NDOT Research Report

Report No. 557-14-803

Development and Calibration of Regional Dynamic Traffic Assignment Models for the Estimation of Traffic Performance Measures in Nevada

February 2017

Nevada Department of Transportation
1263 South Stewart Street
Carson City, NV 89712

Disclaimer

This work was sponsored by the Nevada Department of Transportation. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of Nevada at the time of publication. This report does not constitute a standard, specification, or regulation.

DEVELOPMENT AND CALIBRATION OF REGIONAL

DYNAMIC TRAFFIC ASSIGNMENT MODELS

FOR THE ESTIMATION OF TRAFFIC PERFORMANCE MEASURES IN NEVADA

University of Nevada, Reno

Center for Advanced Transportation Education and Research (CATER)

DEVELOPMENT AND CALIBRATION OF REGIONAL

DYNAMIC TRAFFIC ASSIGNMENT MODELS

FOR THE ESTIMATION OF TRAFFIC PERFORMANCE MEASURES IN NEVADA

Prepared by:

Ali Gholami, Ph.D.

Zong Tian, Ph.D., P.E.

Dongmei Lin, Ph.D.

Center for Advanced Transportation Education and Research (CATER)

University of Nevada, Reno, NV

Xuesong Zhou, Ph.D.

Golnoosh Miri

Jiangtao Liu

Pengfei Li

School of Sustainable Engineering and the Built Environment

Arizona State University, Tempe, AZ

February 2017

EXECUTIVE SUMMARY

This project covered the development and calibration of a Dynamic Traffic Assignment (DTA) model and explained the procedures, constraints, and considerations for usage of this model for the Reno-Sparks area roadway network in Northern Nevada.

A literature review was conducted regarding DTA model development and applications. The details of the DTA model development for the Reno-Sparks network were documented along with necessary model calibration. DTALite was used for developing the DTA model and NeXTA was implemented as the user interface. This software package was selected since it has a rigorous traffic queuing model and its built-in parallel computing capability speeds up the analysis process by using multi-core CPU hardware.

Since the Regional Transportation Commission of Washoe County (RTC) maintains the regional TransCAD travel demand model, the up-to-date network and Origin Destination (OD) demand data was obtained from RTC. After initial model preparation, the model was calibrated by comparing the DTA generated link volumes and those obtained from NDOT's TRINA database, which includes 24-hour link volumes on major roadway links collected from permanent and temporary traffic count stations.

The DTA model development included the following major steps:

- Import the network and demand data from TransCAD into NeXTA;
- Perform a dynamic traffic assignment in DTALite to achieve an equilibrium to produce an initial DTA model;
- Prepare field data and run Origin Destination Matrix Estimation (ODME) for initial network calibration;
- Validate and demand modifications to better match the observed counts versus simulation volumes;
- Cut a subarea and perform further analysis in micro simulation software packages (optional).

Major findings from this research are summarized below:

Capabilities and Benefits of DTA

- DTA is mesoscopic in nature, providing a connection between regional travel demand forecasting and micro-simulation models. It is one step further from the planning level of travel forecasting towards the operating details of micro-simulation, i.e., DTA analyzes large networks as a travel demand forecasting tool and provides time-varying traffic network performance (e.g., queue formation, bottleneck identification) but not as much detail as micro-simulation models.

- Compared to micro-simulation models, which normally represent known traffic flow patterns, DTA can represent both current traffic performance and evaluate near-term traffic flow impacts from network changes. It is particularly useful to model a regional level network to forecast traffic flow pattern changes and operational impacts due to incidents such as work zone, special events, and accidents.

Requirements for DTA Development and Applications

- Geometric data, traffic control data, traffic demand, OD demand data and transit demand are basic requirements for network development.
- For model calibration, the fidelity of a DTA model depends on more than link volumes. Typical types of data for calibration strategies can include: volume, travel times, and travel speeds. However, since travel time speed had some issues in this study, only volume was used for calibration. Travel time and travel speed were used for some major links validation.
- Transportation modeling techniques and various levels of efforts are needed depending on the model complexity and data availability.

Limitations of DTA Applications

- For long-term planning, DTA may not be able to produce a well-calibrated model because of the lack of future travel demand data and corresponding field data. Instead, travel-forecasting models such as TransCAD are a better fit for bottleneck estimation studies. While DTA can only identify active bottlenecks, travel demand forecasting models can predict all potential bottleneck locations, which is necessary for long-term planning purposes.
- The level of precision from DTA models largely depends on data availability. DTA requires a significantly larger amount of data, which may not be readily available in most cases. Decision makers need to assess the desired level of

precision and the available resources and choose if DTA or conventional travel demand models should be used.

- For a more localized network or subarea where detailed traffic operational analysis is desired (e.g., transit service and pedestrian facility, turn pockets design, signal control, freeway reconstruction), micro-simulation is a better tool.
- DTA is not suitable for simple traffic impact analysis (TIA).

NDOT Future Usage of DTA Model

Though the Reno/Sparks DTA model was calibrated, similar to STA models, the DTA model also needs maintenance and some consideration for each usage. DTA model users should consider the following:

- Continual System Monitoring and Recalibration: A transportation network is constantly changing in various ways. This means the DTA model requires a regular data collection effort supported by expertise in the area of rigorous model calibration and validation. Ideally, if a planning agency realizes and appreciates the value and benefit of a regional DTA model, continual efforts and resources may be planned and committed to regularly keep the DTA model up to date. This update can be put in place in conjunction with the regular update of the travel demand model.
- Model Consistency Checking: NDOT should continue the effort on the calibration of the existing models. Before a basic model can be calibrated or a future model can be used in an analysis, it is an essential component of the modeling to ensure that the model is consistent with the design year network conditions and is free from possible coding mistakes. Errors of omission and transposition of information are common occurrences.
- DTA Model Usage: When the model is readily built and calibrated against field data, it can be used for different types of predictive analysis. These analysis cases can include work zone analysis, bottleneck identification study, incident analysis, etc. However, there are some restrictions and considerations that should be carefully understood to obtain the best results from DTA models.

ACKNOWLEDGMENTS

We wish to gratefully acknowledge the contributions of several people who made the progress of this research possible. This especially includes Rui Yue, Mark Wooster, Randy Travis, Mizuho Wood, and Hoang Hong. Special thanks go to Mrs. Erika Hutton for her effort on proofreading the report.

GLOSSARY OF ACRONYMS

AADT: Annual Average Daily Traffic

ACO: Ant Colony Optimization

ADT: Average Daily Traffic

AMS: Analysis Modeling Simulation

ATIS: Automatic Traveler Information System

ATR: Automatic Traffic Recorder

BTDA: Bus Transit demand Assignment

CATER: Center for Advanced Transportation Education and Research

CCT: Consistent Cross-evaluation Tool

CSV: Comma-separated Values

DBF: dBASE Table File Format

DMS: Dynamic Message Sign

DNL: Dynamic Network Loading

DOT: Department of Transportation

DTA: Dynamic Traffic Assignment

DTALite: Light-weight Dynamic Traffic Assignment

DUE Dynamic User Equilibrium

DynusT: Dynamic Urban Systems for Transportation

ETT: Experience Travel Time

FHWA: Federal Highway Administration

GDAL: Geospatial Data Abstraction Library

GIS: Geographic Information System

GUI: Graphical User Interface

HCM: Highway Capacity Manual

HOT: High Occupancy Toll

HOV: High Occupancy Vehicle

ITE: Institute of Transportation Engineers

ITS: Intelligent Transportation System

ITT: Instantaneous Travel Time

LP: Linear Programming

MB: Breakdown Minimization

MOE: Measure of Effectiveness

MPO: Metropolitan Planning Organization

NDOT: Nevada Department of Transportation

NeXTA: Network eXplorer for Traffic Analysis

NPMRDS: National Performance Management Research Data Set

OD: Origin Destination

ODME: Origin-Destination Matrix Estimation

QEM: Quick Estimation Method

RTC: Regional Transportation Commission

SHP: Shapefile Shape Format

SO: System Optimum

SODTA: System Optimum Dynamic Traffic Assignment

SOV: Single Occupancy Vehicle

SPF: Safety Performance Function

STA: Static Traffic Assignment

TAZ: Traffic Area Zone

TDM: Travel demand Model

TIA: Traffic Impact Analysis

TMC: Traffic Management Channel

TNP: Transportation Network Project

TRINA: Traffic Record Information Access

UE: User Equilibrium

UNLV: University of Nevada, Las Vegas

UNR: University of Nevada Reno

UTDF: Universal Tracking Data Forma

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

Dynamic Traffic Assignment (DTA) is a method for estimation of some traffic performance measures. This chapter tries to explain why DTA can be used for this purpose. It also gives a short explanation of DTA and its difference with Static Traffic Assignment (STA) models.

1.1. Problem Description

An accurate prediction of traffic volumes such as the Annual Average Daily Traffic (AADT) serves as an important component in transportation planning and operations. The evaluation of many transportation projects, especially those with a regional scope, requires a clearer picture of how regional, interstate, primary, and secondary routes are used and to what extent through a methodology that combines available datasets with advanced modeling tools. Aiming to capture important spatial and temporal effects, Dynamic Traffic Assignment (DTA) models have been used in many transportation agencies due to the many advantages these models possess compared to macroscopic travel demand models and microscopic traffic simulation models. Current static statewide or regional models can help in planning and decision-making but they lack the level of details to analyze the temporal aspects of congestion. On the other hand, planning agencies are in need of conducting more detailed project-level analyses that require capturing spatial and temporal changes in traffic pattern such as congestion, queue buildup and dissipation. Therefore, representing user response to emerging strategies is becoming increasingly important for effective regional and statewide planning since it requires a time dimension and accurate AADT on different types of roads. DTA methodology can provide measures such as time dependent link volume, speed, density, queue length, and can track individual vehicles. NDOT has recently completed projects that involved development of DTA models for two major metropolitan areas in both southern and northern Nevada.

A DTA model was recently built for the Reno-Sparks areas [1] and a software tool was developed for RTC of Southern Nevada [2] that converts network and demand from TransCAD to VISSIM simulation, where a DTA network was created and then subareas were converted into VISSIM. These previous projects highlighted the fact that precision of a DTA model requires a significant amount of effort and depends largely on data availability. However, the calibration of the current DTA models for the two urban areas in Nevada are considered limited due to lack of detailed data and time constraints.

Recently, a simplified queue-based, capacity-constrained DTA methodology has been incorporated into DTALite that allows running large-scale regional/statewide transportation models with a 24-hour planning time period. DTALite was developed for the FHWA, which serves as the data hub for creating and running DTA models. Now, DTALite is able to address challenges introduced by the regional/statewide models such as the need to accommodate long distance trips that typically traverse multiple geographies (i.e. regions, states) and multiple time periods (i.e. peak and off-peak periods), use of less detailed demand and supply representation due to an almost prohibitively large network size. This project will continue to use a simplified simulation-based DTA method that can represent congestion dynamics, i.e. queue buildup and bottleneck formation, without the need for very detailed network coding and signal timing input.

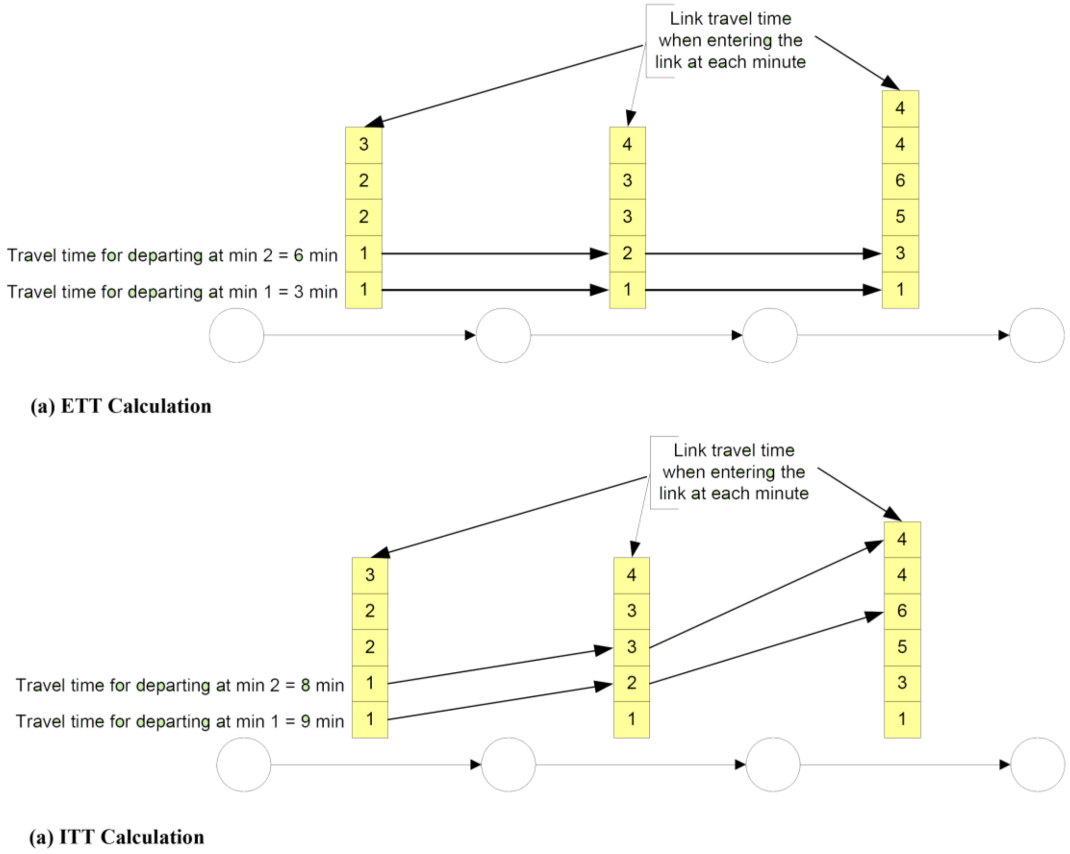
Instead of creating a statewide DTA model for Nevada, it is proposed to focus on a better calibration of the current DTA models for the two urban areas in Nevada. A full-scale fine-resolution statewide DTA model is not necessary based on the following reasons. First, the road network of Nevada consists of two urbanized regions, i.e. the Reno-Sparks network in northern Nevada and the Las Vegas network in southern Nevada, and several suburban regions. The routes to connect different regions in Nevada are not as complex as in other regions, and the inter-regional traffic does not show a significant increase to warrant dynamic traffic assignment between regions. Second, NDOT has already created DTA models for the two urbanized regions, but they are not readily calibrated for use in traffic planning analysis. Through this project, a well-calibrated DTA model was created that can be used to estimate traffic performance measures including AADTs, especially on streets where counts are not available for the purpose of safety studies such as SafetyAnalyst or Safety Performance Function (SPF). A case study was performed to demonstrate the applicability of DTA models such as work zone. The outcome of this research provides NDOT and other agencies in Nevada with some levels of confidence when deciding on using DTA models in planning and operational analyses.

1.2. Definition of DTA

DTA models offer dynamic network equilibrium modeling capability that is not available in STA and most microscopic traffic simulation models. The primary application areas for DTA models can be identified as operational planning and real-time operational control of vehicular traffic systems. Operational planning (or planning for operations) is aimed at making planning decisions for major operations, construction, or demand management actions that are likely to induce a temporal or spatial pattern shift of traffic among different roadway facilities at a corridor–network wide level. Such types of projects include but are not limited to (a) significant changes of roadway configuration (e.g., change downtown streets from one-way to two-way configuration), (b) freeway expansions, (c) construction of a city bypass, (d) adding or converting HOV–HOT lanes, (e) integrated freeway or highway corridor improvement–construction, and (f) travel demand management strategies such as peak spreading or congestion pricing [3].

Experienced travel time (ETT) plays a key role in establishing a dynamic equilibrium condition that is consistent with a traveler's route choice decision. The notion of experienced travel time departs from the notion of instantaneous travel time (ITT) that is typically applied in the static assignment context as well as in the micro-simulation context for one-shot assignment-simulation modeling. The example illustrated in Figure 1 shows the difference of ETT and ITT. This simple network has four nodes and three links. The stack of values represents the different times used to traverse a link when departing from the upstream node (and entering the link) at different times. Time-varying link travel time is common during peak hours due to congestion buildups. As an example, the time needed to traverse Link 1 is 1 time unit when departing the upstream node at time 1, and 3 time units when departing the upstream node at time 5. Similarly, the travel time for Link 2 is 1 and 2 time units when departing the Link 2 upstream node at times 1 and 2, respectively. The instantaneous travel time for the entire route at each departure time is calculated by summing up the link travel time corresponding to that same departure time for all links comprising the route. As an example, for vehicles departing at time 1, the travel time is 1 + 1 + 1 = 3 time units; for vehicle departing at time 2, the travel time is 1 + 2 + 3 = 6 time units [3].

The experienced travel time calculation accounts for the time needed for traversing one link, and looks up the downstream link travel time based on the time of entering that downstream link (assuming that traversing a node takes no time). Based on this approach, the travel time for the route when starting at Departure Time 1 should be 1 + 2 (vehicle entering Link 2 at Time 2 so the Link 2 travel time is 2 time units), plus 6 (vehicle arriving at Link 3 at Time 4, so the Link 3 travel time is 6 time units). The experienced travel time is 1 + 2 + 6 = 9 time units. Similarly for departure at Time 2, the instantaneous travel time is 6 time units and the experienced travel time is 8 time units.


Link travel time when entering the link at each minute
3
2
2
1
1
4
3
3
2
1
4
4
6
5
3
1
Travel time for departing at min 2 = 6 min
Travel time for departing at min 1 = 3 min
Link travel time when entering the link at each minute
Travel time for departing at min 2 = 8 min
Travel time for departing at min 1 = 9 min

(a) ETT Calculation

(a) ITT Calculation

Figure 1: Experienced travel time versus instantaneous travel time calculation [3]

Clearly, these two methods produce different route travel times and, likewise, differing results for the shortest route(s).

The shortest route obtained based on the instantaneous travel time calculation has the minimum travel time based on the snapshot of the link travel times prevailing at departure. However, because link travel times change dynamically (due to congestion), that route does not necessarily result in minimal experienced travel time because there is no provision in this procedure to reflect the anticipation of congestion that is to occur at a later time (e.g., congestion caused by vehicles departing later but entering the same link, one which the vehicle being modeled is still traversing) [3].

Assigning vehicles with an instantaneous travel time route is not necessarily incorrect, but its corresponding underpinning assumptions need to be understood. The route choice associated with instantaneous travel time may be interpreted as (a) travelers know what the shortest route is at departure through pre-trip information (e.g., 511, news, or website) or en route in-vehicle information system (if the traveler is to take another route when en route); or (b) from day to day, travelers do not assess the route travel time from the experience standpoint, but rely instead on the traveler information. In contrast, the shortest route obtained based on the experienced travel time calculation method will yield a time-dependent shortest route with minimal experienced travel time. This assumes that travelers are willing to seek routes that minimize their experienced travel time instead of the route that appears to be the best only at the departure time [3]. Obviously, with the prevalence of traveler information systems, one could argue that both types of travelers may co-exist in the traveler population. Some models do have the capability to include multiple user classes and present travelers with different information accessibility and route choice behavior to be jointly modeled, but the ATIS market penetration is likely to remain relatively low in Nevada, utilizing DTA models instead of STA models, wherever appropriate.

2. BACKGROUND SUMMARY

Though DTA is well known, it is not well understood [3]. This chapter tries to explore and show to what extent DTA studies have moved forward. First, the chapter shows the current practices and studies in Nevada; then performs a literature review.

2.1. Current Practice

NDOT has recently sponsored relevant research projects to develop and calibrate DTA models for both southern and northern Nevada regional networks. A University Transportation Center (UTC)-NDOT project developed a DTA model for the Reno-Sparks network in northern Nevada using NeXTA/DTALite [1]. It successfully demonstrated the conversion from an established TransCAD regional travel demand model into DTALite for mesoscopic analysis. Model calibration and validation against major roads link counts using AADTs was conducted, which demonstrated the capability of NeXTA/DTALite software package to better match link volume to observed data by adjusting OD matrices. Although the calibration process improved the initial DTA model to a certain extent, it was considered a limited one because the field data obtained was not sufficient enough to perform more calibration methods and hence produce a better-calibrated model.

Another research was conducted for the Regional Transportation Commission (RTC) of Southern Nevada [2]. In Phase 1 of this project, a software tool was developed to convert TransCAD models into VISSIM micro-simulations. In that process, NeXTA was used as an intermediate tool to convert a TransCAD model into a mesoscopic DTA model first, and then a Consistent Cross-resolution Tool (CCT) was developed to convert the DTA model into VISSIM for micro-simulation. The model was tested on the Las Vegas network. Phase 2 of this project was focused on model calibration through cross resolution analysis by making a subarea cut in the DTA model, adding signal timing into the subarea, and exporting it into Synchro and VISSIM for micro-simulation. The calibration requires a great amount of effort to produce a good DTA model for practical analysis.

