr(LastSubcellID, CL) += CLng(CPartX * Vehicle.MinS / Vehicle.Length)
CC.Xr(LastSubcellID - 1, CL) += CLng(PPartX * Vehicle.MinS / Vehicle.Length)
mConnector.Y += Y2move
mConnector.YFront += Min(CPartX, Y2move)
'How much is remaining in the last subcell?
Dim FrontY2move As Long = CPartX 'store before you loose it
CPartX = Max((CPartX - Y2move), 0)
'How much is remaining in the previous subcell?
PPartX = Min(PPartX, VehX - Y2move)
CC.Disutility(LastSubcellID - 1, CL) += PPartX * (1 - PPartID / Vehicle.Parts.Count)
CC.Disutility(LastSubcellID, CL) += CPartX * (1 - CPartID / Vehicle.Parts.Count)
MPart = Vehicle.Parts(CPartID)
NC.Disutility(NS, NL) += Y2move * (1 - MPart.ID / Vehicle.Parts.Count)
'Place this vehicle in the buffer of next cell
NC.PartBuffer.Add(MPart)
'Remove this vehicle from Queue, Parts, and QueueX
Q(CCIdx).Remove(VehID)
QX(CCIdx).Remove(VehID)
NC.SubX(NS) += Y2move
Yreceived(NL) += Y2move
TotalReceived += Y2move
If CPartID = 0 Then
    'Update platoons for a front part
    Platoon.Add(NL, Vehicle, CC, CS(CCIdx), Yreceived(NL))
    Vehicle.EntryTick = Vehicle.ExitTick
Vehicle.ExitTick = Me.Tick
End If
MinHout(CCIdx)(CL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
'check if the current lane should be blocked based on flow capacity
If CC.SubTau = MinHout(CCIdx)(CL) Then
    'block the current lane
    CLOOpen(CCIdx)(CL) = False
    NCCurrentLanesOpen(CCIdx) -= 1
End If
MinHin(NL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
'check if the next lane should be blocked based on flow capacity
If CC.SubTau = MinHin(NL) And NLOOpen(NL) Then
    'block the next lane
    VBlockLanes(CC, NC, ijConnector, CL, NL, NLOOpen, NNNextLanesOpen)
End If
Else
    'We could not advance this vehicle because of lane changing
    idx(CCIdx) += 1
End If
If mConnector.Y = Max2Send(CCIdx) Or idx(CCIdx) > Q(CCIdx).Count Then
    'CC must be closed
    CCOpen(CCIdx) = False 'no more vehicle can be advanced from this cell
End If

Loop Until NOpenLanes = 0 Or TotalReceived = Max2Receive
'Adjust the vehicle wait times in the next subcell by lane
VAdjustNSWaitTimes(NC.NLanes, NC, NetworkTau, Platoon, Yreceived, "E")
For i As Integer = 0 To Nk - 1
    'Adjust the vehicle wait times of front parts remaining in the last subcell of current cell ONLY
    'The one next-to-last will be taken care of in internal update
    VAdjustCSWaitTimes(CCells(i), CCells(i).NSubCells - 1, NetworkTau)
Next
End Sub
Sub Stage2_Diverge(ByRef ijConnector As cDConnector, ByVal NetworkTau As Integer)

'Sending cell is icell and receiving cell is jcell
Dim CC As cCell = ijConnector.FromCell
Dim NC As cCell = Nothing
Dim Nj As Integer = CC.Successors.Count
Dim CCNLanes As Int16 = CC.NLanes
'Monitor how many have advanced in by lane
Dim Ysent As Long() = Array.CreateInstance(GetType(Long), CCNLanes)
'Stochastic variations in free-flow speeds
'Create collections of Platoon, PlatoonX, and PlatoonCS for each lane out of the DCell
Dim Platoon As cPlatoon = New cPlatoon(CCNLanes)