The NDOT Traffic Operation Division sponsored a project at the University of Nevada, Las Vegas (UNLV) to develop a DTA model using DynusT for the Las Vegas metropolitan area. Because no final report was produced for the project, the details of their model development and calibration are not known.

2.2. Literature Review

Traditionally, transportation planners and analysts use Travel Demand Models (TDM) to forecast regional network conditions and evaluate the impacts of traffic planning projects. While a certain traffic pattern can be obtained through the static traffic assignment in the last step of the conventional four-step transportation planning process [4], it is limited in its ability to capture the detailed behavior of traffic. The traditional static techniques fail to incorporate the knowledge of non-stationary conditions on roadway segments, such as queue spillover, oversaturation and dynamic routing.

Microscopic simulation models are typically employed to represent a finer resolution of traffic dynamics by simulating the movement of individual vehicles [5]. They are usually applied to analyze various geometric design configurations and evaluate the impact of new traffic signal systems in a local area or corridor. However, microscopic simulations are limited to corridor level evaluations, lacking the regional level scope of travel behavior patterns.

Dynamic Traffic Assignment (DTA) models, on the other hand, provide more realistic traffic patterns by taking into considerations the time-varying traffic conditions such as queue formation, traffic signal timing, and route choice decisions. DTA overcomes the unrealistic assumptions of static models and incorporates the travel behavior information at a regional level [6]. Since DTA models are able to represent the interactions between travel choices and time-varying network conditions in a coherent manner, the results can serve as many meaningful measures to analyze travel time reliability, congestion and sensitivity analysis for region-wide planning and operational purposes.

Given the time-dependent characteristics of demand and traffic conditions over the network as input, DTA models seek time-space vehicular trajectories to achieve certain objectives. The typical objectives are to minimize total experienced travel costs or to achieve the converged traffic equilibrium patterns. In practice, this could be done by simulation-based DTA algorithms, which iteratively determine time-varying routes, link volumes and travel times in terms of so-called dynamic user equilibrium [3]. It typically requires extensive computation to complete the following three major procedures: path set update, routing choice algorithm, and traffic flow simulation. Although these simulation-based algorithms assign vehicles along paths according to the updated road segment information using a certain traffic flow model, their approaches differ from each other in many ways.

Merchant and Nemhauser formulated two of the early DTA models [7, 8]. One was a discrete time model and the other was a nonlinear and nonconvex mathematical program. Carey [9] also studied the nonconvexity of the dynamic traffic assignment problem. Other models were provided by numerous researchers; for example, Janson [10] presented a nonlinear programming formulation of the dynamic user-equilibrium assignment problem (DUE) for urban road networks with multiple trip origins and destinations; Ziliaskopoulos [11] used the Daganzo's cell transmission model concept [12, 13] to formulate the single destination System Optimum Dynamic Traffic Assignment (SODTA) problem as a Linear Program (LP); and Jayakrishnan et al [14], Carey [15], Smith [16], Lo [17], and Lam and Huang [18] provided additional models. Recently, Kerner [19] reviewed physical results of applications of the breakdown minimization (BM) principle versus applications of the classical Wardrop's equilibria (Wardrop's user equilibrium (UE) and system optimum (SO)) for dynamic traffic assignment and control in traffic and transportation networks. It is shown that depending on the total network inflow rate there are two different applications of the BM principle: (i) the network throughput maximization approach that maximizes the network throughput ensuring free flow conditions in the network; (ii) the minimization of the network breakdown probability at relatively large network inflow rates.

2.3. DTA Software Packages Review

The three major simulation-based DTA software packages that are widely used are DynusT, DTALite, and TransModeler.

DynusT (Dynamic Urban Systems for Transportation) is a package of open-source DTA software used for operational planning analysis [20]. It employs an iterative framework to perform traffic assignment for evaluating long-term impact of changed network conditions. DynusT uses the anisotropic mesoscopic simulation model, a modified version of Greenshield's model, to simulate the traffic flow propagation [21]. In this model, the vehicle speed depends upon the average density in a certain downstream area. Five user classes (unresponsive, system optimal, user equilibrium, en-route information, and pre-trip information) are defined in DynusT and are provided in proportion to the traffic stream in order to provide capabilities for various purposes in practice.

DTALite (Light-weight Dynamic Traffic Assignment Engine) is another widely adopted mesoscopic simulation-assignment software. It features a minimum requirement of data input from static traffic assignment data and some time-dependent OD demand estimates

[22]. Compared to other similar software packages, DTALite uses a built-in parallel computing technique to dramatically shorten the simulation and routing processes with widely available multi-core CPU hardware. Furthermore, its underlying traffic flow model is more theoretically rigorous. It embeds a simplified spatial queuing model, which uses Newell’s cumulative-count based traffic flow theory, to consider queue propagation and spillback [23]. This model is theoretically equivalent to widely used cell transmission models [11, 12]. The above features equip DTALite with the capability of both traffic realism and computational efficiency.

Both DynusT and DTALite share the same user-friendly front-end graphical user interface (GUI), NeXTA (Network eXplorer for Traffic Analysis). NeXTA is very convenient and flexible in practice. For data input, it allows users to create new networks, modify existing datasets, specify analysis scenarios and set simulation parameters. For output data, it allows to perform post-processing analysis and conduct analysis of the results from the simulation-based assignment [24]. Compared to other similar mesoscopic simulation software, these features in NeXTA make DynusT and DTALite more convenient for use in practice. In addition, these software packages are open source. Figure 1 shows a screen shot of the NeXTA environment.

Another powerful DTA software package is TransModeler, which has the ability for hybrid simulation (scalable micro, meso, and macro simulation) based on the availability of high fidelity data (Figure 2). Another feature of TransModeler is the ease of integration of travel demand and traffic modeling (Figure 3). Some other DTA software packages are: DynaMIT-R, DYNASMART-X, DynaMIT-P, DYNASMRAT-P, and Dynameq.

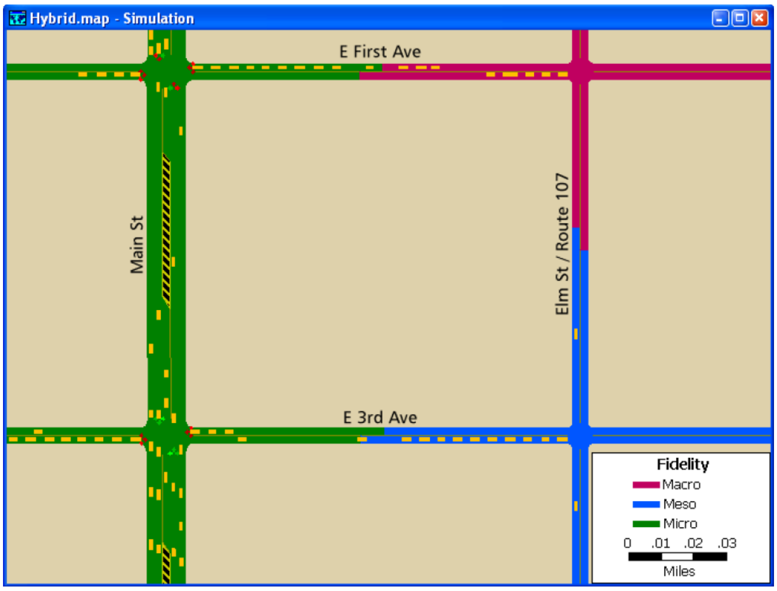
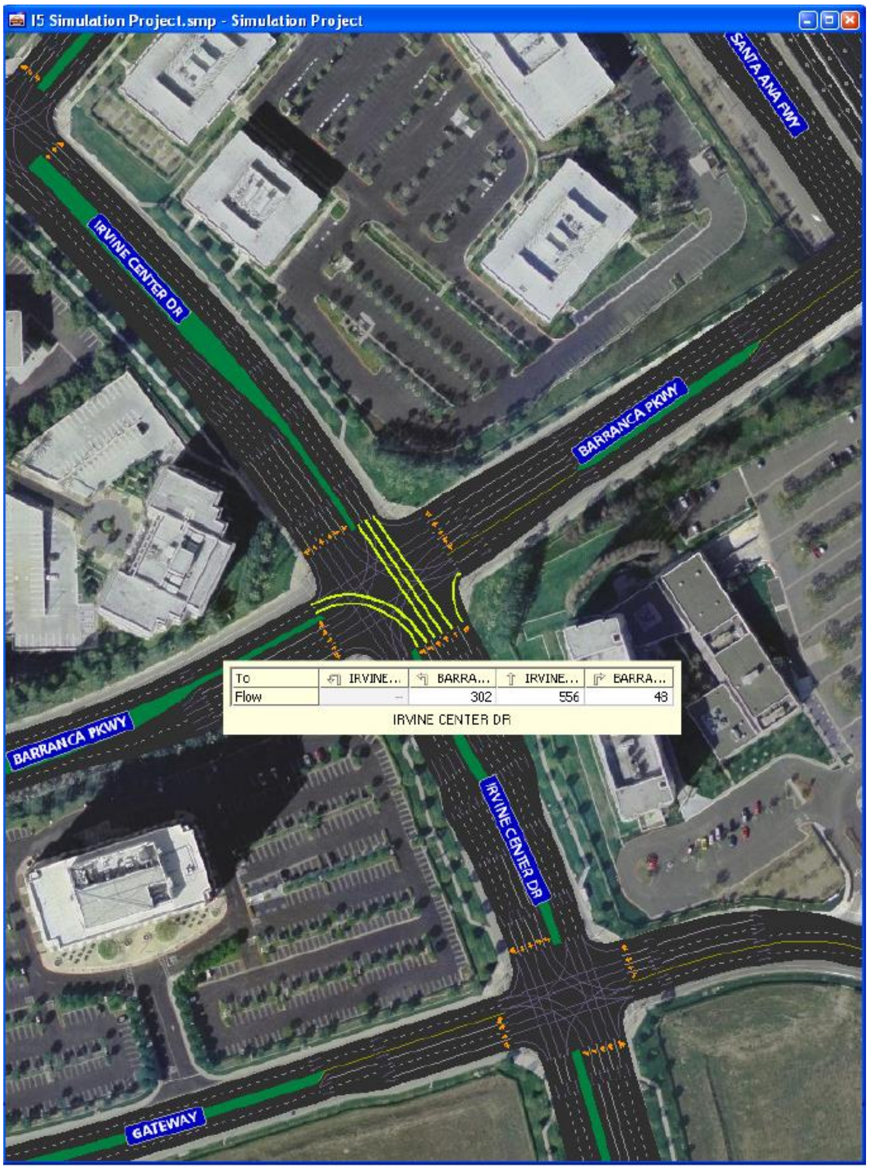


Figure 2: Hybrid simulation in TransModeler [25]



I5 Simulation Project.smp - Simulation Project

To	↰ IRVINE...	↰ BARRA...	↑ IRVINE...	↱ BARRA...
Flow	--	302	556	48

IRVINE CENTER DR

Figure 3: Integrated travel demand and traffic modeling in TransModeler [25]

Among these software packages, DynusT and DTALite, due to their flexibility and capability for transportation operational and planning purposes, are anticipated to be widely used in a variety of FHWA initiatives such as Connected Vehicles, integrated corridor management, and active transportation and demand management. There is no question that the application of DTA models to the region-wide network in Nevada is also beneficial in many ways to support the FHWA initiatives. In this project, the NeXTA/DTALite software package was used as the development tool to build a regional DTA model for the northern Nevada network and perform limited model calibration based on field data. It is evident that this project, which demonstrates the procedure of building and partially calibrating a regional DTA model, will directly contribute to advancing NDOT's traffic operational and planning programs while supporting FHWA initiatives.

Overall, developing a fully functioning micro- or meso-level statewide or regional model is a challenging task due to extensive data requirements and lack of expert knowledge in modeling. The scales of the models, coarser zone systems, databases that are geocoded to county or larger sized geographical areas, and inability to represent peak-hour travel with existing static assignment methods because of long intercity trip lengths are some of the ongoing problems identified in a comprehensive review report by Horowitz [26].

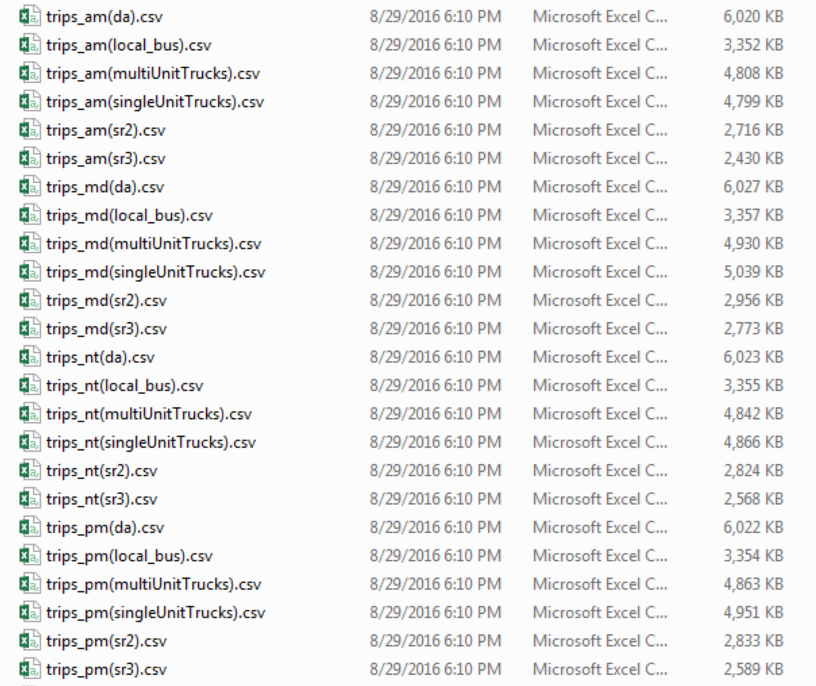
Mesoscopic simulation-based DTA models, on the other hand, provide more realistic traffic patterns by taking into considerations the time-varying traffic conditions such as queue formation, traffic signal timing, and route choice decisions [27]. DTA models combine most of the advantages of both microscopic and TDM models. DTA overcomes the unrealistic assumptions of static models and incorporates the travel behavior information at the regional level [6]. Since DTA models are able to represent the interactions between travel choices and time-varying network conditions in a coherent manner, the results can serve as many meaningful measures to analyze travel time reliability, congestion and sensitivity analysis for region-wide planning and operational purposes.

3. DATA SET PREPARATION

DTA models typically have numerous inputs and parameters that need to be specified before model application. The exact nature of these inputs and parameters depends largely on the individual components making up the DTA model. At a high level, however, they can be grouped into demand-side and network-side quantities. The demand quantities typically include, at a minimum, time-dependent O-D matrices or trip tables and traveler behavior model inputs and parameters. The network quantities include capacities, link performance functions, traffic control information and strategy information such as incident impact parameters or ITS elements [3].

3.1. Demand Data

Time-dependent trip tables are common demand inputs to DTA models, although some may also accept individual trip activity records (e.g., trip tour or trip chain). The patterns can vary across origins, destinations and departure times. The most common method for capturing these variations is through a series of trip tables, each containing information about the trip departures within a relatively short time interval. The duration of this interval depends on the variability of real-world demand patterns as well as the desired modeling accuracy. Figure 4 shows trip tables, which are available for Reno/Sparks area. Table 1 shows the meaning of abbreviations used for the name of these trip tables.



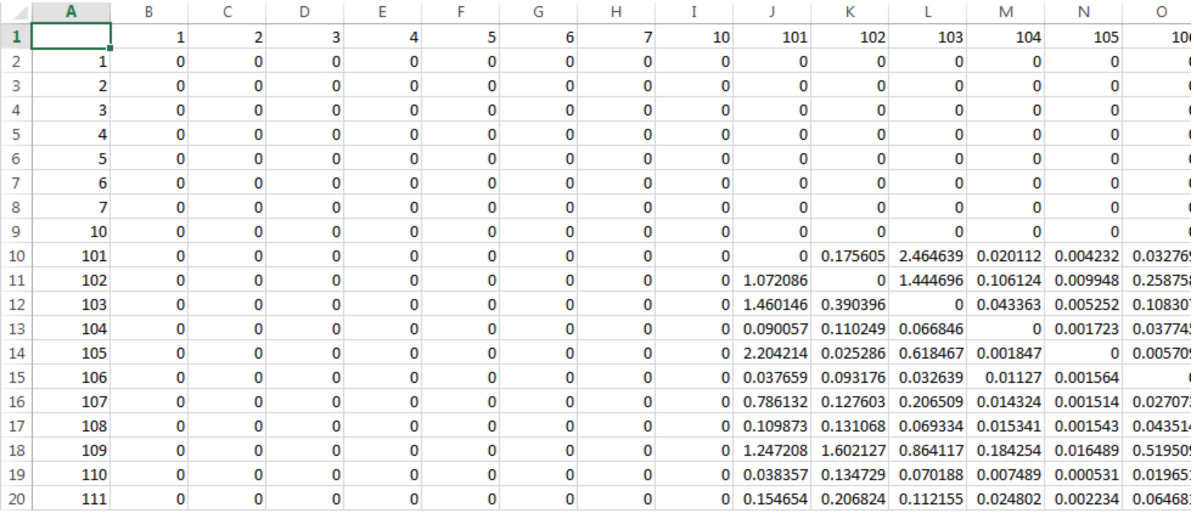
trips_am(da).csv	8/29/2016 6:10 PM	Microsoft Excel C...	6,020 KB
trips_am(local_bus).csv	8/29/2016 6:10 PM	Microsoft Excel C...	3,352 KB
trips_am(multiUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,808 KB
trips_am(singleUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,799 KB
trips_am(sr2).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,716 KB
trips_am(sr3).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,430 KB
trips_md(da).csv	8/29/2016 6:10 PM	Microsoft Excel C...	6,027 KB
trips_md(local_bus).csv	8/29/2016 6:10 PM	Microsoft Excel C...	3,357 KB
trips_md(multiUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,930 KB
trips_md(singleUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	5,039 KB
trips_md(sr2).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,956 KB
trips_md(sr3).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,773 KB
trips_nt(da).csv	8/29/2016 6:10 PM	Microsoft Excel C...	6,023 KB
trips_nt(local_bus).csv	8/29/2016 6:10 PM	Microsoft Excel C...	3,355 KB
trips_nt(multiUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,842 KB
trips_nt(singleUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,866 KB
trips_nt(sr2).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,824 KB
trips_nt(sr3).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,568 KB
trips_pm(da).csv	8/29/2016 6:10 PM	Microsoft Excel C...	6,022 KB
trips_pm(local_bus).csv	8/29/2016 6:10 PM	Microsoft Excel C...	3,354 KB
trips_pm(multiUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,863 KB
trips_pm(singleUnitTrucks).csv	8/29/2016 6:10 PM	Microsoft Excel C...	4,951 KB
trips_pm(sr2).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,833 KB
trips_pm(sr3).csv	8/29/2016 6:10 PM	Microsoft Excel C...	2,589 KB

Figure 4: Available trip data for Reno/Sparks area

Table 1: Abbreviations used for name of trip tables

Name	Meaning	Time of Day
am	a.m. peak period	6:00am – 9:00am
md	Midday period	9:00am – 4:00pm
pm	p.m. peak period	4:00pm – 7:00pm
nt	Night time	7:00pm – 6:00am
da	Drive Alone	-
sr2	Shared Ride with two persons	-
sr3	Shared Ride with three persons	-

The size of each trip table is determined by the number of zones. However, the number of O-D pairs (cells in each matrix) with positive flows between them is relatively small. This sparsity depends on the geographic distribution of the zones and the density of the transportation links connecting these zones. As an example, Figure 5 shows a part of the a.m. peak trip table for local buses. As it can be seen in this table, many cells are zero. The total number of O-D flows to be supplied to the DTA is thus a product of the number of nonzero cells of one matrix and the number of time intervals chosen for the modeling exercise. Typical O-D tables contain decimal-value entries (Figure 5). These values are eventually converted to integers during simulation since the basic entity in simulation is discrete (e.g., vehicle or person). This conversion step creates minor discrepancies between the number of vehicles generated between each O-D pair and the specified number of trips in the O-D tables.



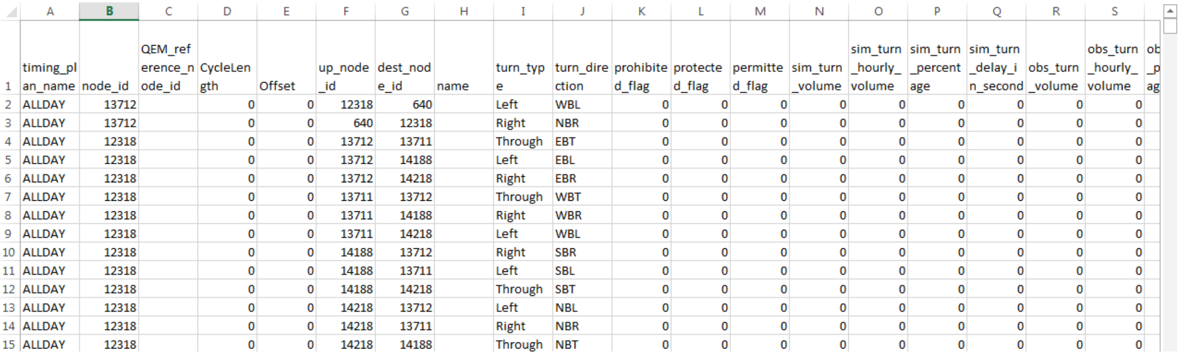
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		1	2	3	4	5	6	7	10	101	102	103	104	105	10
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	101	0	0	0	0	0	0	0	0	0	0.175605	2.464639	0.020112	0.004232	0.03276
11	102	0	0	0	0	0	0	0	0	1.072086	0	1.444696	0.106124	0.009948	0.25875
12	103	0	0	0	0	0	0	0	0	1.460146	0.390396	0	0.043363	0.005252	0.10830
13	104	0	0	0	0	0	0	0	0	0.090057	0.110249	0.066846	0	0.001723	0.03774
14	105	0	0	0	0	0	0	0	0	2.204214	0.025286	0.618467	0.001847	0	0.00570
15	106	0	0	0	0	0	0	0	0	0.037659	0.093176	0.032639	0.01127	0.001564	0
16	107	0	0	0	0	0	0	0	0	0.786132	0.127603	0.206509	0.014324	0.001514	0.02707
17	108	0	0	0	0	0	0	0	0	0.109873	0.131068	0.069334	0.015341	0.001543	0.04351
18	109	0	0	0	0	0	0	0	0	1.247208	1.602127	0.864117	0.184254	0.016489	0.51950
19	110	0	0	0	0	0	0	0	0	0.038357	0.134729	0.070188	0.007489	0.000531	0.01965
20	111	0	0	0	0	0	0	0	0	0.154654	0.206824	0.112155	0.024802	0.002234	0.06468

Figure 5: A part of the a.m. peak trip table for local buses

If a DTA model is developed from an existing static database, it may be desirable to use a finer zone structure implying a larger number of zones (especially if considering integrating the effort with activity-based models). This choice is motivated by the higher fidelity of DTA models in general and the way that zones are connected to the network. In a

DTA model, due to the higher realism of the representation of traffic flow, vehicles generally enter and exit the network using links that represent actual roads (rather than abstract centroid connectors), which helps to avoid creating false congestion points. Generally, DTA models can better portray the advantages of simulation by employing a higher resolution of the spatial distribution of demand and networks [3].

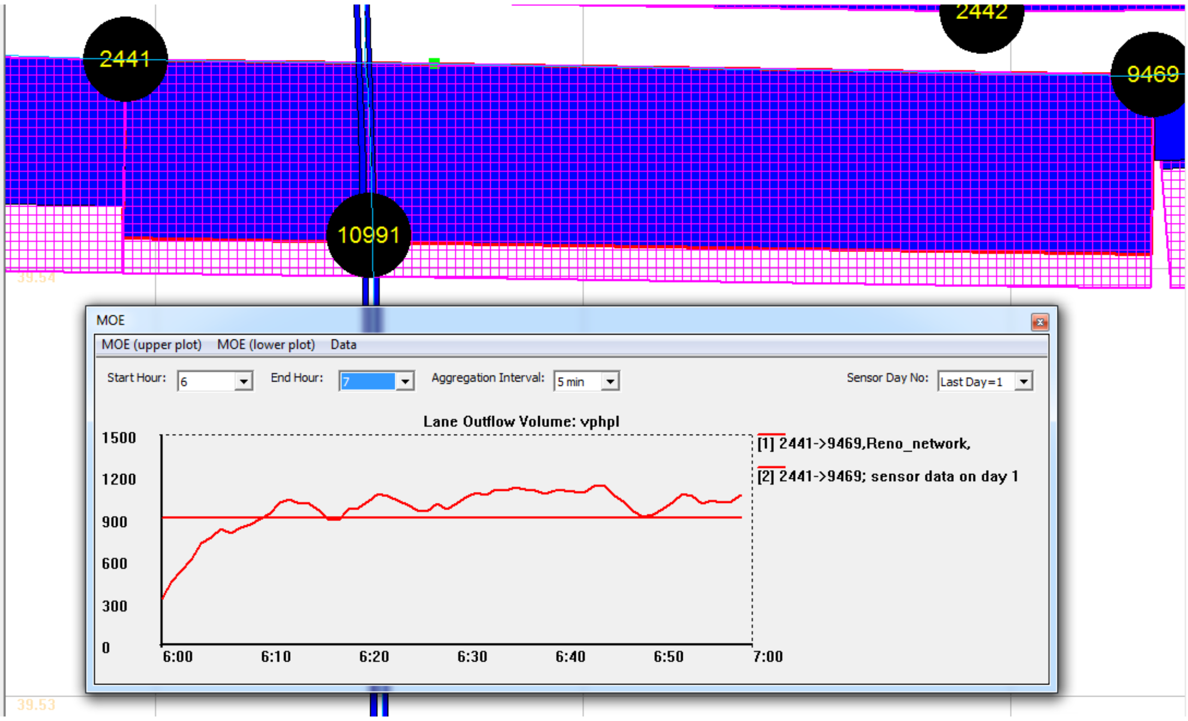
DTALite can simulate multiple vehicle types or classes. Separate classes should be used for the purposes of controlling access to different network elements (e.g., prohibited turns and special-use lanes). Figure 6 shows a part of file AMS_movement.csv that has the column prohibited_flag. Value of this column should be zero. As can be seen in Figure 4, every class has its own demand matrices. Typical classes might be single-occupancy vehicle, HOV, light truck, heavy truck, taxi, etc.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	timing_plan_name	node_id	QEM_reference_node_id	CycleLength	Offset	up_node_id	dest_node_id	name	turn_type	turn_direction	prohibited_flag	protected_flag	permitted_flag	sim_turn_volume	sim_turn_hourly_volume	sim_turn_percentage	sim_turn_delay_in_second	obs_turn_volume	obs_turn_hourly_volume	ob_pag
2	ALLDAY	13712		0	0	12318	640		Left	WBL	0	0	0	0	0	0	0	0	0	
3	ALLDAY	13712		0	0	640	12318		Right	NBR	0	0	0	0	0	0	0	0	0	
4	ALLDAY	12318		0	0	13712	13711		Through	EBT	0	0	0	0	0	0	0	0	0	
5	ALLDAY	12318		0	0	13712	14188		Left	EBL	0	0	0	0	0	0	0	0	0	
6	ALLDAY	12318		0	0	13712	14218		Right	EBR	0	0	0	0	0	0	0	0	0	
7	ALLDAY	12318		0	0	13711	13712		Through	WBT	0	0	0	0	0	0	0	0	0	
8	ALLDAY	12318		0	0	13711	14188		Right	WBR	0	0	0	0	0	0	0	0	0	
9	ALLDAY	12318		0	0	13711	14218		Left	WBL	0	0	0	0	0	0	0	0	0	
10	ALLDAY	12318		0	0	14188	13712		Right	SBR	0	0	0	0	0	0	0	0	0	
11	ALLDAY	12318		0	0	14188	13711		Left	SBL	0	0	0	0	0	0	0	0	0	
12	ALLDAY	12318		0	0	14188	14218		Through	SBT	0	0	0	0	0	0	0	0	0	
13	ALLDAY	12318		0	0	14218	13712		Left	NBL	0	0	0	0	0	0	0	0	0	
14	ALLDAY	12318		0	0	14218	13711		Right	NBR	0	0	0	0	0	0	0	0	0	
15	ALLDAY	12318		0	0	14218	14188		Through	NBT	0	0	0	0	0	0	0	0	0	

Figure 6: Columns turn_direction and prohibited_flag in file AMS_movement.csv

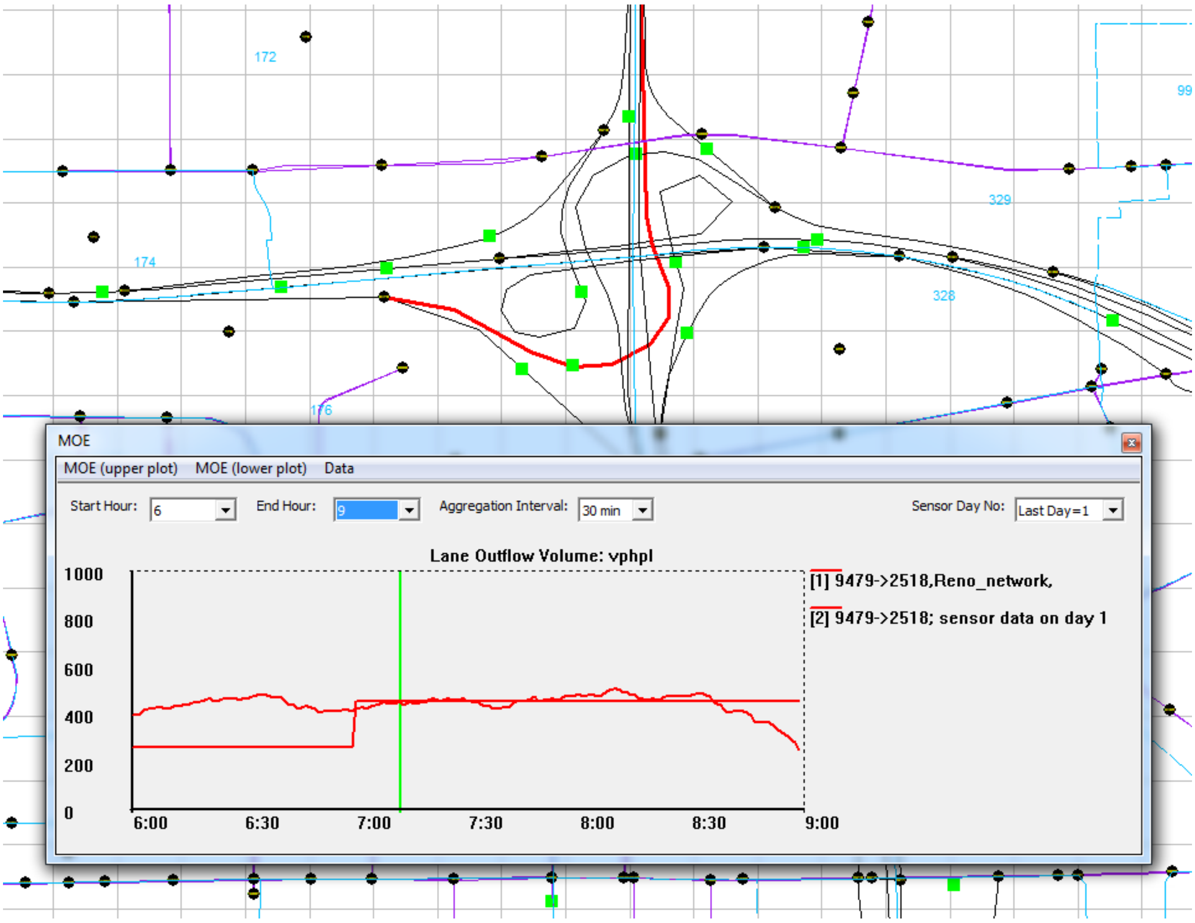
The solution of a dynamic model starts with an empty network, accumulates vehicles according to the input demand rates, and gradually empties out during low-demand intervals. Because it takes time to load the network, it is important that the demand period start earlier than the time window (observation period) over which the model is expected to produce meaningful results regarding the state of traffic. For example, Figure 7 shows that for the path from Node 2441 to Node 9469, it took 10 minutes from 6 a.m. for system warmup. Therefore, since the a.m. peak is from 6 a.m., the simulation should start at least 15 minutes before 6 a.m. to get reasonable results. It is also useful to end the demand period beyond the desired study end period to allow all (or most) vehicles to load during the actual study period and clear the network. Having the demand period longer than the study period will produce more realistic results, as the process of the network clearing (which will experience lower travel times due to lack of demand) will not significantly affect route choices of travelers during the study period. As a general rule, the start and end buffer windows should be as long as it takes to get across the network (longest trip) under the prevailing traffic conditions. Note that the end buffer window should still contain realistic demand rates so that the vehicles already on the network experience reasonable travel times and make route choices accordingly. Using a zero-demand end buffer window can bias the speeds upwards and travel times downwards, resulting in inaccurate route choices. Furthermore, at the end of simulation, certain vehicles may still exist in the network and have not completed the trip. It is important to understand how these vehicles are dealt with in the model output statistics to avoid biased statistics (e.g., incorporating incomplete trips may result in shorter distance and travel time) [3].



Start Hour: 6
End Hour: 7
Aggregation Interval: 5 min
Sensor Day No: Last Day=1

Figure 7: System warmup from 6 a.m. to 6:10 a.m.

Time-varying demand data may be derived from several sources. The most convenient way is to utilize the existing trip tables associated with travel forecasting models. Now, RTC has O-D tables for different periods in a day (Figure 4), with each table spanning several hours. If hourly factors are available for the time of interest, they can be used to derive a temporal profile in order to disaggregate the existing tables into finer time resolutions (e.g., hourly or 15-min tables). However, one should be warned that simply applying the hourly factors to a 24-hour table to derive the hourly table is a flawed exercise, as the directionality of O-D trips are typically lost when trips are aggregated into the 24-hour table. Factoring a 24-hour table does not retrieve the critical O-D directionality information. The travel pattern would deviate a great deal from reality on the ground [3]. The RTC trip tables represent a.m. peak, p.m. peak, or off-peak periods. To some extent the directionality is preserved in these time-of-day tables compared to the 24-hour table. If DTA is applied only for peak-hour analysis, then a corresponding time-of-day table can be a reasonable starting point. The temporal profile within the period of interest may still need to be specified by the model user. The RTC trip tables are aggregated based on one-hour intervals that can lead to some significant differences between simulation results and sensor data. Figures 8 to 10 show some samples from the Reno/Sparks network that illustrate differences of one-hour aggregated sensor data and simulation volumes that are based on RTC a.m. peak hour volumes. Chapter 5 will explain how to adjust this peak hour data to match better with sensor data. For a 24-hour simulation and assignment, one may consider combining these time-of-day tables to form 24-hour demand tables.



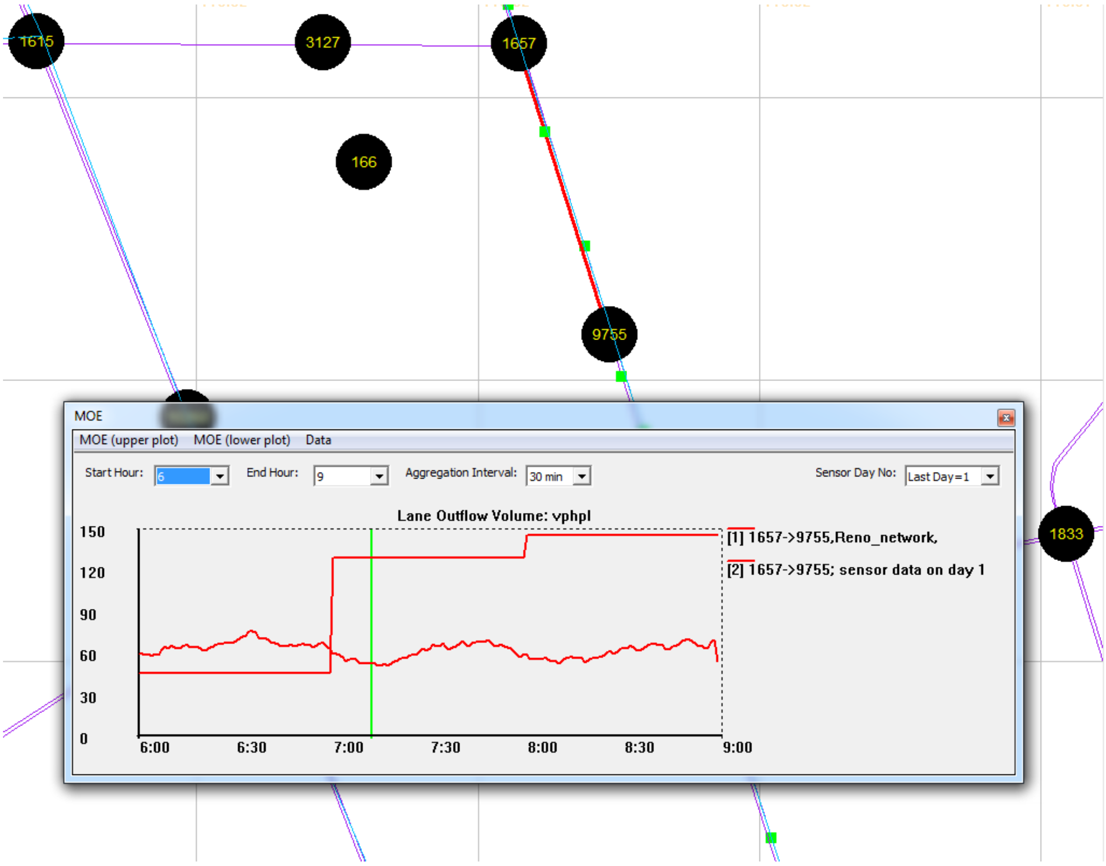
Start Hour: 6

End Hour: 9

Aggregation Interval: 30 min

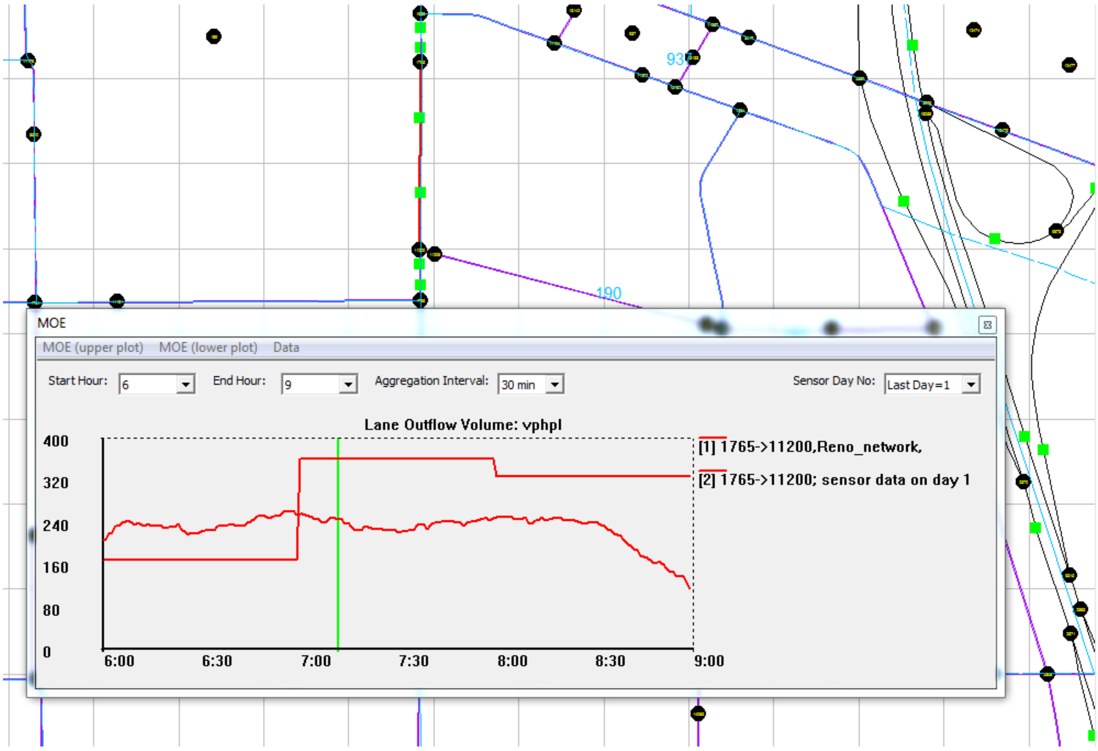
Sensor Day No: Last Day=1

Figure 8: Difference of Sensor count vs simulation volume at link 9479-2518 during a.m. peak



Start Hour: 6 End Hour: 9 Aggregation Interval: 30 min Sensor Day No: Last Day=1

Figure 9: Difference of Sensor count vs simulation volume at link 1657-9755 during a.m. peak