Dim NS As Integer = 0
Dim CS As Integer = -1
Dim idx As Integer = 1
Dim Y2move As Long = 0
Dim NL As Integer = -1
Dim NOpenLanes As Integer = 0
Dim Max2Send As Long = 0
Dim TotalSent As Long = 0
Dim CLOpen As Boolean() = Array.CreateInstance(GetType(Boolean), CCNLanes)
Dim NLOpen As Array() = Array.CreateInstance(GetType(Array), Nj)
Dim NCOpen As Boolean() = Array.CreateInstance(GetType(Boolean), Nj)
Dim MinHout As Integer() = Array.CreateInstance(GetType(Integer), CCNLanes)
Dim MinHin As Array() = Array.CreateInstance(GetType(Array), Nj)
Dim NSIdx As Integer() = Array.CreateInstance(GetType(Integer), Nj)
Dim NNextLanesOpen As Integer() = Array.CreateInstance(GetType(Integer), Nj)
Dim NCurrentLanesOpen As Integer = CCNLanes
Dim Q As New Collection
Dim QX As New Collection

For i As Integer = 0 To Nj - 1
    Dim dConnector As cDConnector = CC.Successors(i + 1)
    Dim ReceivingCell As cCell = dConnector.ToCell
    Dim NCNlanes As Integer = ReceivingCell.NLanes
    NOpenLanes += ReceivingCell.NLanes
    NCOpen(i) = True
Next

End Sub
NLOpen(i) = Array.CreateInstance(GetType(Boolean), NCNLanes)
MinHin(i) = Array.CreateInstance(GetType(Integer), NCNLanes)
NNextLanesOpen(i) = NCNLanes
For j As Integer = 0 To NCNLanes - 1
    CType(NLOpen(i), Boolean())(j) = True
Next
NSIdx(i) = 1
Max2Send += dConnector.y
dConnector.Y = 0
NNextLanesOpen(i) = NCNLanes
Next i

For i As Integer = 0 To CCNLanes - 1
    CLOpen(i) = True
Next

'exit if nothing advances out of the diverging
If Max2Send = 0 Then Exit Sub

For CS = CC.NSubCells - 1 To CC.NSubCells - 2 Step -1
    While idx <= CC.VQ(CS).Count
        Dim Part As cXCSL = CC.VQ(CS)(idx)
        Try
            Q.Add(Part, Part.V.ID)
            QX.Add(Part.X, Part.V.ID)
        Catch ex As ArgumentException 'update the Qx
            Dim tX As Long = QX(Part.V.ID)
            QX.Remove(Part.V.ID)
            QX.Add(tX + Part.X, Part.V.ID)
        End Try
        idx += 1
    End While
    idx = 1 'reset idx for next-to-last subcell
Next

'reset index to 1 for collection-type index\
idx = 1
Dim CSIdx As Integer() = Array.CreateInstance(GetType(Integer), 2)
CSIdx(0) = 1
CSIdx(1) = 1

Do While idx <= Q.Count And NOpenLanes > 0 And TotalSent < Max2Send
    Dim CPart As cXCSL = Q(idx)
    Dim NPart As cXCSL = Nothing
    Dim CPartID As Integer = CPart.ID

Dim Vehicle As cVehicle = CPart.V
Dim VehID As String = Vehicle.ID
Dim Nparts As Integer = Vehicle.Parts.Count
Dim VehX As Long = QX(VehID)
Dim CL As Integer = CPart.L
Dim dConnector As cDConnector = CType(Vehicle.TargetConnector(CC.ID),
cDConnector)
   NC = dConnector.ToCell
Dim NCIdx As Integer = dConnector.orderID

CS = CPart.S
Y2move = Min(VehX, Max2Send - dConnector.Y)
If CLOpen(CL) And NCOpen(NCIdx) Then
   Y2move = Min(VehX, Max2Send - dConnector.Y)
   'is this vehicle eligible to advance based on the minimum headway remaining in current lane?
   Y2move = Min(Y2move, Int((CC.SubTau - MinHout(CL)) * Vehicle.Length / Vehicle.MinH))
End If
If Y2move > 0 Then
   'Is this vehicle eligible to advance based on its minimum free-flow travel time in the current subcell?
   If CPartID = 0 Then
      'This is a front part that we need to check its wait time against the minimum wait time
      VCheckMinWait(Vehicle, CC, CS, Y2move, "E")
      If Y2move > 0 Then
         'NL = use mandatory LC or discretionary LC based on cell type
         NL = ComputeNextLaneExternal(Y2move, Vehicle, CS, CL, CC, NC, dConnector.ID, NetworkTau)
         If NL < 0 Then
            Y2move = 0
         ElseIf Not CType(NLOpen(NCIdx), Boolean())(NL) Then
            Y2move = 0
         End If
         'Ensure that the next lane does not overfill in the LC algorithm
      End If
   Else
      'This is not a front part and is not subject to wait time check
      NPart = Vehicle.Parts(CPartID - 1)
      NL = NPart.L
   End If
   If Y2move > 0 Then
      'is this vehicle eligible to advance based on the minimum headway remaining in next lane?
Y2move = Min(Y2move, Int((CC.SubTau - MinHin(NCIdx)(NL)) * Vehicle.Length / Vehicle.MinH))

'Is this vehicle eligible to advance based on the minimum spacing remaining in next lane?

Y2move = Min(Y2move, NC.Xr(NS, NL) * Vehicle.Length / Vehicle.MinS)

End If
End If
If Y2move > 0 Then
  Dim PPart As cXCSL
  Dim CPartX As Long = 0
  Dim PPartX As Long = 0
  Dim PPartID As Integer = Min(CPartID + 1, Nparts - 1)
  Dim MPart As cXCSL
  Dim LastSubcellID As Integer = CC.NSubCells - 1
  If CS = LastSubcellID Then
    CPartX = Min(CPart.X, Y2move)
    End If
  PPartX = Y2move - CPartX
  'Remove the disutility from CS, CL
  CC.Disutility(LastSubcellID, CL) -= CPartX * (1 - CPartID / Nparts)
  'Remove the disutility from CS-1, CL
  CC.Disutility(LastSubcellID - 1, CL) -= PPartX * (1 - PPartID / Nparts)

  If CS = LastSubcellID - 1 Or CPartID = Nparts - 1 Then
    'Vehicle part exists in one of the two subcells
    If VehX = Vehicle.Length Then
      'This is a whole vehicle in CS
      If Y2move = VehX Then
        'Advance the whole vehicle
        'Remove it from VQ of CS
        CC.VQ(CS).Remove(VehID)
        'update vehicle part coordinates
        CPart.C = NC
        CPart.S = NS
        CPart.L = NL
      Else
        'Advance a part of this whole vehicle
        VSplit(Y2move, CPart, NC, NS, NL)
        'Place the remaining part CPart in front of VQ of the current subcell
        CC.VQ(CS).Remove(VehID)
        CC.VQ(CS).Add(CPart, VehID, CSIdx(LastSubcellID - CS))
        CSIdx(LastSubcellID - CS) += 1
        idx += 1
      If CS < LastSubcellID Then
        'And CC.FrontTau > 0 And
        CType(Vehicle.Parts(CPart.ID), cXCSL).C.ID <> CC.ID Then