93
190
MOE
MOE (upper plot)
MOE (lower plot)
Data
Lane Outflow Volume: vphpl
400
320
240
160
80
0
6:00
6:30
7:00
7:30
8:00
8:30
9:00
[1] 1765->11200,Reno_network,
[2] 1765->11200; sensor data on day 1

Start Hour: 6 End Hour: 9 Aggregation Interval: 30 min Sensor Day No: Last Day=1

Figure 10: Difference of Sensor count vs simulation volume at link 1765-11200 during a.m. peak

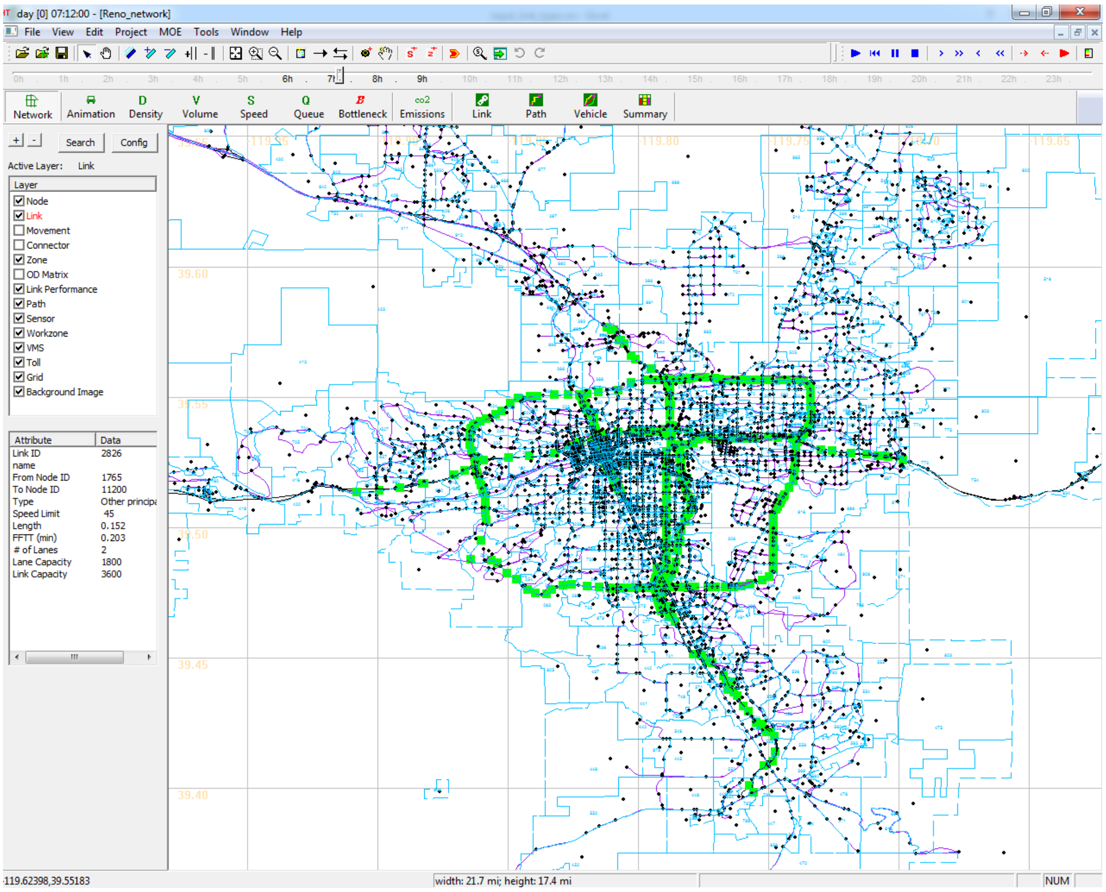
In summary, the time-dependent demand data that was imported into DTALite are in the format of an OD matrix, which covers a 24-hour trip demand for different vehicle types and with different peak and off-peak periods. Demand types considered in the demand files contain Single Occupancy Vehicle (SOV) passenger car, 2-person High Occupancy Vehicle (HOV2), HOV3, single-unit truck, multi-unit truck and local bus. The demand data covers 24 hours of an entire day. A 24-hour day is then divided into four time periods, i.e. a.m., md, p.m., and night.

3.2. Network Data

The Regional Transportation Committee (RTC) of Washoe County maintains the regional travel demand model in Caliper TransCAD®$^{®}$. This network was converted to a set of shapefiles before being imported into NeXTA. This was accomplished using the export tool in TransCAD®. The full network is imported into NeXTA to retain accuracy. A screen shot of the network with TAZs (Traffic Area Zone) in NeXTA is presented in Figure 11 and an overview of the Reno-Sparks network data is provided in Table 2.

DTA models are generally more data-intensive than static models. For example, though both models work on a network of the study region, DTA requires a more detailed network including the number of lanes in each link, the presence of acceleration–deceleration lanes, and turn bays and lane connectivity. Such data must be collated from various sources. The network representation, for example, could be based on existing models, geographic information system (GIS) files, online maps or aerial photographs. In the absence of a network already coded in the format required by the chosen DTA software, the model user must create such a network from scratch. Most existing DTA models provide GUI (Graphical User Interface) for this purpose.

GIS files can considerably simplify this step, as the centerlines and other geometry information from such sources can be expected to be reasonably accurate. Section 4.1 will explain how to import GIS files. Additional work may be involved in defining all allowed and prohibited lane movements at link and segment boundaries. Online maps and aerial photographs can be invaluable sources of data at this stage, especially for validating lane connections. Existing data sets will most probably be derived from static planning models. The network from such a data set must be upgraded to include at least the basic DTA requirements. Such an upgrade can be time-consuming, depending on the spatial extent and density of the network and the level of detail in the static network representation. The model user should further remember that static networks are often based on nodes connected by straight-line segments. A move to a more accurate geography will ensure better results and output visualization that will be true to the real-world transportation system.


day [0] 07:12:00 - [Reno_network]
File View Edit Project MOE Tools Window Help
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
Network Animation Density Volume Speed Queue Bottleneck Emissions Link Path Vehicle Summary
+ - Search Config
Layer
39.60
39.55
39.50
39.45
39.40
NUM

Active Layer: Link

☑ Node
☑ Link
□ Movement
□ Connector
☑ Zone
□ OD Matrix
☑ Link Performance
☑ Path
☑ Sensor
☑ Workzone
☑ VMS
☑ Toll
☑ Grid
☑ Background Image

Attribute	Data
Link ID	2826
name	
From Node ID	1765
To Node ID	11200
Type	Other principa
Speed Limit	45
Length	0.152
FFTT (min)	0.203
# of Lanes	2
Lane Capacity	1800
Link Capacity	3600

-119.62398,39.55183

width: 21.7 mi; height: 17.4 mi

Figure 11: Screen shot of Reno-Sparks network with TAZs in NeXTA

Table 2: Reno/Sparks network data overview

Name	Value
Nodes	4533
Links	11919
Zones	904
Activity Locations	904
Link Types	10
Sensor Records	1692

3.3. Control Data

Coding signal timing and ramp meter control increase the fidelity of the network for both the simulation and assignment standpoints. Properly coded signals realistically represent the delays at surface streets and sometimes lead to more realistic assignment results. Some

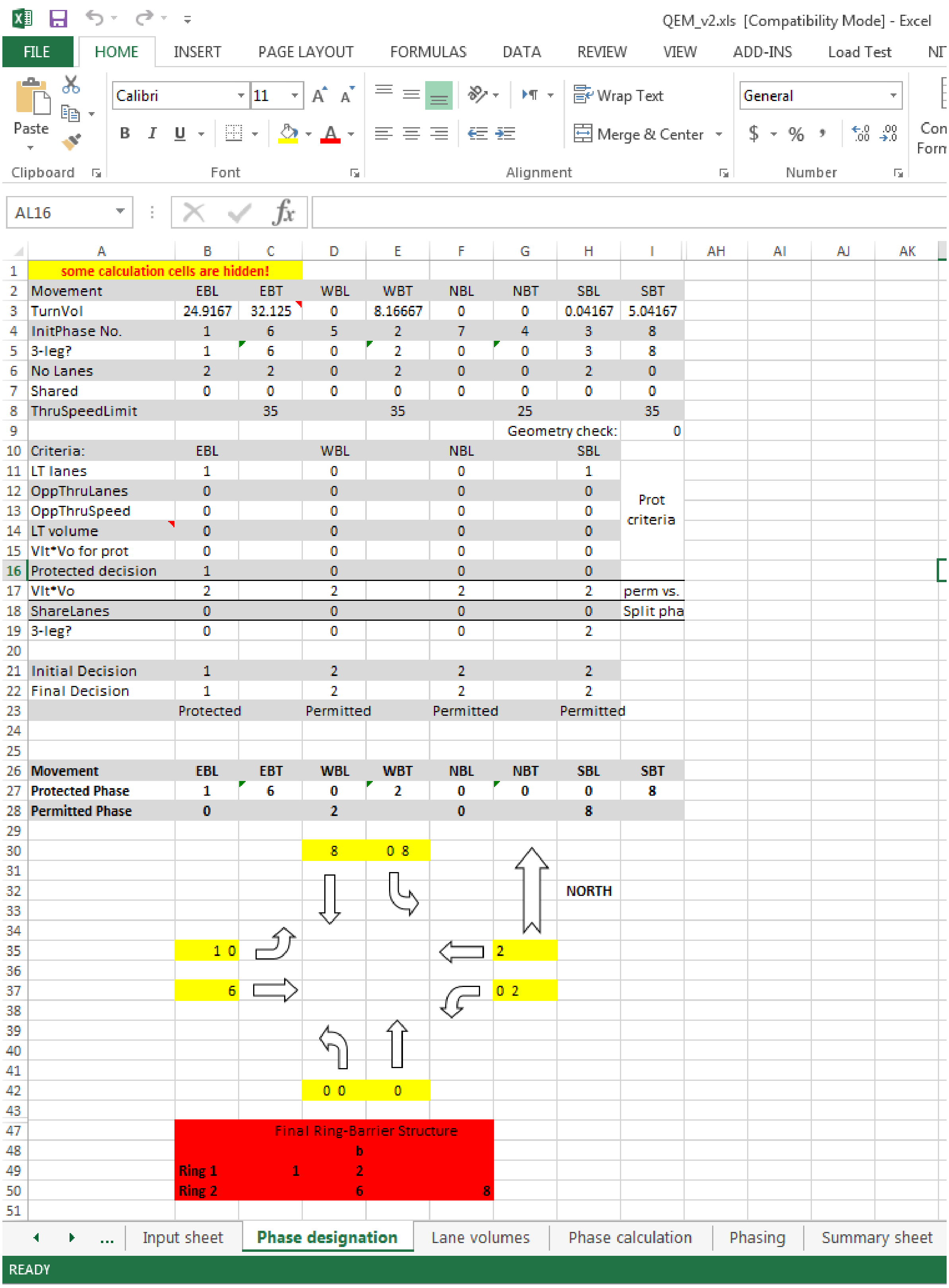
DTA models can typically represent the operation of pre-timed (fixed) traffic signals, potentially with the necessary parameters to represent signal synchronization. Various software systems may have different options available for representing actuated signals, though these options normally involve a set of parameters such as minimum and maximum green times per phase, which are used to approximate the operation of this type of controller in a simplified way [3].

Some DTA software approximates actuated signals using fixed-time plans by obtaining average green splits per movement from a signal timing–optimization software package. Some DTA software allows analysts to directly enter actual timing including actuated control so that actuated controls are more explicitly represented. Often, however, the detailed logic of modern traffic controllers is not modeled, but approximated.

DTALite currently provides for the specification of standard uncontrolled intersections such as all-way stop controlled, two-way stop controlled, roundabouts, freeway merges and yield signs.

Another commonly raised question is about the signal timing setting for existing intersections or for those existing only in future planning years. One suggestion is to leave the existing time unchanged and apply actuated timing for future intersections; then, conduct DTA test runs and identify locations where sustained congestion exists.

Currently there is a signal Quick Estimation Method (QEM) spreadsheet that can be used to generate initial signal phasing and timing for the subarea network (Figure 12). NeXTA writes the geometry and volume information to the spreadsheet, the spreadsheet calculates the appropriate phasing and timing data, which can then be used in NeXTA.



	A	B	C	D	E	F	G	H	I	AH	AI	AJ	AK
1	some calculation cells are hidden!												
2	Movement	EBL	EBT	WBL	WBT	NBL	NBT	SBL	SBT				
3	TurnVol	24.9167	32.125	0	8.16667	0	0	0.04167	5.04167				
4	InitPhase No.	1	6	5	2	7	4	3	8				
5	3-leg?	1	6	0	2	0	0	3	8				
6	No Lanes	2	2	0	2	0	0	2	0				
7	Shared	0	0	0	0	0	0	0	0				
8	ThruSpeedLimit		35		35		25		35				
9								Geometry check:	0				
10	Criteria:	EBL		WBL		NBL		SBL					
11	LT lanes	1		0		0		1					
12	OppThruLanes	0		0		0		0	Prot criteria				
13	OppThruSpeed	0		0		0		0					
14	LT volume	0		0		0		0					
15	Vlt*Vo for prot	0		0		0		0					
16	Protected decision	1		0		0		0					
17	Vlt*Vo	2		2		2		2	perm vs.				
18	ShareLanes	0		0		0		0	Split pha				
19	3-leg?	0		0		0		2					
20													
21	Initial Decision	1		2		2		2					
22	Final Decision	1		2		2		2					
23		Protected		Permitted		Permitted		Permitted					
24													
25													
26	**Movement**	**EBL**	**EBT**	**WBL**	**WBT**	**NBL**	**NBT**	**SBL**	**SBT**				
27	**Protected Phase**	**1**	**6**	**0**	**2**	**0**	**0**	**0**	**8**				
28	**Permitted Phase**	**0**		**2**		**0**		**8**					
29													
30				8	0 8								
31													
32								NORTH					
33													
34													
35		1 0					2						
36													
37		6					0 2						
38													
39													
40													
41													
42				0 0	0								
43													
47			Final Ring-Barrier Structure										
48				b									
49		Ring 1	1	2									
50		Ring 2		6		8							
51													

Figure 12: QEM Spreadsheet

QEM application consists of two files, *QEM_Signal_Hub_ver2.xlsm* and *QEM_v2.xls*. They should be both placed in the subarea project folder.

In order to estimate initial signals for the subarea, the user should open file *QEM_Signal_Hub_ver2.xlsm*, and click on the button shown in Figure 13 to run QEM.

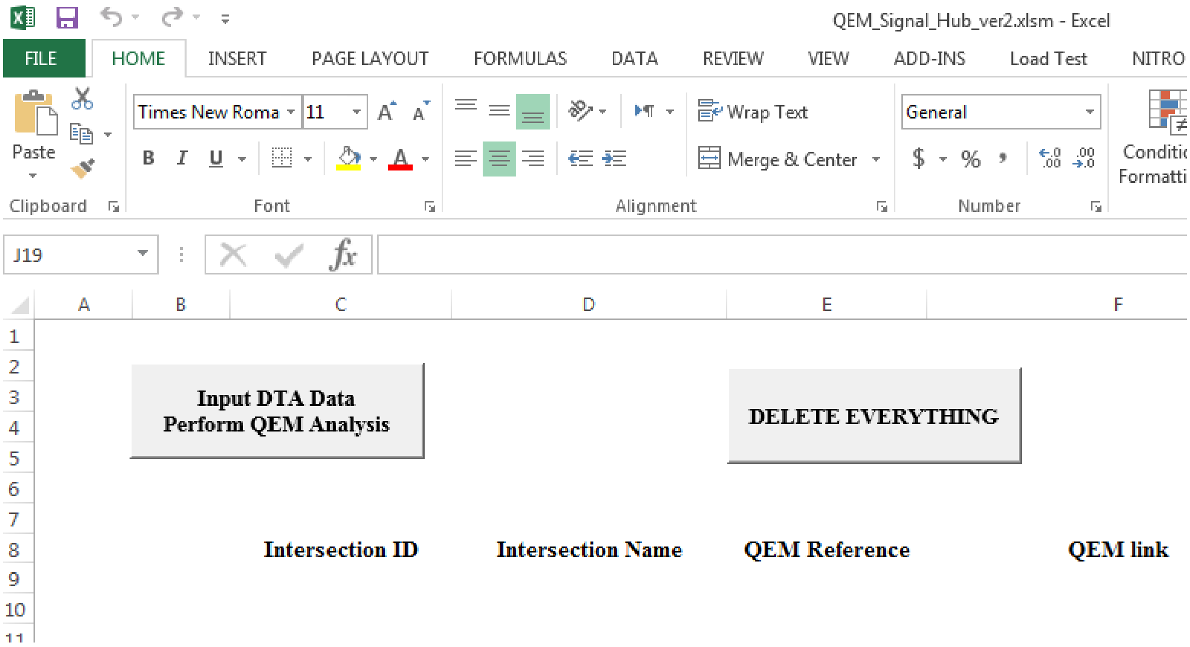


Figure 13: Run Quick Estimation Method (QEM) to generate initial signals

All the result files that QEM generates are located in the same project folder of the subarea.

Another detail that can be added to intersections is intersection links. Since the Reno/Sparks network was converted from TransCAD, intersections are represented with one node. Therefore, it is not possible to apply a high-fidelity signal control mechanism on such network structure. One may use MetroSim [29] to integrate the high-fidelity signal control mechanism in DTALite. Figure 14 represents the MetroSim architecture for making an intersection with its entire turning movement links. Figure 15 shows an intersection that is coded with all turning links.

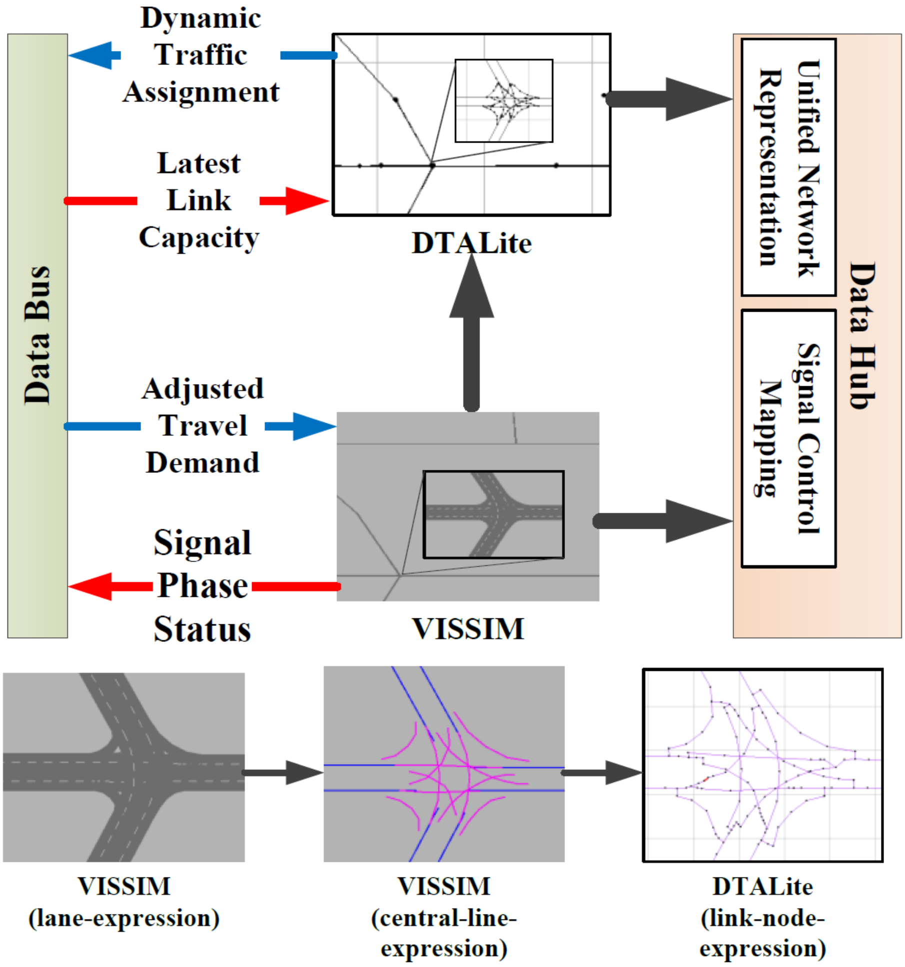
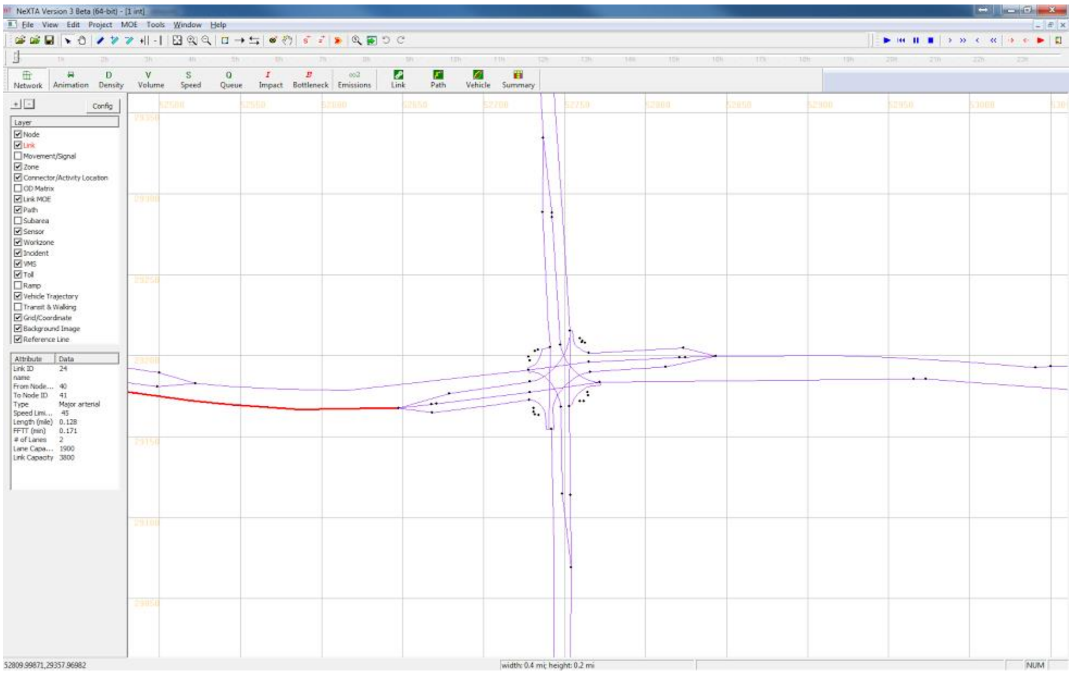


Figure 14: System Architecture of MetroSim and Network Representation [29]


NeXTA Version 3 Beta (64-bit) - [1 int]
File View Edit Project MOE Tools Window Help
Network Animation Density Volume Speed Queue Impact Bottleneck Emissions Link Path Vehicle Summary
Config

Layer
☑ Node
☑ Link
□ Movement/Signal
☑ Zone
☑ Connector/Activity Location
□ OD Matrix
☑ Link MOE
☑ Path
□ Subarea
☑ Sensor
☑ Workzone
☑ Incident
☑ VMS
☑ Toll
□ Ramp
☑ Vehicle Trajectory
□ Transit & Walking
☑ Grid/Coordinate
☑ Background Image
☑ Reference Line

Attribute	Data
Link ID	24
name	
From Node...	40
To Node ID	41
Type	Major arterial
Speed Limi...	45
Length (mile)	0.128
FFTT (min)	0.171
# of Lanes	2
Lane Capa...	1900
Link Capacity	3800

52809.99871,29357.96982

width: 0.4 mi; height: 0.2 mi

NUM

Figure 15: Intersection links

However, it is not currently possible to replace the intersections with a high-fidelity network in NeXTA. One needs to draw intersections by hand within the existing Reno/Sparks regional network in NeXTA. Due to practical issues, this was not performed in this project. Since DTALite uses the first-in first-out rule, some links experience unrealistic congestion after adding intersection links. For example, suppose right turn is congested. Since any vehicle at this approach that has entered sooner should exit first, other approaches also experience unrealistic congestion.

3.4. Scenario Data

Once the O-D demand, network, traffic control, transit line, and scenario data are properly coded in the correct format, the next step is model calibration and validation. This involves identification of all other DTA model inputs and parameters such as link or segment traffic flow parameters, capacities or performance functions, O-D adjustments, route choice model parameters, and historical (or perceived) network travel times. Scenario data usually relate to the application of interest. Properly specifying scenario data is crucial for using the model properly for the problem at hand [3].

4. PROCEDURE FOR MODELING WITH NeXTA/DTALite

This chapter provides the procedure for making the basic DTA model in NeXTA. First, it explains how to prepare GIS files and then describes the model building steps in NeXTA. The validation and modification of this model will be explained in Chapter 5.

The procedure for making a DTA model in NeXTA/DTALite is as follows:

Step 1: Export network and demand files from TransCAD® regional travel demand model;

Step 2: Use NeXTA's import network tool to import the GIS network;

Step 3: Import demand data into NeXTA from the regional travel demand model;

Step 4: Run assignment with DTALite to equilibrium;

Step 5: Change configurations in input_scenarios_settings.csv file;

Step 6: Prepare field data for ODME (Origin Destination Matrix Estimation);

Step 7: Run ODME using field data for calibration of the whole network;

Step 1 is the basic process of data preparation before running dynamic traffic assignment. Steps 2 and 3 conduct network and demand data conversion from the regional travel demand model to a DTA-compatible network. These three steps are explained in Section 4.1 using NeXTA_for_GIS in a general way. In Sections 4.2 to 4.4, these steps are specifically explained for this project. For more details regarding these three steps, one can refer to information provided in Section 4.1. In Step 4, the simulation runs to reach an initial DTA model according to the network profile and travel demand data. Steps 5 to 7 are the procedure for initial calibration of the DTA model using field data (in this case, link volume data). After running ODME, the model should be validated and calibrated again. This process will be explained in the next chapter.

4.1. Preparing and Importing GIS Files

The following sections describe techniques and processes for importing network data using NeXTA_for_GIS. Specifically, these tools provide the capabilities of preparing and importing shapefiles from VISUM, Cube, TransCAD and other shapefiles.

First, download the GIS Network data into the NeXTA using the following link:

https://github.com/xzhou99/learning-transportation/blob/master/Lessons/Lesson%201/Lesson%201.3/GIS_Import_Export_Tool/GIS-Import_Export_Tool.zip

Unzip the file to a known location on your computer, and open the NeXTA_for_GIS.exe file.

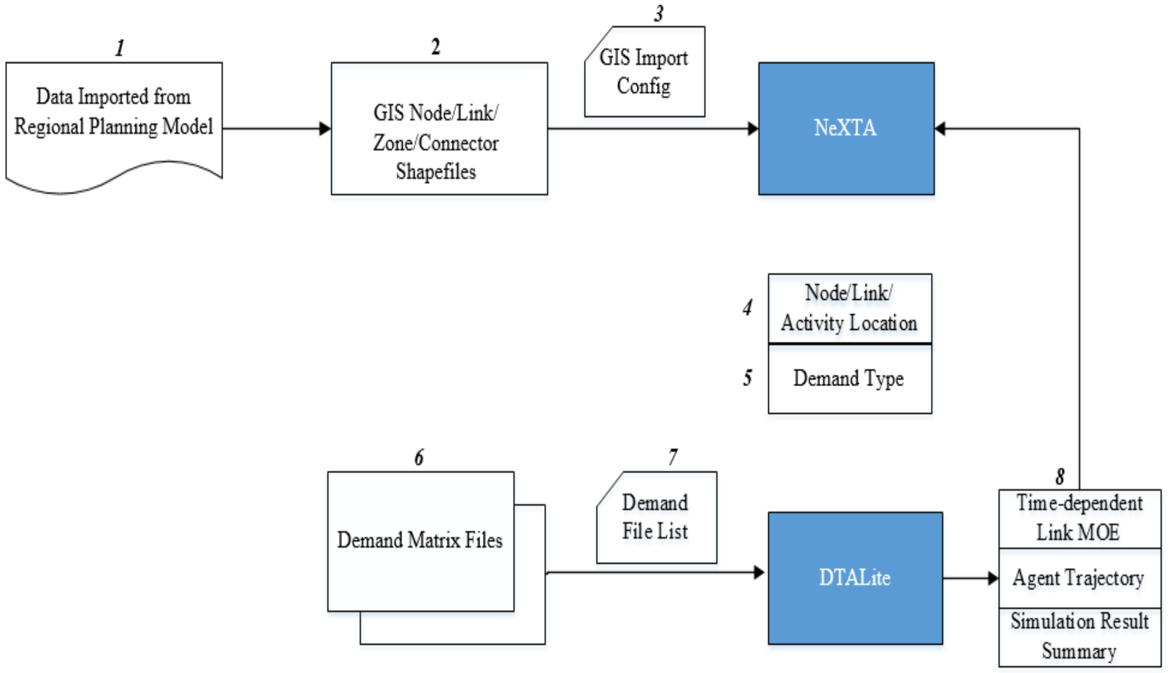

1
Data Imported from Regional Planning Model
2
GIS Node/Link/ Zone/Connector Shapefiles
3
GIS Import Config
NeXTA
4
Node/Link/ Activity Location
5
Demand Type
6
Demand Matrix Files
7
Demand File List
DTALite
8
Time-dependent Link MOE
Agent Trajectory
Simulation Result Summary

Figure 16: Data flow for importing GIS network to NeXTA_for_GIS

NeXTA_for_GIS network conversion tool specifically imports network data from shapefiles, which are geospatial vector data files used with GIS software and commonly supported by many transportation modeling software packages. Shapefiles contain spatial information about points, lines, and polygons in the transportation network, with separate files for different shapes. Road networks are represented as a graph of links (lines) and nodes (points), where links may represent road segments or transit lines and nodes may represent intersections or connections between individual links, and trips are often made between zones (polygons) on the network. The node and link layers can use an arbitrary coordinate system, but a WGS 84 (long/lat) coordinate system is preferred to export data in Google Earth/ Google Map.

Network attribute data is stored in database (DBF) files for each shapefile, which must be read by NeXTA during the conversion process. Network data formatting is very flexible between network modeling software packages, with many applications allowing users to define custom formats and data fields for use within a software package. To support these applications, NeXTA_for_GIS uses a configuration CSV file for importing GIS settings, to identify and connect the fields in the input DBF files to their corresponding fields in the NeXTA data format (import_GIS_settings.csv).

The following sections describe processes for preparing network data using NeXTA's network conversion utilities. A network data set for the NeXTA package includes four major CSV files: input_node.csv, input_link.csv, input_zone.csv, input_activity_location.csv, as well as a number of definitional files such as input_node_control_type.csv and input_link_type.csv. There are multiple ways of constructing a network data set for the NeXTA_for_GIS package:

4.1.1. Step 1: Preparing required shapefiles

Obtain shapefiles from regional planning packages such as VISUM, TransCAD, or Cube, which typically consist of node, link and traffic analysis zone layers, with or without centroid and connector layers. Zone shapefile is optional. Sample GIS files can be accessed in the GIS_Import_Export_Tool folder from downloaded package.

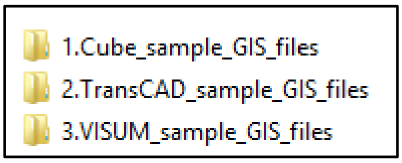

1.Cube_sample_GIS_files
2.TransCAD_sample_GIS_files
3.VISUM_sample_GIS_files

Figure 17: Sample GIS Files

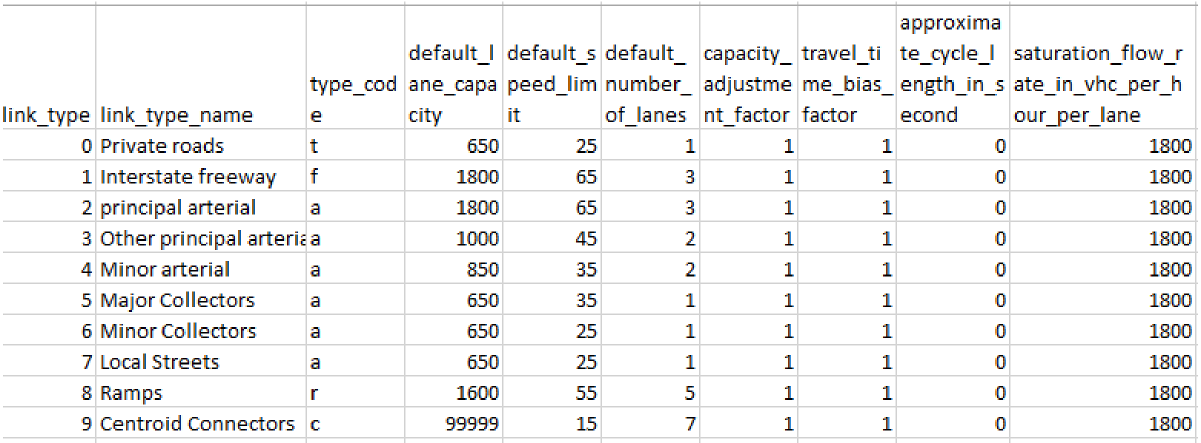
Another option is to obtain shapefiles from GIS shapefiles with a link layer only, such as an OpenStreetMap layer, a Traffic Message Channel (TMC) coded network file. First, those GIS sources might not have a separate node layer, so users need to use NeXTA_for_GIS to automatically create the node layer based on the end feature points of a link curve. In order to support the traffic simulation, users also need to create zone layers and define the additional activity layer to link the node layer to the traffic analysis zones.

4.1.2. Step 2: Extract GIS information from shapefiles and importing to configuration files

For importing shapefiles from regional planning packages, users need to prepare input_link_type.csv, import_GIS_settings.csv (required) and input_node_control_type.csv.

1. input_link_type.csv file

The input_link_type table allows users to define their own specific link types, as long as the flag variables are correctly used to identify how the different link types are connected/related (e.g., freeways connect to arterials using ramps). Only one flag may be used for each link type. Link types can also be used to determine how links are visualized in NeXTA. The data set of link types can be seen in Figure 18.

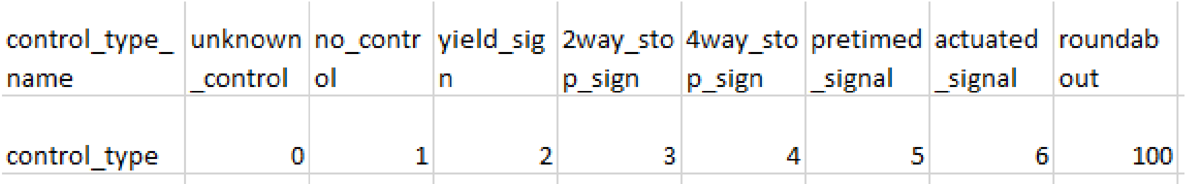


link_type	link_type_name	type_code	default_lane_capacity	default_speed_limit	default_number_of_lanes	capacity_adjustment_factor	travel_time_bias_factor	approximate_cycle_length_in_second	saturation_flow_rate_in_vhc_per_hour_per_lane
0	Private roads	t	650	25	1	1	1	0	1800
1	Interstate freeway	f	1800	65	3	1	1	0	1800
2	principal arterial	a	1800	65	3	1	1	0	1800
3	Other principal arteri	a	1000	45	2	1	1	0	1800
4	Minor arterial	a	850	35	2	1	1	0	1800
5	Major Collectors	a	650	35	1	1	1	0	1800
6	Minor Collectors	a	650	25	1	1	1	0	1800
7	Local Streets	a	650	25	1	1	1	0	1800
8	Ramps	r	1600	55	5	1	1	0	1800
9	Centroid Connectors	c	99999	15	7	1	1	0	1800

Figure 18: input_link_type.csv file

2. input_node_control_type.csv

The input_node_control_type table (Figure 19) defines the control type of nodes in the network in terms of control type name, unknown control, no control, yield sign, 2-way stop sign, 4-way stop sign, pre-timed signal, actuated signal and roundabout. This file is required when using the network import tool, and the control type field is read from the node shapefile.



control_type_name	unknown_control	no_control	yield_sign	2way_stop_sign	4way_stop_sign	pretimed_signal	actuated_signal	roundabout
control_type	0	1	2	3	4	5	6	100

Figure 19: input_node_control_type.csv file

3. import_GIS_settings.csv file

File import_GIS_settings.csv is a user-defined configuration file, and a user can use Excel to edit it. This file can help users to associate/map fields in shapefile DBF files to the AMS Data Hub schema data format, and NeXTA_for_GIS can use this file to read the network geometry from shapefiles and create an Analysis Modeling Simulation (AMS) Data Hub compatible transportation network project file (which is readable by both DTALite and NeXTA).

Depending on the type of data to be imported, the configuration file consists of several sections. As defined in Column A, there are general sections of file, configuration, and sections for different network objects such as node, link, and zone.

The “key” settings given in Column B are used by DTALite/NeXTA to match and convert the shapefiles from the DBF files. “value” in Column C is the matching name between NeXTA’s “key” values, and those given in the shapefiles. Column “required_or_optional” shows which values must be defined for a successful import. Column “allowed_values” show which definitions are related to “value” for binary values (such as oneway vs. twoway, lane vs. link, etc). The “notes” column gives some information on how to fill the column value.

Although this configuration contains many columns, users only need to prepare or change the data in column “value”. The converted shapefiles can be opened in any GIS software (for example open-source Q-GIS) and accessed through attribute tables.

Once a configuration file is established for a regional travel demand format, it can be replicated as a template for future imports from that format.

The open-source GIS library GDAL (Geospatial Data Abstraction Library) is used by NeXTA to import the geometry and field data from GIS shapefiles.

The user needs to be careful with attribute names that have more than 10 characters (such as ControlType in this case). For example, an exported SHP/DBF file will convert the file name to the 10-character length (Controlt~1). To see the correct attribute name definitions, the user can use any GIS software to open the corresponding shapefile and read the name from the attribute table.

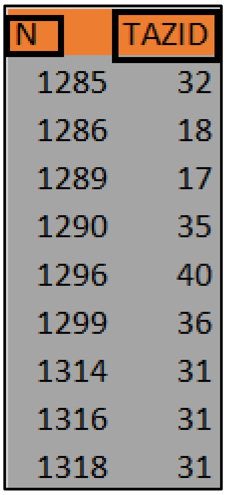
The first file section defines the shapefile names for GIS layers of node, link, and zone. In this example, the file name table looks like Table 3. The reference file names should correspond to the layers exported from planning packages such as VISUM, CUBE and TransCAD. Node, link and zone are the required layers for a successful GIS network import.

Table 3: File section

section	key	value	required_or_optional
file_name	node	Reno_Network_Node.shp	optional
file_name	link	Reno_Network_Link.shp	required
file_name	zone	Reno_Network_Zone.shp	optional

The next section defines node shapefile attributes. The import values are node ID, node name, the zone (TAZ) to which a certain node belongs to, and node control type. The sample of node table is as Table 4, while fields node_id and TAZ are required if there is a node layer. This part will fill out with the data provided in Reno_Network_Node.dbf as shown in Table 5.

Table 4: Reno_Network_Node.dbf



N	TAZID
1285	32
1286	18
1289	17
1290	35
1296	40
1299	36
1314	31
1316	31
1318	31

Table 5: Node section

section	key	value	required_or_optional
node	node_id	N	Required only when the node layer file is provided
node	name		Optional, for display only
node	TAZ	TAZID	Optional, if not provided, then users can use node_number_threshold_as_centroid to specify the first n nodes directly as centroids of zones.
node	control_type		optional, if node control type has been specified. If not, set value from file input_node_control_type.csv

Table 6: Configuration

Configuration	key	value	allowed_values
1	with_decimal_long_lat	yes	yes;no
2	node_number_threshold_as_centroid	0	Default 0

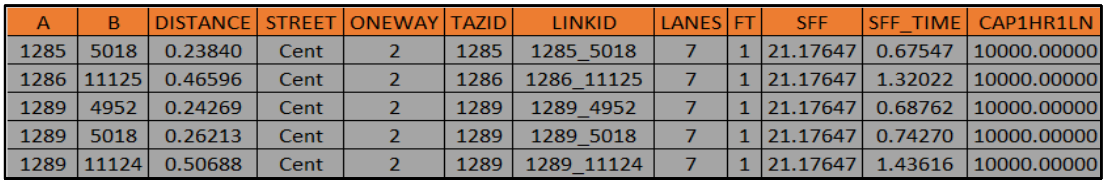
Table 6 shows the configuration table. Key with_decimal_long_lat indicates if a decimal long/lat format is used.

For example, longitude -111.943375 has a decimal point, while -111943375 has no decimal point. If “value” is set to no, a multiplier of 0.000001 is used to convert the sample value of -119813277 to -119.813277.

Key node_number_threshold_as_centroid specifies the value for using nodes directly as zone centroids. In VISUM or TransCAD data set, a centroid layer is provided, while Cube might assume a set of nodes below a certain number (say 3000) is used as zone centroids by default. This number typically is specified in a script for traffic assignment in Cube. For example, there are 2000 nodes with node numbers < 3000. By reading this number threshold, NeXTA can automatically (a) add 2000 traffic analysis zones, and (b) assign a node to the activity location for each newly created zone. Thus, the zone and activity location layers (namely files input_activity_location.csv and input_zone.csv) are created automatically.