      End If
    End If
  End If
End If
'Create a Zero part in the last subcell to ensure processing the remaining parts first
  VSsplit(0, CPart, CC, LastSubcellID, CL)
Dim ZPart As cXCSL = Vehicle.Parts(CPart.ID - 1)
CC.VQ(LastSubcellID).Add(ZPart, VehID, CSIdx(0))
CSIdx(0) += 1
End If
VBlocKlanes(CC, NC, dConnector, CL, NL, CType(NLOpen(NCIdx), Boolean()), NOpenLanes)
ElseIf CPart.ID < Nparts - 1 Then
  'This is not the last part in CS; it must be in the next-to-last subcell
  'We must split it anyway; a zero part may be created if Y2move = VehX
  VSsplit(Y2move, CPart, NC, NS, NL)
  'Place the remaining part CPart in front of VQ of the current subcell
  CC.VQ(CS).Remove(VehID)
  CC.VQ(CS).Add(CPart, VehID, CSIdx(LastSubcellID - CS))
  CSIdx(1) += 1
If CS < LastSubcellID Then
  'Create a Zero part in the last subcell to ensure processing the remaining parts first
  VSsplit(0, CPart, CC, LastSubcellID, CL)
Dim ZPart As cXCSL = Vehicle.Parts(CPart.ID - 1)
CC.VQ(LastSubcellID).Add(ZPart, VehID, CSIdx(0))
CSIdx(0) += 1
End If
idx += 1
VBlocKlanes(CC, NC, dConnector, CL, NL, CType(NLOpen(NCIdx), Boolean()), NOpenLanes)
Else
  'This is a last part, it must be in the last subcell
  'Split it only if we cannot advance the whole part
  If Y2move < VehX Then
    'Split Y2move out of VehX
    VSsplit(Y2move, CPart, NC, NS, NL)
    'Place the remaining part CPart in front of VQ of the current subcell
    CC.VQ(CS).Remove(VehID)
    CC.VQ(CS).Add(CPart, VehID, CSIdx(LastSubcellID - CS))
    CSIdx(LastSubcellID - CS) += 1
    idx += 1
    VBlockLanes(CC, NC, dConnector, CL, NL, CType(NLOpen(NCIdx), Boolean()), NOpenLanes)
  Else
    'The last part is advancing out in full
    'Remove it from VQ of CS
    CC.VQ(CS).Remove(VehID)
Else
'Sequential part spans both subcells
PPart = Vehicle.Parts(PPartID)
If Y2move <= CPartX Then
'Advance out of the last subcell ONLY; we must split Y2move out of the part in last subcell
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part CPart in front of VQ of the current subcell
CC.VQ(CS).Remove(VehID)
CC.VQ(CS).Add(CPart, VehID, CSIdx(0))
CSIdx(0) += 1
idx += 1
VBlockLanes(CC, NC, dConnector, CL, NL, CType(NLOpen(NCIdx), Boolean()), NOpenLanes)
Else
'Advance out of both subcells
If Y2move = Vehicle.Length Then
'A whole vehicle is advancing out of both subcells; no need to split, just join
VJoin(PPart, CPart)
'update vehicle part coordinates
CPart.C = NC
CPart.S = NS
CPart.L = NL
'Remove both parts from VQ
CC.VQ(CS).Remove(VehID)
CC.VQ(LastSubcellID - 1).Remove(VehID)
Else
'This is not a whole vehicle
'is either a last part but not all part is advancing(Y2move<Vehx)
or a middle part advancing in full(need to leave something in place
If Y2move < VehX Or PPartID < Nparts - 1 Then
'First, split the part in next-to-last subcell
'Take ?Y2move - CPartX? out of PPartX
VSplit(PPartX, PPart, NC, NS, NL)
MPart = Vehicle.Parts(PPartID)
'Then, join
VJoin(MPart, CPart)
'Place the remaining part CPart in front of VQ of the next to last subcell
CC.VQ(LastSubcellID - 1).Remove(VehID)
CC.VQ(LastSubcellID - 1).Add(PPart, Vehicle.ID, CSIdx(1))
CSIdx(1) += 1
'Create a Zero part in the last subcell
VSplit(CPart.X, CPart, CC, LastSubcellID, CL)
idx += 1
If CPartID = 0 Then
 'update vehicle part coordinates
 MPart = Vehicle.Parts(CPartID)
 MPart.C = NC
 MPart.S = NS
 MPart.L = NL
End If
VBlockLanes(CC, NC, dConnector, CL, NL, CType(NLOpen(NCIdx),
Boolean()), NOpenLanes)
Else
 'The last part is advancing out in full
 'Join PPart and CPart
 VJoin(PPart, CPart)
 'Remove this vehicle from last subcell
 CC.VQ(CS).Remove(VehID)
 'Remove it also from the previous subcell
 CC.VQ(LastSubcellID - 1).Remove(VehID)
End If
End If
End If
 'update subX, Xr
 CC.SubX(LastSubcellID) -= CPartX
 CC.SubX(LastSubcellID - 1) -= PPartX
 NC.Xr(NS, NL) -= CLng(Y2move * Vehicle.MinS / Vehicle.Length)
 CC.Xr(LastSubcellID, CL) += CLng(CPartX * Vehicle.MinS / Vehicle.Length)
 CC.Xr(LastSubcellID - 1, CL) += CLng(PPartX * Vehicle.MinS / Vehicle.Length)
 dConnector.Y += Y2move
 TotalSent += Y2move
 dConnector.YFront += Min(CPartX, Y2move)
 'How much is remaining in the last subcell?
 Dim FrontY2move As Long = CPartX 'store before you loose it
 CPartX = Max((CPartX - Y2move), 0)
 'How much is remaining in the previous subcell?
 PPartX = Min(PPartX, VehX - Y2move)
 CC.Disutility(LastSubcellID - 1, CL) += PPartX * (1 - PPartID / Vehicle.Parts.Count)
 CC.Disutility(LastSubcellID, CL) += CPartX * (1 - CPartID / Vehicle.Parts.Count)
 MPart = Vehicle.Parts(CPartID)
 NC.Disutility(NS, NL) += Y2move * (1 - MPart.ID / Vehicle.Parts.Count)
 'Place this vehicle in the buffer of next cell
NC.PartBuffer.Add(MPart)
' Remove this vehicle from Queue, Parts, and QueueX
Q.Remove(VehID)
QX.Remove(VehID)
NC.SubX(NS) += Y2move
Ysent(CL) += Y2move
If CPartID = 0 Then
' Update platoons for a front part
Platoon.Add(CL, Vehicle, CC, CS, Ysent(CL))
Vehicle.EntryTick = Vehicle.ExitTick
Vehicle.ExitTick = Me.Tick
End If
If NC.Type = CellType.DIVERGE Then
   Dim i As Integer = CType(Vehicle.TargetConnector(NC.ID), cConnector).orderID
   CType(NC, cDCell).yInByDest(i) += Y2move
   CType(NC, cDCell).yInFrontByDest(i) += FrontY2move
End If
MinHout(CL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
' check if the current lane should be blocked based on flow capacity
If CC.SubTau = MinHout(CL) Then
   ' block the current lane
   CLOpen(CL) = False
   NCurrentLanesOpen -= 1
End If
MinHin(NCIdx)(NL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
' check if the next lane should be blocked based on flow capacity
If CC.SubTau = MinHin(NCIdx)(NL) And NCOpen(NCIdx) Then
   ' block the next lane
   VBlockLanes(CC, NC, ijConnector, CL, NL, CType(NLOpen(NCIdx), Boolean()), NNextLanesOpen(NCIdx))
End If
Else
   ' We could not advance this vehicle because of lane changing
   idx += 1
End If
Loop
' Adjust the vehicle wait times in the next subcell by lane
VAdjustNSWaitTimes(CC.NLanes, CC, NetworkTau, Platoon, Ysent, "E")
' Adjust the vehicle wait times of front parts remaining in the last subcell of current cell
ONLY
   ' The one next-to-last will be taken care of in internal update
   VAdjustCSWaitTimes(CC, CC.NSubCells - 1, NetworkTau)
End Sub
Sub Stage3(ByRef SourceCell As cCell, ByVal networkTau As Integer)

    Dim CC As cCell = SourceCell
    Dim NC As cCell = CC

    If CC.Type = CellType.SOURCE Then
        CC.X -= CType(CC.Successors(1), cConnector).Y
        Exit Sub
    ElseIf CC.Type = CellType.SINK Then
        CC.X += CType(CC.Predecessors(1), cConnector).Y
        'GoTo CleanBuffer
        Exit Sub
    End If

    'The following cell properties are required to perform stage 2:
    'Cell.Disutility(m,l) = Lane changing disutility of lane l and subcell m
    'Cell.SubX(m) = number of vehicles in subcell m
    'Cell.VQ(m) = an array of pointers to the vehicle parts in subcell m ordered by arrival
    'into the subcell
    'Cell.SubY(m) = the number of vehicles estimated to advance from subcell m to m+1
    'In addition, each vehicle has the following properties:
    'XCSL.X = The amount of vehicle part p
    'XCSL.C = The cell of vehicle part p
    'XCSL.S = The subcell of vehicle part p
    'XCSL.L = The lane of vehicle part p
    'Vehicle.Parts = a dynamic array containing pointers to all vehicle parts
    'Vehicle.Length = the ratio of the vehicle length to the length of a typical passenger
    'car
    'Vehicle.ffs = the assigned free-flow speed of vehicle at time of entry
    'Vehicle.Wait = the amount of time spent by the vehicle in each subcell
    'vehicle.MinH=the minimum headway acceptable by the driver
    ' = MinGap+VehicleLen(ft)/VehSpeedAtCapacity
    Dim NLanes = CC.NLanes
    'Pseudo(Code)
    For CS As Integer = CC.NSubCells - 2 To 0 Step -1
        Dim NS As Integer = CS + 1
        Dim Max2advance As Long = CC.SubY(CS)
        If Max2advance > 0 Then
Utils.t1 = DateTime.Now