The link section defines the link shapefile attributes. In order to code the corresponding attributes correctly, the user can access the link shapefile through GIS software, and read the link attributes. The available link attributes are from and to node, name, link ID, link type, transportation modes that link is open for, direction definition, number of lanes, hourly capacity, speed limit, and number of lanes, capacity, speed limit and link type for reversible lanes, if links are defined as two-way links. Table 7 shows the link DBF file and Table 8 represents how user should extract GIS information from this file and modifying GIS setting file based on that.

Table 7: Reno_Network_Link.dbf file



A	B	DISTANCE	STREET	ONEWAY	TAZID	LINKID	LANES	FT	SFF	SFF_TIME	CAP1HR1LN
1285	5018	0.23840	Cent	2	1285	1285_5018	7	1	21.17647	0.67547	10000.00000
1286	11125	0.46596	Cent	2	1286	1286_11125	7	1	21.17647	1.32022	10000.00000
1289	4952	0.24269	Cent	2	1289	1289_4952	7	1	21.17647	0.68762	10000.00000
1289	5018	0.26213	Cent	2	1289	1289_5018	7	1	21.17647	0.74270	10000.00000
1289	11124	0.50688	Cent	2	1289	1289_11124	7	1	21.17647	1.43616	10000.00000

Table 8: Link section of import_GIS_setting.csv

	section	key	value	required/optional	Remark
1	link	from_node_id	A	Optional but highly recommend to fill in the cell	Typical values include usn. If data is not provided, please set identify_from_node_id_and_to_node_id_based_on_geometry to yes
2	link	to_node_id	B	Optional but highly recommend to fill in the cell	Typical values include dsn. If data is not provided, please set identify_from_node_id_and_to_node_id_based_on_geometry to yes
3	link	name	STREET	optional	For display only
4	link	link_id	LINKID	optional	For searching links by ids
5	link	link_type		optional	If data is not provided, it will set value from input_link_type.csv
6	link	mode_code			Reserved for transit, pedestrian modeling.
7	link	direction	ONEWAY		If no value is given, a one-way link is assumed by default.
8	link	length	DISTANCE	required	The unit should be miles.
9	link	number_of_lanes	LANES	optional	Users can specify use_default_number_of_lanes_from_link_type based on link type.
10	link	hourly_capacity	CAP1HR1LN	optional	Users can specify use_default_lane_capacity_from_link_type based on link type
11	link	speed_limit	SFF	optional	Users can specify use_default_speed_limit_from_link_type based on link type

The link types (input_link_type.csv) at Figure 14 needs to be updated with the correct link types. Number of lanes, speed and capacity data are indirectly defined through the link type table. Table 9 shows the links configuration section.

This configuration section describes general model attributes and import options that can accommodate different network coding conventions. These settings include generating from node and to node IDs, whether the capacity is given per lane, or per link, and default direction of links. Its advanced settings also allow users to import link type, specific speed limit and capacity values, number of lanes, and minimum length for importing links. Explanations of each item in Table 9 are as follows.

Table 9: Link configuration

configuration	key	value	allowed_values
1	identify_from_node_id_and_to_node_id_based_on_geometry	no	yes;no
2	lane_capacity_vs_link_capacity	lane	lane;link
3	default_link_direction	oneway	oneway;twoway
4	use_default_speed_limit_from_link_type	no	yes;no
5	use_default_lane_capacity_from_link_type	no	yes;no
6	use_default_number_of_lanes_from_link_type	no	yes;no
7	minimum_length_for_importing_links	0.00001	

1) In the link layer, if the from_node_id and to_node_id of a link is not provided, for example, in a TransCAD data set, then the flag of identify_from_node_and_to_node_id_based_on_geometry can allow NeXTA to use the geometry information of a link to find the nearby nodes from the node layer to construct the from_node_id and to_node_id fields for a link. If no such a node is available or even no node layer is provided, then NeXTA_for_GIS will create new nodes and use the corresponding new node numbers for the fields of from_node_id and to_node_id. This setting significantly relaxes the data requirements.

2) Key lane_capacity_vs_link_capacity indicates if the capacity filed in the link layer is referred to link capacity across all lanes or simple lane capacity. If "value" = link, then the capacity of link will be divided by the number of lanes as DTALite/NeXTA uses lane-based capacity.

3) Key default_link_direction specifies if a link record is a one-way link or a two-way link. If a record is a two-way link, then two directional links will be created by NeXTA.

4,5,6) Keys use_default_speed_limit_from_link_type; use_default_lane_capacity_from_link_type; and use_default_number_of_lanes_from_link_type allow users to input their link-type specific lane capacity, number of lanes and speed limit fields in file input_link_type.csv. If "value" = yes, then users do not need to specify the field names in section link for these three important link attributes.

7) Key minimum_length_for_importing_links specifies the minimum length threshold for importing a link. With a default value of 0.00001, NeXTA does not import a link with a length of 0. If a negative value of -0.00001 is given, then NeXTA will keep the links with zero distance in the final imported data set.

The zone section (Table 10) defines the zone shapefile attributes. Only a zone ID field is needed. If the zone layer does not present, one can set a positive value (say 3000) for key node_number_threshold_as_centroid in a section configuration to add zones and the corresponding activity locations. Zone is required only if the zone file is given in the section file_name.

Table 10: Zone section

section	key	value	required_or_optional
zone	zone_id	Id	optional

4.1.3. Step 3: Generating the traffic network using NeXTA_for_GIS

1: Open NeXTA_for_GIS.exe

2: Click on "File"→"Import GIS Data Set" (Figure 20)

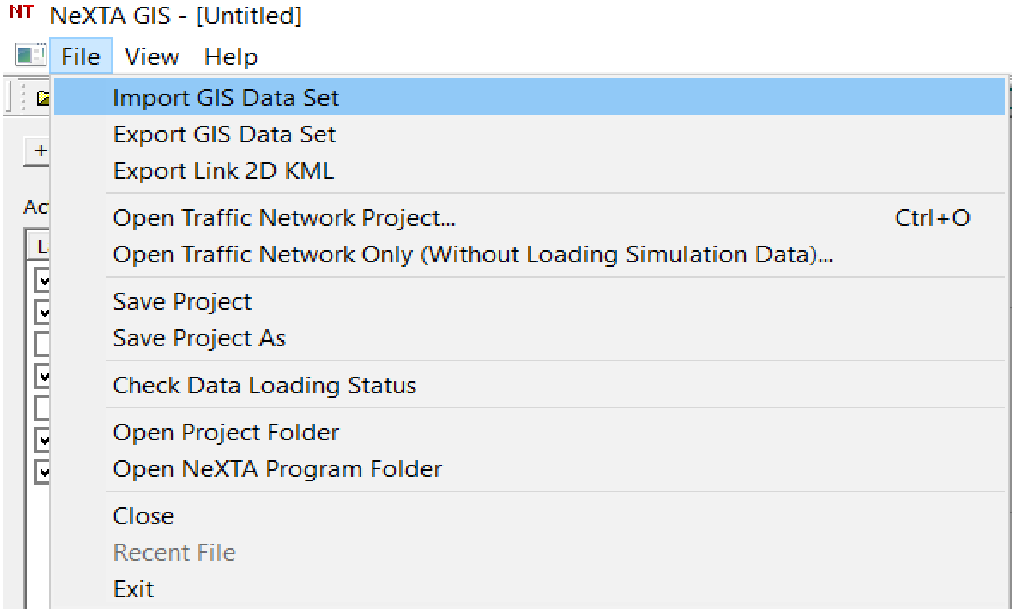
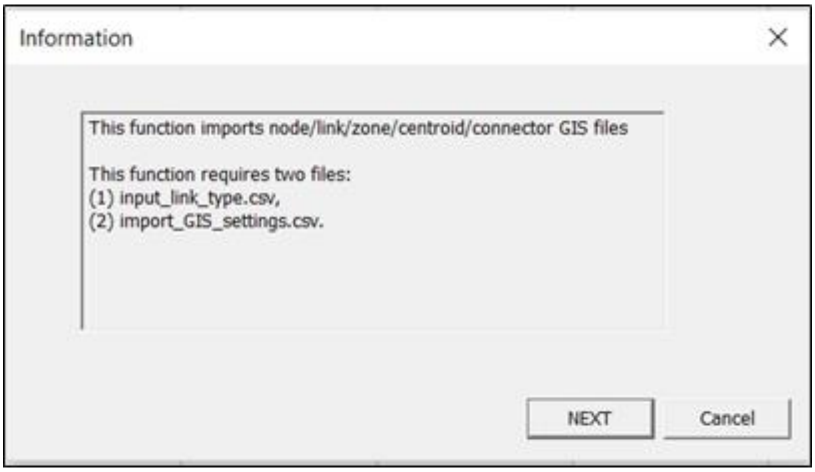

NT NeXTA GIS - [Untitled]
File View Help
Import GIS Data Set
Export GIS Data Set
Export Link 2D KML
Open Traffic Network Project... Ctrl+O
Open Traffic Network Only (Without Loading Simulation Data)...
Save Project
Save Project As
Check Data Loading Status
Open Project Folder
Open NeXTA Program Folder
Close
Recent File
Exit

Figure 20: Import GIS Date Set command

The following dialogue box (Figure 21) will be opened.

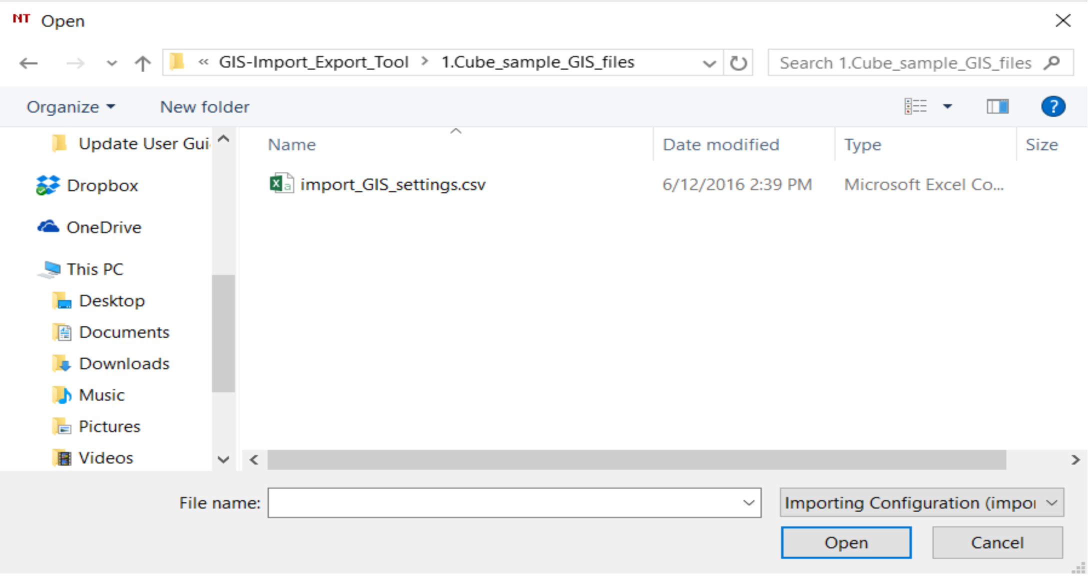


This function imports node/link/zone/centroid/connector GIS files

This function requires two files:
(1) input_link_type.csv,
(2) import_GIS_settings.csv.

Figure 21: Information dialogue box after clicking "Import GIS Data Set"

Click on NEXT to load import_GIS_setting.csv as shown in Figure 22.

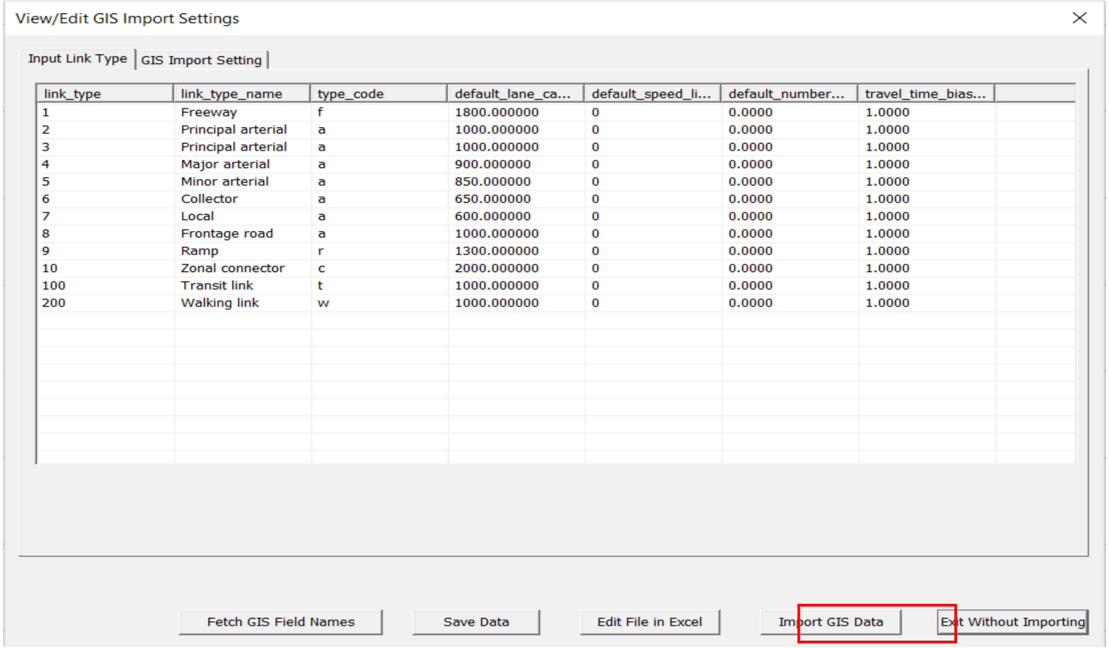


Name	Date modified	Type	Size
import_GIS_settings.csv	6/12/2016 2:39 PM	Microsoft Excel Co...	

File name:
Importing Configuration (impo

Figure 22: Import "import_GIS_settings.csv" file

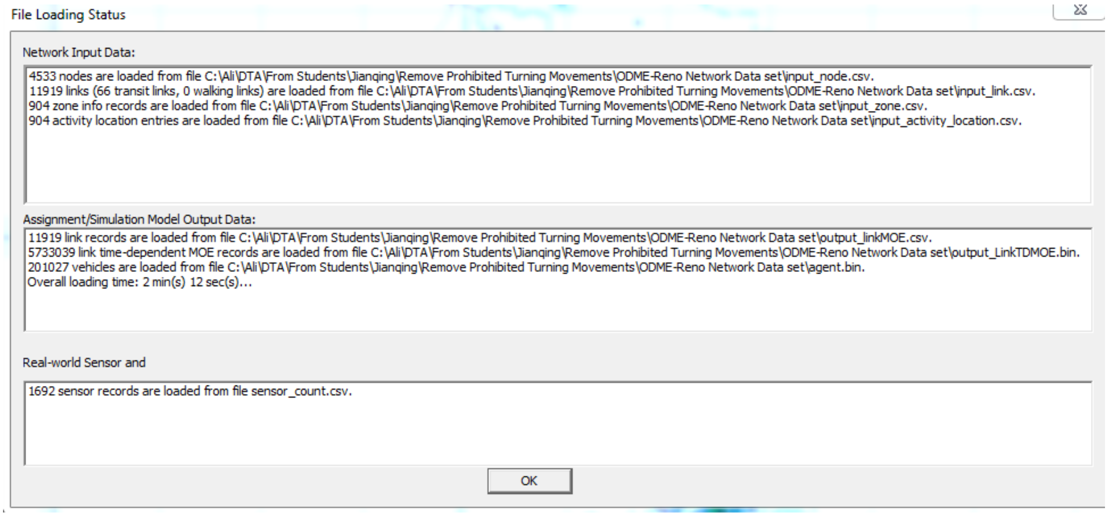
3: Select "Import GIS Data" as shown in Figure 23.



link_type	link_type_name	type_code	default_lane_ca...	default_speed_li...	default_number...	travel_time_bias...
1	Freeway	f	1800.000000	0	0.0000	1.0000
2	Principal arterial	a	1000.000000	0	0.0000	1.0000
3	Principal arterial	a	1000.000000	0	0.0000	1.0000
4	Major arterial	a	900.000000	0	0.0000	1.0000
5	Minor arterial	a	850.000000	0	0.0000	1.0000
6	Collector	a	650.000000	0	0.0000	1.0000
7	Local	a	600.000000	0	0.0000	1.0000
8	Frontage road	a	1000.000000	0	0.0000	1.0000
9	Ramp	r	1300.000000	0	0.0000	1.0000
10	Zonal connector	c	2000.000000	0	0.0000	1.0000
100	Transit link	t	1000.000000	0	0.0000	1.0000
200	Walking link	w	1000.000000	0	0.0000	1.0000

Figure 23: "View/Edit GIS Import Settings" dialogue box

The file loading status will pop up, which gives all the information of networks such as number of nodes, links, and zones (Figure 24).



Network Input Data:

```
4533 nodes are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\input_node.csv.
11919 links (66 transit links, 0 walking links) are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\input_link.csv.
904 zone info records are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\input_zone.csv.
904 activity location entries are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\input_activity_location.csv.
```

Assignment/Simulation Model Output Data:

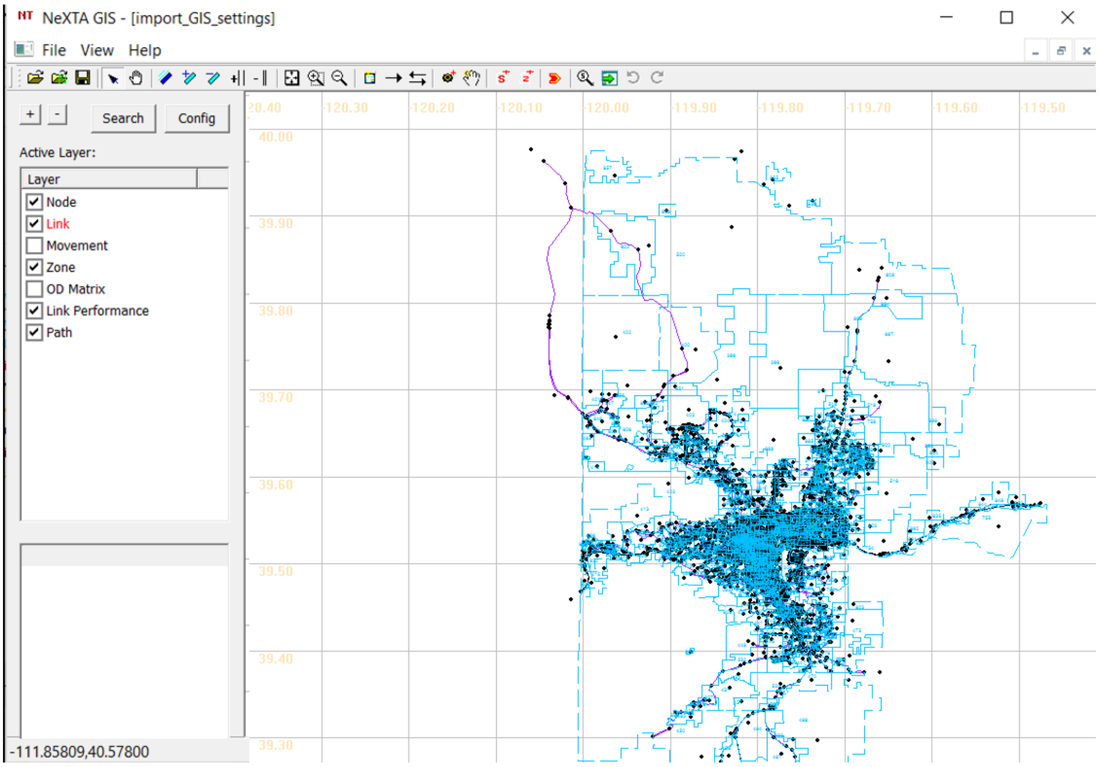
```
11919 link records are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\output_linkMOE.csv.
5733039 link time-dependent MOE records are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\output_LinkTDMOE.bin.
201027 vehicles are loaded from file C:\Ali\DTA\From Students\Jianqing\Remove Prohibited Turning Movements\ODME-Reno Network Data set\agent.bin.
Overall loading time: 2 min(s) 12 sec(s)...
```

Real-world Sensor and

```
1692 sensor records are loaded from file sensor_count.csv.
```

Figure 24: "File Loading Status" information box

After clicking "OK", the network as shown in Figure 25 is displayed.