CC.SubY(CS) = 0 'reset the internal connector
Dim Yreceived As Long() = Array.CreateInstance(GetType(Long), NLanes)
Dim NCurrentLanesOpen As Integer = NLanes
Dim NNextLanesOpen As Integer = NLanes
Dim CLOpen As Boolean() = Array.CreateInstance(GetType(Boolean), NLanes)
Dim NLOpen As Boolean() = Array.CreateInstance(GetType(Boolean), NLanes)
Dim MinHout As Integer() = Array.CreateInstance(GetType(Integer), NLanes)
Dim MinHin As Integer() = Array.CreateInstance(GetType(Integer), NLanes)
Dim Platoon As cPlatoon = New cPlatoon(NLanes)

'Create collections of Platoon and PlatoonX for each lane in the next subcell
Dim idx As Integer = 1
Dim CSidx As Integer = 1
Dim NSidx As Integer = 1
Dim Vehicle As cVehicle = CPart.V
Dim VehID As String = Vehicle.ID
Dim NParts As Integer = Vehicle.Parts.Count
Dim VehX As Integer = CPart.X
Dim CL As Integer = CPart.L
Dim NL As Integer = -1
Dim Y2move As Long = 0

If CLOpen(CL) Then
    Y2move = Min(VehX, Max2advance - CC.SubY(CS))
    'is this vehicle eligible to advance based on the minimum headway remaining in current lane?
    Y2move = Min(Y2move, Int((CC.SubTau - MinHout(CL)) * Vehicle.Length / Vehicle.MinH))
End If

If CPart.ID = 0 Then
'This is a front part that we need to check its wait time against the minimum wait time

VCheckMinWait(Vehicle, CC, CS, networkTau, Y2move)
If Y2move > 0 Then
    'NL = use mandatory LC or discretionary LC based on cell type
    'Ensure that the next lane does not overfill in the LC algorithm
    NL = ComputeNextLaneInternal(Y2move, Vehicle, CS, CL, CC, networkTau)
    If NL < 0 Then Y2move = 0
End If
Else
    'This is not a front part and is not subject to wait time check
    NPart = Vehicle.Parts(CPart.ID - 1)
    NL = NPart.L
End If
If Y2move > 0 Then
    'is this vehicle eligible to advance based on the minimum headway remaining in next lane?
    Y2move = Min(Y2move, Int((CC.SubTau - MinHin(NL)) * Vehicle.Length / Vehicle.MinH))
    'Is this vehicle eligible to advance based on the minimum spacing remaining in next lane?
    Y2move = Min(Y2move, NC.Xr(NS, NL) * Vehicle.Length / Vehicle.MinS)
End If
End If

If Y2move > 0 Then
    Dim CPartID As Integer = CPart.ID
    'Remove the disutility from CS, CL
    CC.Disutility(CS, CL) -= VehX * (1 - CPartID / NParts)
    Yreceived(NL) += Y2move
    If VehX = Vehicle.Length Then
        'A whole vehicle is advancing
        If Y2move = VehX Then
            'Advance the whole vehicle
            'Remove it from VQ of CS
            CC.VQ(CS).Remove(VehID)
            NC.VQ(NS).Add(CPart, VehID)
            CPart.S = NS
            CPart.L = NL
        Else
            'Advance a part of this whole vehicle
            'Split Y2move out of VehX
            VSplit(Y2move, CPart, NC, NS, NL)
            'Place the remaining part in front of VQ of current subcell
        End If
    Else
        'Advance a part of this whole vehicle
        'Split Y2move out of VehX
        VSplit(Y2move, CPart, NC, NS, NL)
        'Place the remaining part in front of VQ of current subcell
    End If
CC.VQ(CS).Remove(VehID)
CC.VQ(CS).Add(CPart, VehID, CSidx)
CSidx += 1
'Place the moving part into next subcell
NC.VQ(NS).Add(Vehicle.Parts(CPartID), VehID)
idx += 1
NLOpen(NL) = False
NNextLanesOpen -= 1
End If
'Update Platoons ONLY if moving a front part
Platoon.Add(NL, Vehicle, CC, CS, Yreceived(NL))
ElseIf CPartID = 0 Then
'This is a first part; Split it anyway
VSplit(Y2move, CPart, NC, NS, NL)
'Keep it in VQ even if it leaves a zero part
'Place the remaining part in front of VQ of current subcell
CC.VQ(CS).Remove(Vehicle.ID)
CC.VQ(CS).Add(CPart, Vehicle.ID, CSidx)
CSidx += 1
'Add it to VQ of NS
NC.VQ(NS).Add(Vehicle.Parts(CPartID), Vehicle.ID)
idx += 1
NLOpen(NL) = False
NNextLanesOpen -= 1
Else
'This is not a front part
If CPartID = NParts - 1 And Y2move = VehX Then
'This is the last part and can advance in full
'Remove it from VQ of CS
CC.VQ(CS).Remove(Vehicle.ID)
CPart.S = NS
CPart.L = NL
Else
'This part cannot advance in full, split it
VSplit(Y2move, CPart, NC, NS, NL)
'Place the remaining part in front of VQ of current subcell
CC.VQ(CS).Remove(Vehicle.ID)
CC.VQ(CS).Add(Vehicle.Parts(CPartID + 1), Vehicle.ID, CSidx)
CSidx += 1
idx += 1
NLOpen(NL) = False
NNextLanesOpen -= 1
End If
'There should be a part of this vehicle in the next subcell
MPart = Vehicle.Parts(CPartID)
VJoin(MPart, NPart)
End If
'Add disutilities to CL and NL
If Vehicle.Parts.Count - 1 < CPartID Then
  MPart = NPart
Else
  MPart = Vehicle.Parts(CPartID)
End If
NC.Disutility(NS, NL) += Y2move * (1 - MPart.ID / Vehicle.Parts.Count)
CC.SubX(CS) -= Y2move
NC.SubX(NS) += Y2move
CC.SubY(CS) += Y2move
Yreceived(NL) += Y2move
NC.Xr(NS, NL) -= CLng(Y2move * Vehicle.MinS / Vehicle.Length)
CC.Xr(CS, CL) += CLng(Y2move * Vehicle.MinS / Vehicle.Length)
If CPartID = 0 Then
  'Update platoons for a front part
  Platoon.Add(NL, Vehicle, CC, CS, Yreceived(NL))
End If
Dim t1 As DateTime = DateTime.Now
MinHout(CL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
MinHin(NL) += Int(Y2move * Vehicle.MinH / Vehicle.Length)
'check if the current lane should be blocked based on flow capacity
If CC.SubTau = MinHout(CL) Then
  'block the current lane
  CLOpen(CL) = False
  NCurrentLanesOpen -= 1
End If
'check if the next lane should be blocked based on flow capacity
If CC.SubTau = MinHin(NL) Then
  'block the next lane
  NLOpen(NL) = False
  NNextLanesOpen -= 1
End If
Dim t2 As DateTime = DateTime.Now
Else
  'We could not advance this vehicle because of LC
  idx += 1
End If
Loop
'Adjust the vehicle wait times in the next subcell by lane
VAdjustNSWaitTimes(NLanes, CC, networkTau, Platoon, Yreceived)
'Adjust the vehicle wait times of front parts remaining in the current subcell
VAjustCSWaitTimes(CC, CS, networkTau)
'Proceed to update of next subcell upstream
End If
Next CS

'Advance vehicles from cell buffer into the first subcell
Do While CC.PartBuffer.Count > 0
  Dim Part As cXCSL = CC.PartBuffer(1)
  CC.PartBuffer.Remove(1)
  'CC.SubXByLane(0, Part.L) += Part.X
  If Part.ID = 0 Then
    'This is a front part that should be added to VQ of next subcell
    CC.VQ(0).Add(Part, Part.V.ID)
  Else
    'This is a non-front part of a vehicle; it must join
    VJoin(Part, CType(Part.V.Parts(Part.ID - 1), cXCSL))
    'Since we joined, we do not need to update the part in VQ
  End If
Loop