☑ Node
☑ Link
□ Movement
☑ Zone
□ OD Matrix
☑ Link Performance
☑ Path

Figure 25: The Reno/Sparks network on display view

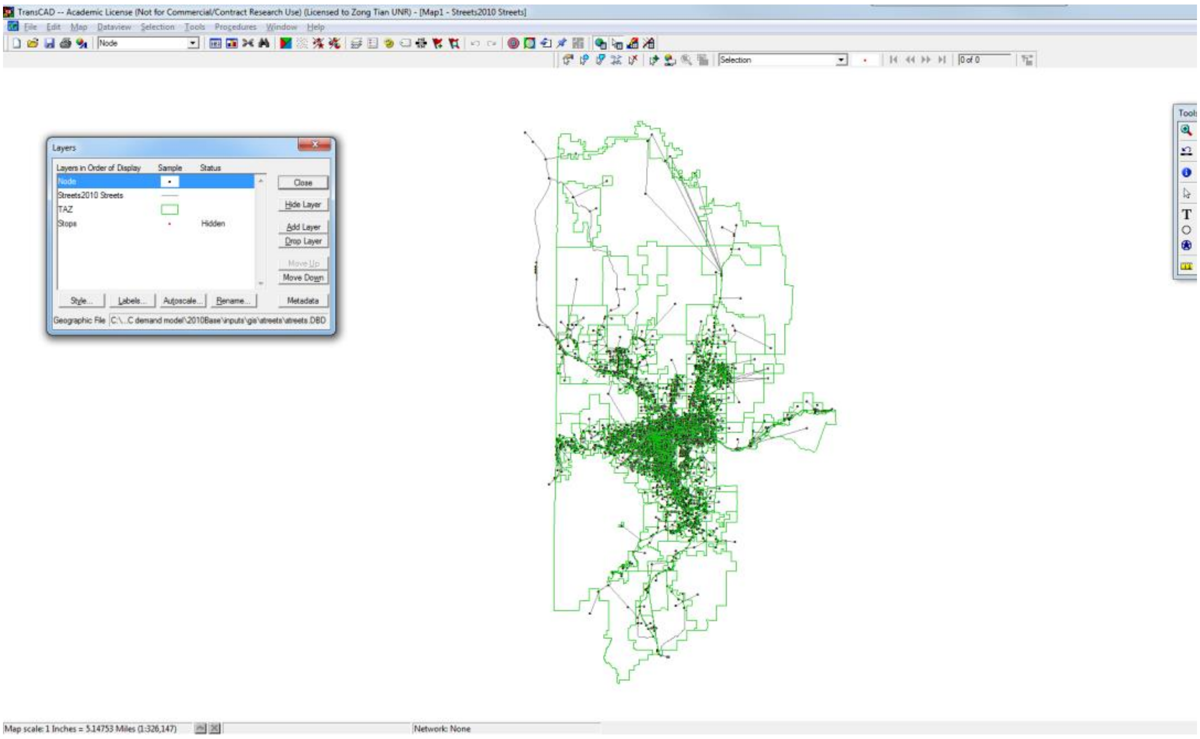
4: Go to the “File” -> “Save Project As” to save the imported network as a new text file and generate the necessary input files for the model.

4.2. Export Network and Demand Files from TransCAD®

The first step in the network conversion process is to create a set of shapefiles describing the network to be imported into NeXTA. This is normally accomplished using export functions in the software used to prepare the selected network, and can be divided into three internal steps: 1) Load the network of the regional demand model in the originating software; 2) Export the network as shapefiles; and 3) Export demand matrices/tables.

4.2.1. Load the network in TransCAD®

The Reno-Sparks network was coded in TransCAD® and must be exported as a set of shapefiles. First, the network is loaded in TransCAD® as shown in Figure 26.


TransCAD -- Academic License (Not for Commercial/Contract Research Use) (Licensed to Zong Tian UNR) - [Map1 - Streets2010 Streets]
File Edit Map Dataview Selection Tools Procedures Window Help
Node
Selection
0 of 0
Tools
Layers
Close
Hide Layer
Add Layer
Drop Layer
Move Up
Move Down
Style...
Labels...
Autoscale...
Rename...
Metadata
Map scale: 1 Inches = 5.14753 Miles (1:326,147)
Network: None

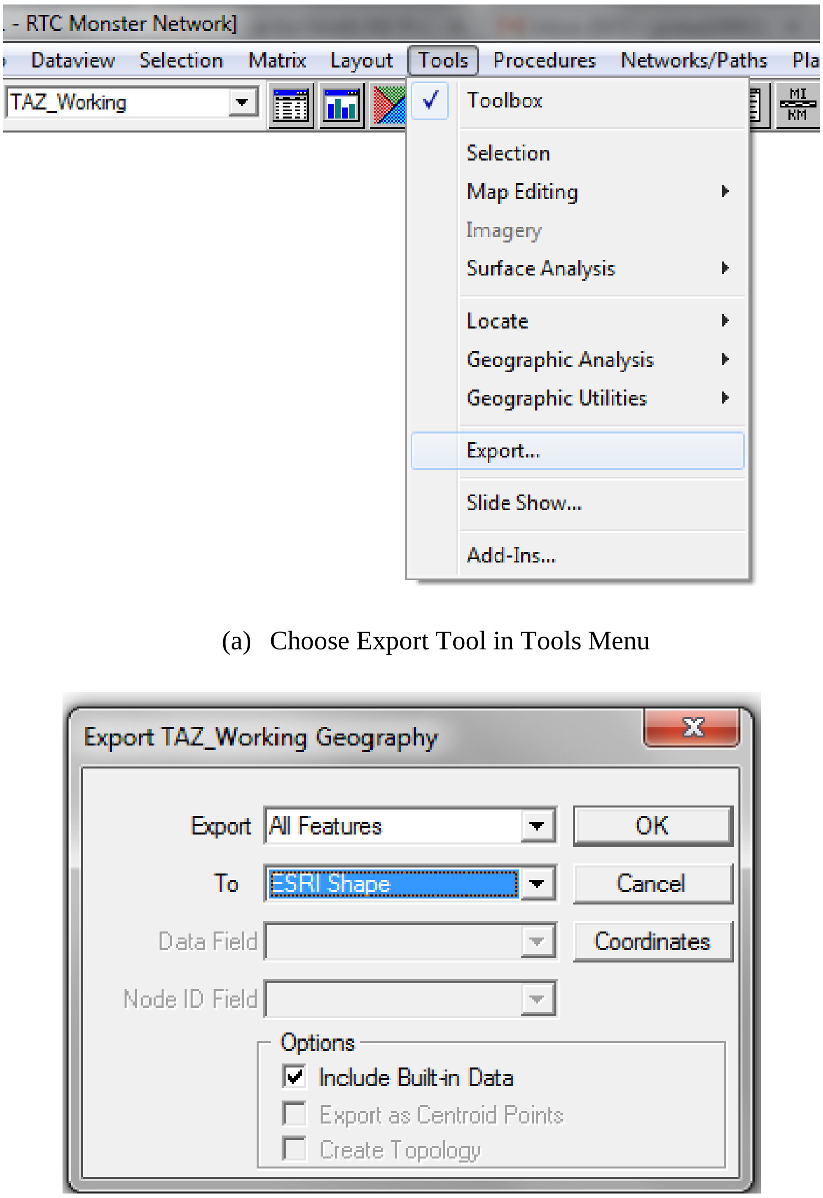
Layers in Order of Display	Sample	Status
Node	■	
Streets2010 Streets	—	
TAZ	▭	
Stops	•	Hidden

Geographic File: C:\...C demand model\2010Base\inputs\gis\streets\streets.DBD

Figure 26: Reno-Sparks Regional Travel Demand Model Loaded in TransCAD®

4.2.2. Export network as shapefiles from TransCAD®

By using the export tool in TransCAD® to export the network GIS shapefiles, the network is split into multiple component layers and saved as separate shapefiles. One should select the node, link and TAZ zone layers to ensure that the conversion process can successfully create a new network in NeXTA. An example process of exporting a layer to a shapefile is shown in Figure 27.



(a) Choose Export Tool in Tools Menu

Export TAZ_Working Geography

Export: All Features

To: ESRI Shape

Data Field:

Node ID Field:

Options

☑ Include Built-in Data

☐ Export as Centroid Points

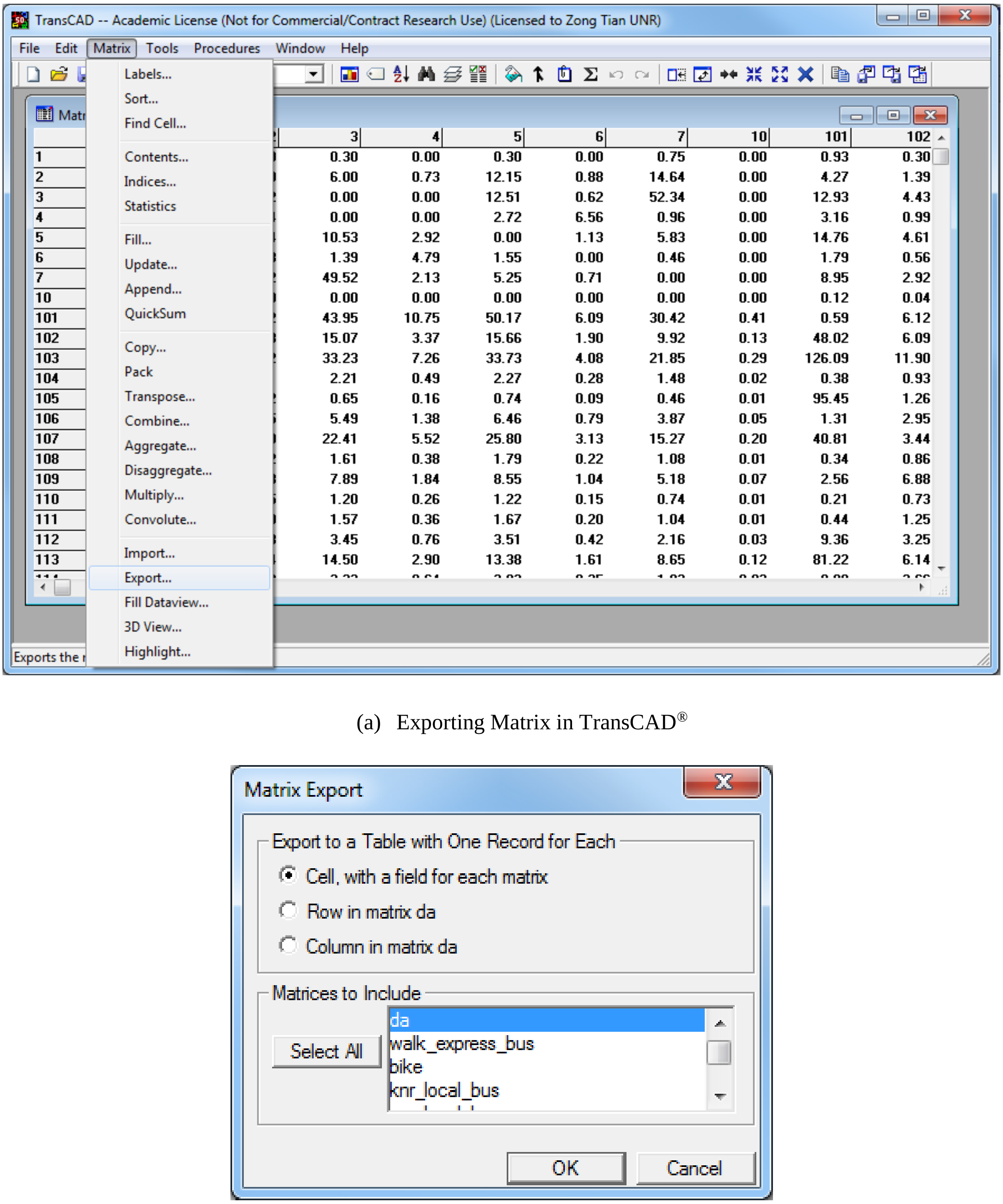
☐ Create Topology

(b) Choose Export Option

Figure 27: Exporting Network Data as Shape Files

4.2.3. Export demand tables/matrices from TransCAD®

The demand matrices in TransCAD® are stored as binary files. In order for NeXTA/DTALite to read demand data, one can export the demand matrix files from TransCAD® into the "Export to a table with one record for each cell, with a field for each matrix" option when preparing tables for export to NeXTA, as shown in Figure 28. Exporting the demand matrices for the Reno-Sparks travel demand model produces 24 demand tables describing the number of trips between zones for different demand types and different time periods.



	3	4	5	6	7	10	101	102
1	0.30	0.00	0.30	0.00	0.75	0.00	0.93	0.30
2	6.00	0.73	12.15	0.88	14.64	0.00	4.27	1.39
3	0.00	0.00	12.51	0.62	52.34	0.00	12.93	4.43
4	0.00	0.00	2.72	6.56	0.96	0.00	3.16	0.99
5	10.53	2.92	0.00	1.13	5.83	0.00	14.76	4.61
6	1.39	4.79	1.55	0.00	0.46	0.00	1.79	0.56
7	49.52	2.13	5.25	0.71	0.00	0.00	8.95	2.92
10	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.04
101	43.95	10.75	50.17	6.09	30.42	0.41	0.59	6.12
102	15.07	3.37	15.66	1.90	9.92	0.13	48.02	6.09
103	33.23	7.26	33.73	4.08	21.85	0.29	126.09	11.90
104	2.21	0.49	2.27	0.28	1.48	0.02	0.38	0.93
105	0.65	0.16	0.74	0.09	0.46	0.01	95.45	1.26
106	5.49	1.38	6.46	0.79	3.87	0.05	1.31	2.95
107	22.41	5.52	25.80	3.13	15.27	0.20	40.81	3.44
108	1.61	0.38	1.79	0.22	1.08	0.01	0.34	0.86
109	7.89	1.84	8.55	1.04	5.18	0.07	2.56	6.88
110	1.20	0.26	1.22	0.15	0.74	0.01	0.21	0.73
111	1.57	0.36	1.67	0.20	1.04	0.01	0.44	1.25
112	3.45	0.76	3.51	0.42	2.16	0.03	9.36	3.25
113	14.50	2.90	13.38	1.61	8.65	0.12	81.22	6.14

(a) Exporting Matrix in TransCAD®

Matrix Export

Export to a Table with One Record for Each

- ● Cell, with a field for each matrix
- ○ Row in matrix da
- ○ Column in matrix da

Matrices to Include

- da
- walk_express_bus
- bike
- knr_local_bus

(b) Matrix Export Option in TransCAD®

Figure 28. Export Demand Matrices from TransCAD®

4.3. Use NeXTA's Import Network Tool to Import GIS Network Shape Files

4.3.1. Prepare configuration files for conversion

The first step in converting the network is to create several configuration CSV files in the folder containing the exported shapefiles. The required configuration files include (1) input_node_control_type.csv, (2) input_link_type.csv, (3) input_demand_meta_data.csv (and related demand files), and (4) import_GIS_settings.csv. To ensure that NeXTA imports a correct network profile, the user needs to update the configurations of link types, node control types and GIS settings in this step. For more details regarding configuring these files, refer to Section 4.1.

Link types to be imported into NeXTA should be consistent with the types used in the original TransCAD® network. In the network, a different list of link types is used instead of the default values given in the NeXTA configuration files. For that reason, the input_link_type.csv file needs to be updated to reflect the current types.

The updated link type table should look like Table 11.

Table 11. Updated Configurations for Link Type

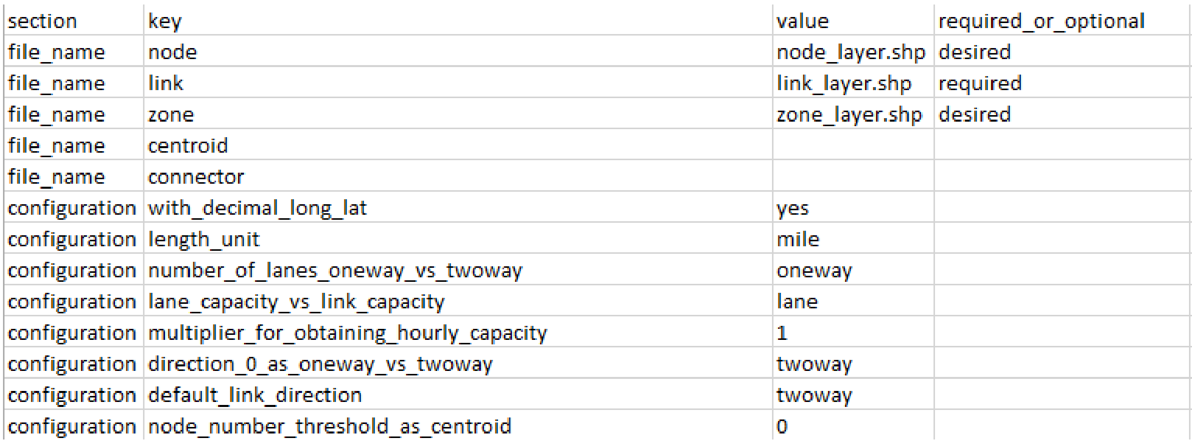
link_type	**link_type_name**	**type_code**	**default_lane_capacity**	**default_speed_limit**
0	Private roads	t	650	25
1	Interstate freeway	f	1800	65
2	principal arterial	a	1800	65
3	Other principal arterial	a	1000	45
4	Minor arterial	a	850	35
5	Major collectors	a	650	35
6	Minor collectors	a	650	25
7	Local streets	a	650	25
8	Ramps	r	1600	55
9	Centroid connectors	c	99999	15

The input_node_control_type.csv file should also be updated with the current control type, especially for signalized intersections. Codes for control types must be consistent with the settings in the network profile exported from TransCAD®. The updated node control type table should look like Table 12.

Table 12. Configurations for input_node_control_type.csv

control_ type_name	**unknown _control**	**no_ control**	**yield_ sign**	**2way_ stop_ sign**	**4way_ stop_ sign**	**pretimed _signal**	**actuated _signal**	**roundabout**
control_type	0	1	2	3	4	5	6	100

As mentioned in Section 4.1, the import_GIS_settings.csv file is to identify and connect the fields in the input shapefile (DBF files) to the AMS data hub schema data format, allowing NeXTA to read the network geometry from shapefiles and create an AMS data hub compatible transportation network project (.tnp) file (which is readable by both DTALite and NeXTA). Figure 29 shows a screenshot of the beginning section of the updated GIS setting table.



section	key	value	required_or_optional
file_name	node	node_layer.shp	desired
file_name	link	link_layer.shp	required
file_name	zone	zone_layer.shp	desired
file_name	centroid		
file_name	connector		
configuration	with_decimal_long_lat	yes	
configuration	length_unit	mile	
configuration	number_of_lanes_oneway_vs_twoway	oneway	
configuration	lane_capacity_vs_link_capacity	lane	
configuration	multiplier_for_obtaining_hourly_capacity	1	
configuration	direction_0_as_oneway_vs_twoway	twoway	
configuration	default_link_direction	twoway	
configuration	node_number_threshold_as_centroid	0	

Figure 29. Screenshot of beginning section of input_GIS_settings.csv

4.3.2. Use NeXTA's import network tool to convert the network

Using NeXTA's "Import GIS Planning Data Set" tool under Menu—File—Import—GIS Planning Data Set, the network exported from TransCAD® can now be imported into NeXTA. The conversion process is shown in Figure 30.