'update the F arrays
'it contains the cumulative number of vehicles
'that waited at least kTau time units in the cell
'at the end of the current update
Dim n As Integer = SourceCell.NSubCells
Dim yInByConn As Long = 0
Dim yInFrontByConn As Long = 0
Dim kn As Integer = SourceCell.Predecessors.Count
Dim yIn As Long = 0
Dim yOut As Long = 0
If SourceCell.Type = CellType.DIVERGE Then
  Dim dCell As cDCell = CType(SourceCell, cDCell)
  Dim predConn As cConnector = CType(SourceCell.Predecessors(1), cConnector)
  Dim yOutByConn As Long = 0
  yIn = predConn.Y
  For i As Integer = 0 To SourceCell.Successors.Count - 1
    Dim conn As cDConnector = SourceCell.Successors(i + 1)
    yOutByConn = conn.Y
    yOut += conn.Y
    yInByConn = CType(SourceCell, cDCell).yInByDest(i)
    yInFrontByConn = CType(SourceCell, cDCell).yInFrontByDest(i)
  Next
  yOut = predConn.Y + yOutByConn
  If yOut > 0 Then
    'update all FI's recursively, from back, except the first one
    yOut += yOutByConn
    yIn = yIn + yInByConn
    yInFront = yInFront + yInFrontByConn
  Else
    If yIn > 0 Then
      yIn = yIn + yInByConn
    End If
    If yInFront > 0 Then
      yInFront = yInFront + yInFrontByConn
    End If
  End If
End If
For j As Integer = n - 1 To 2 Step -1
    dCell.Fi(i, j) = Max(dCell.Fi(i, j - 1) - yOutByConn, 0)
Next
'update FI(1)
FI(1) = sourceCell.X
dCell.Fi(i, 1) = dCell.xByDest(i)
If yInByConn * (SourceCell.FrontTau + predConn.FromCell.FrontTau) <
yInByConn * networkTau + yInFrontByConn * predConn.FromCell.FrontTau Then
    dCell.Fi(i, 1) = dCell.Fi(i, 1) + CLng((1 - SourceCell.FrontTau / (networkTau -
    predConn.FromCell.FrontTau * (1 - yInFrontByConn / yInByConn))) * yInByConn)
End If
    dCell.Fi(i, 1) = dCell.Fi(i, 1) - yOutByConn
Next i
Else
    For Each predConn As cConnector In SourceCell.Predecessors
        yIn += predConn.Y
    Next
    Dim SuccConn As cConnector = CType(SourceCell.Successors(1), cConnector)
yOut = SuccConn.Y
'update all FI's recursively, from back, except the first one
For i As Integer = n - 1 To 2 Step -1
    SourceCell.FI(i) = Max(SourceCell.FI(i - 1) - yOut, 0)
Next
'update FI(1)
SourceCell.FI(1) = SourceCell.X
For Each predConn As cConnector In SourceCell.Predecessors
    Dim i As Integer = predconn.orderID
    If predConn.YFront * (SourceCell.FrontTau + predConn.FromCell.FrontTau) <
predConn.Y * networkTau + predConn.YFront * predConn.FromCell.FrontTau Then
        SourceCell.FI(1) = SourceCell.FI(1) + CLng((1 - SourceCell.FrontTau / (networkTau -
        predConn.FromCell.FrontTau * (1 - predConn.Yfront / predConn.Y))) * predConn.Y)
    End If
Next
SourceCell.FI(1) = SourceCell.FI(1) - yOut
End If

'finally update sourcecell within the current update
'ordinary, merge:

SourceCell.X = SourceCell.X + yIn - yOut
If SourceCell.Type = CellType.DIVERGE Then
    Dim sum As Long = 0
    For i As Integer = 0 To SourceCell.Successors.Count - 1
        Dim succConn As cDConnector = CType(SourceCell.Successors(i + 1), cDConnector)
CType(SourceCell, cDCell).xByDest(i) += CType(SourceCell, cDCell).yInByDest(i) - succconn.y
CType(SourceCell, cDCell).yInByDest(i) = 0
CType(SourceCell, cDCell).yInFrontByDest(i) = 0
Next i
End If
End Sub
VITA

Ciprian Alecsandru was born in Birlad, Romania, in 1974. He received the degree of Bachelor of Engineering in electronics for transportation engineering from Polytechnic University of Bucharest, Romania, in 1997, with distinction. One year later he received a Master of Science degree in telematics for transportation at the same university. Four years before he joined the graduate program in Civil Engineering at Louisiana State University in Fall 2001, he worked as an assistant lecturer at the Transportation Faculty at Polytechnic University of Bucharest. He will complete the degree of Doctor of Philosophy in civil engineering in August 2006.