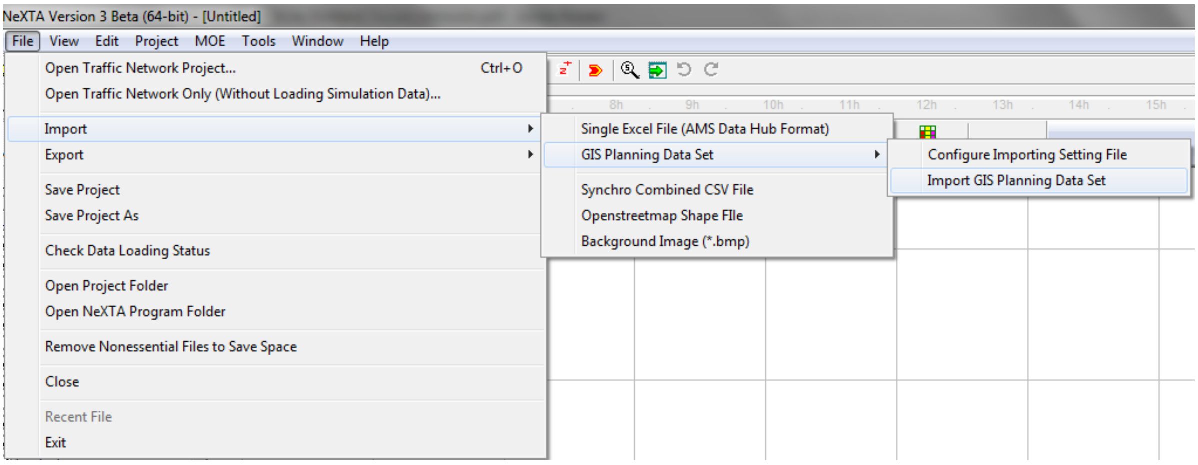
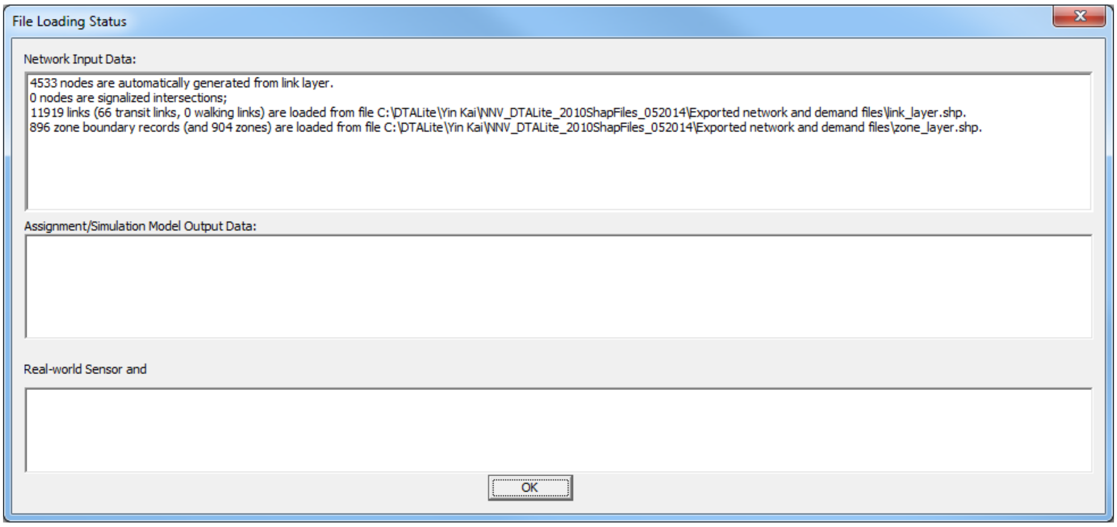

NeXTA Version 3 Beta (64-bit) - [Untitled]
File View Edit Project MOE Tools Window Help
Open Traffic Network Project... Ctrl+O
Open Traffic Network Only (Without Loading Simulation Data)...
Import
Export
Save Project
Save Project As
Check Data Loading Status
Open Project Folder
Open NeXTA Program Folder
Remove Nonessential Files to Save Space
Close
Recent File
Exit
Single Excel File (AMS Data Hub Format)
GIS Planning Data Set
Synchro Combined CSV File
Openstreetmap Shape FIle
Background Image (*.bmp)
Configure Importing Setting File
Import GIS Planning Data Set
8h
9h
10h
11h
12h
13h
14h
15h

Figure 30. Import Planning Shape Files

After the successful conversion process, NeXTA displays a "File Loading Status" window as shown in Figure 31.

The final imported network is shown in Figure 32, which has 4,533 nodes, 11,919 links, and 904 zones.



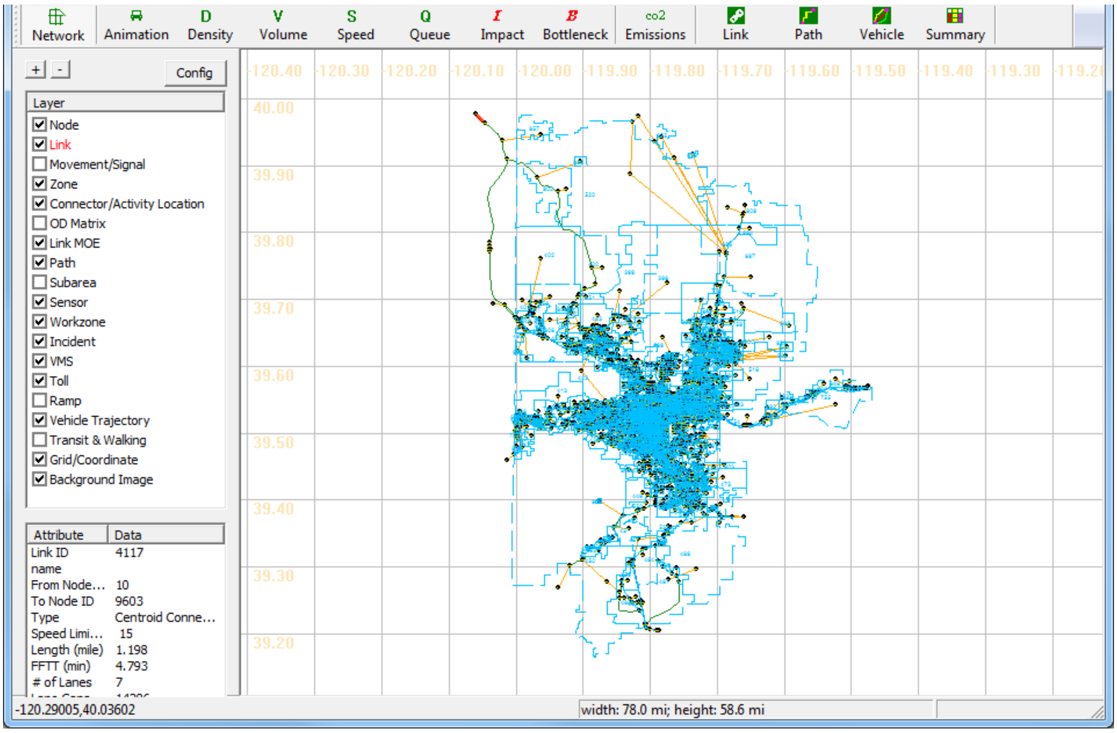
Network Input Data:

```
4533 nodes are automatically generated from link layer.
0 nodes are signalized intersections;
11919 links (66 transit links, 0 walking links) are loaded from file C:\DTALite\Yin Kai\NNV_DTALite_2010ShapFiles_052014\Exported network and demand files\link_layer.shp.
896 zone boundary records (and 904 zones) are loaded from file C:\DTALite\Yin Kai\NNV_DTALite_2010ShapFiles_052014\Exported network and demand files\zone_layer.shp.
```

Assignment/Simulation Model Output Data:

Real-world Sensor and

Figure 31. NeXTA File Loading Status for Reno-Sparks Network



Layer
- ☑ Node
- ☑ Link
- □ Movement/Signal
- ☑ Zone
- ☑ Connector/Activity Location
- □ OD Matrix
- ☑ Link MOE
- ☑ Path
- □ Subarea
- ☑ Sensor
- ☑ Workzone
- ☑ Incident
- ☑ VMS
- ☑ Toll
- □ Ramp
- ☑ Vehicle Trajectory
- □ Transit & Walking
- ☑ Grid/Coordinate
- ☑ Background Image

Attribute	Data
Link ID	4117
name	
From Node…	10
To Node ID	9603
Type	Centroid Conne…
Speed Limi…	15
Length (mile)	1.198
FFTT (min)	4.793
# of Lanes	7

-120.29005,40.03602

width: 78.0 mi; height: 58.6 mi

Figure 32. Reno-Sparks Network Loaded in NeXTA

4.3.3. Save the new network as a new project file

The last step is to create a new destination folder and to save the network as a new network project (File -> Save Project As) in the created folder. The new network can now be viewed through NeXTA. It should be noted that if there are multiple OD tables (see next section), the user needs to manually add multiple OD demand files to the new project folder.

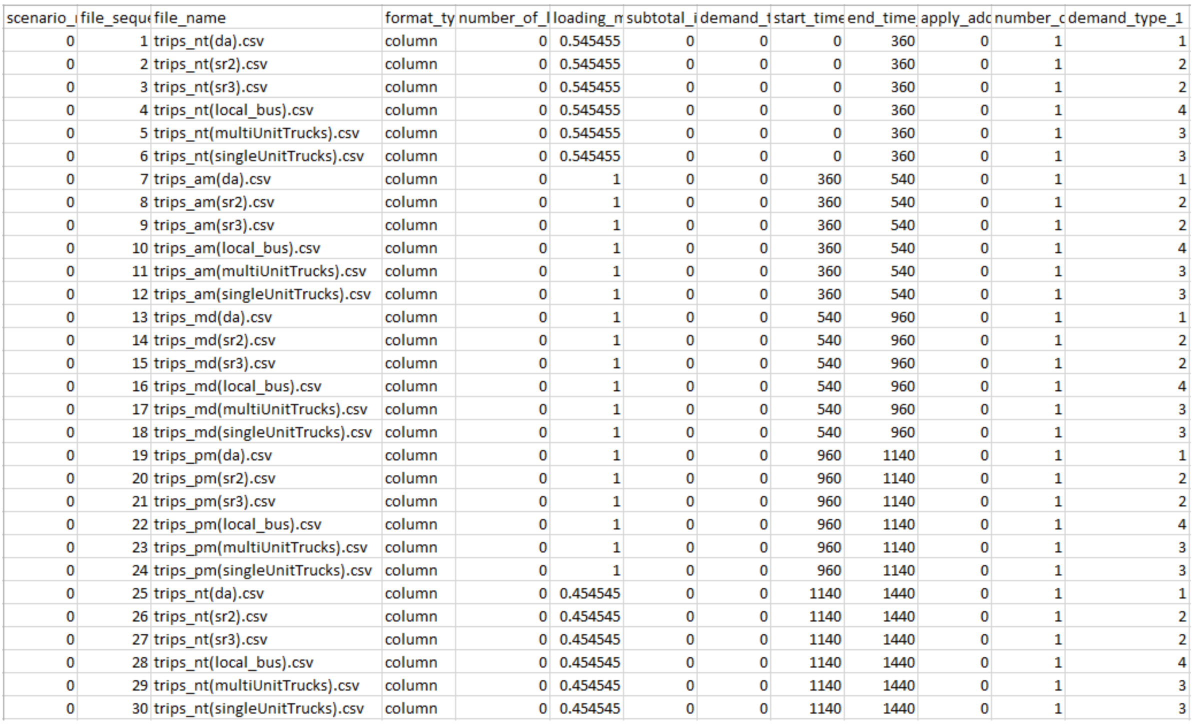
4.4. Read Demand Data into NeXTA from Regional Travel Demand Model

Since there are multiple OD demand tables, the user needs to prepare input_demand_meta_data.csv associated with the demand files.

In the input_demand_meta_data.csv file, the followings are required to be specified:

- Specify the demand file name (e.g., trips_am(local_bus).csv) and format (e.g., column);
- Specify the number of lines in the demand file to be skipped by DTALite (0 for our case);
- Indicate whether subtotals are present in the last column (zero for none);
- Specify the loading start time and end time for the demand file (e.g.: 360 to 540, meaning 6am to 9pm);
- Specify the demand types associated with the demand file.

A screenshot of input_demand_meta_data.csv file is provided in Figure 33.



scenario_	file_seque	file_name	format_ty	number_of_l	loading_m	subtotal_i	demand_t	start_time	end_time	apply_adc	number_c	demand_type_1
0	1	trips_nt(da).csv	column	0	0.545455	0	0	0	360	0	1	1
0	2	trips_nt(sr2).csv	column	0	0.545455	0	0	0	360	0	1	2
0	3	trips_nt(sr3).csv	column	0	0.545455	0	0	0	360	0	1	2
0	4	trips_nt(local_bus).csv	column	0	0.545455	0	0	0	360	0	1	4
0	5	trips_nt(multiUnitTrucks).csv	column	0	0.545455	0	0	0	360	0	1	3
0	6	trips_nt(singleUnitTrucks).csv	column	0	0.545455	0	0	0	360	0	1	3
0	7	trips_am(da).csv	column	0	1	0	0	360	540	0	1	1
0	8	trips_am(sr2).csv	column	0	1	0	0	360	540	0	1	2
0	9	trips_am(sr3).csv	column	0	1	0	0	360	540	0	1	2
0	10	trips_am(local_bus).csv	column	0	1	0	0	360	540	0	1	4
0	11	trips_am(multiUnitTrucks).csv	column	0	1	0	0	360	540	0	1	3
0	12	trips_am(singleUnitTrucks).csv	column	0	1	0	0	360	540	0	1	3
0	13	trips_md(da).csv	column	0	1	0	0	540	960	0	1	1
0	14	trips_md(sr2).csv	column	0	1	0	0	540	960	0	1	2
0	15	trips_md(sr3).csv	column	0	1	0	0	540	960	0	1	2
0	16	trips_md(local_bus).csv	column	0	1	0	0	540	960	0	1	4
0	17	trips_md(multiUnitTrucks).csv	column	0	1	0	0	540	960	0	1	3
0	18	trips_md(singleUnitTrucks).csv	column	0	1	0	0	540	960	0	1	3
0	19	trips_pm(da).csv	column	0	1	0	0	960	1140	0	1	1
0	20	trips_pm(sr2).csv	column	0	1	0	0	960	1140	0	1	2
0	21	trips_pm(sr3).csv	column	0	1	0	0	960	1140	0	1	2
0	22	trips_pm(local_bus).csv	column	0	1	0	0	960	1140	0	1	4
0	23	trips_pm(multiUnitTrucks).csv	column	0	1	0	0	960	1140	0	1	3
0	24	trips_pm(singleUnitTrucks).csv	column	0	1	0	0	960	1140	0	1	3
0	25	trips_nt(da).csv	column	0	0.454545	0	0	1140	1440	0	1	1
0	26	trips_nt(sr2).csv	column	0	0.454545	0	0	1140	1440	0	1	2
0	27	trips_nt(sr3).csv	column	0	0.454545	0	0	1140	1440	0	1	2
0	28	trips_nt(local_bus).csv	column	0	0.454545	0	0	1140	1440	0	1	4
0	29	trips_nt(multiUnitTrucks).csv	column	0	0.454545	0	0	1140	1440	0	1	3
0	30	trips_nt(singleUnitTrucks).csv	column	0	0.454545	0	0	1140	1440	0	1	3

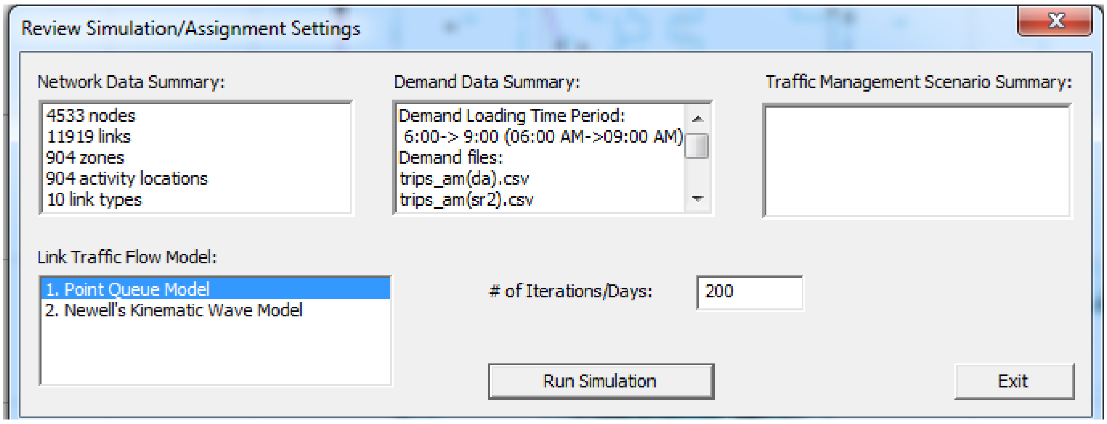
Figure 33. Example of Preparation for Multiple OD Files in input_demand_meta_data.csv

After all network and demand data are imported, the network is ready for running DTA.

4.5. Run Assignment with DTALite to Equilibrium

The dynamic traffic assignment simulation is performed by DTALite, which is directly accessed by pressing the Run Simulation button located in the toolbar menu. Simulation settings should be edited in the input_scenario_settings.csv file prior to initiating the assignment engine.

The popup window that appears shows the defined settings, and allows a selection of the traffic flow model, traffic assignment method, the number of iterations, and the demand loading multiplier, as shown in Figure 34. When the selections are made, pressing "Run Simulation" button will start the DTA simulation.



Review Simulation/Assignment Settings

Network Data Summary:

4533 nodes
11919 links
904 zones
904 activity locations
10 link types

Demand Data Summary:

Demand Loading Time Period:
6:00-> 9:00 (06:00 AM->09:00 AM)
Demand files:
trips_am(da).csv
trips_am(sr2).csv

Traffic Management Scenario Summary:

Link Traffic Flow Model:

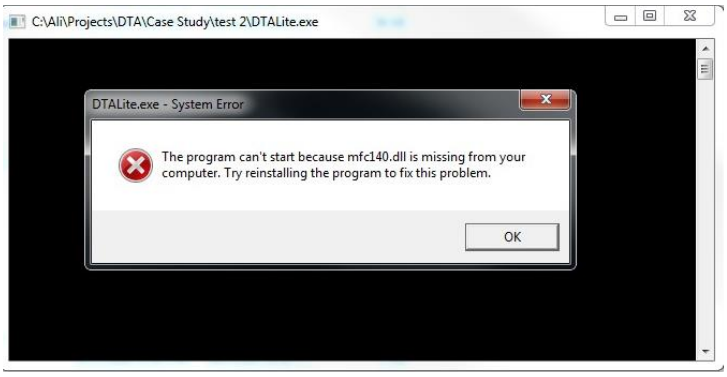
1. Point Queue Model
2. Newell's Kinematic Wave Model

# of Iterations/Days: 200

Figure 34. Model Selection for Running Traffic Assignment

At the first run, the point queue model and method of successive average should be used. The number of iterations can be set to 20. DTALite 64-bit version should be used in order to implement the parallel computation to speed up the calculation. For the Reno-Sparks regional network with 20 simulation runs for around 1,200,000 vehicles, DTALite took 1h 43min to complete on an Intel Core I7 3630QM (2.4GHz quad core) with 16GB RAM. It resulted in an average travel time of 24.90 min and an average trip length of 6.79 miles.

The Microsoft Visual C++ 2015 Redistributable package for parallel computing in DTALite (https://www.microsoft.com/en-us/download/details.aspx?id=48145) should be installed before running the simulation. Failing to install this package may result in the error shown in Figure 35.


C:\Ali\Projects\DTA\Case Study\test 2\DTALite.exe

DTALite.exe - System Error

The program can't start because mfc140.dll is missing from your computer. Try reinstalling the program to fix this problem.

Figure 35: The error if "Microsoft Visual C++ 2015 Redistributable package" is not install

4.6. Change Configurations in input_scenarios_settings.csv File

The initial DTA results might not be consistent with the field observations. Before the OD information can be used reliably to represent the study network conditions, it must be calibrated to observed traffic data. One calibration approach is to adjust the OD matrices. This can be a manual process of adjusting trip table and assigning trips to better fit observed data such as link counts. It can also be an automated step, which is commonly known as Origin Destination Matrix Estimation (ODME).

When massive field observation data are available, the user needs to calibrate the DTA model to adapt the available data by modifying the O-D matrices.

Before using ODME to calibrate OD matrices, the parameters for the input_scenarios_settings.csv file need to be set. Table 13 lists the related attributes for ODME and gives corresponding values in this case.

Table 13. Configuration of Related Attributes for ODME in input_scenarios_setting.csv

Data Field	**Value**	**Notes**
number_of_iterations	50	The total number of iterations for ODME
traffic_flow_model	1	This parameter defines a specific traffic flow model used in both assignment and ODME of DTALite; 1 indicates a point queue model in this example. The selection of Newell's KW model is also feasible.
signal_representation_model	0	This parameter defines a specific signal control for DTALite.
traffic_assignment_method	3	This assignment method of "3" is dedicated to ODME
ODME_start_iteration	20	It defines the first iterative assignment period to converge to the user equilibrium state, and could generate a sufficient number of paths for path flow adjustment. The iteration number also indicates that ODME will begin at the 21th iteration.
ODME_end_iteration	100	It defines that ODME will end at the 50the iteration.
ODME_max_percentage_deviation_wrt_hist_demand	40	The maximum percentage of demand deviation from base-line dynamic demand.
ODME_step_size	0.05	Moving size of each step in path flow adjustment algorithm
calibration_data_start_time_in_min	360	This and the following parameter specify the time window for ODME to use the sensor data. Note that, users can prepare a long period of sensor data, say from 0 to 24 hours, but only use part of sensor data, say between min 990 and 1050, for calibration.
calibration_data_end_time_in_min	540	

4.7. Prepare Field (Sensor) Data for Origin Destination Matrix Estimation (ODME)

NeXTA/DTALite's ODME model requires observed field data to be stored in file sensor_count.csv. Therefore, this file must be prepared before executing the ODME process. Sensor data uses a flexible format and allows multiple types of observed data related to model validation and calibration in the network. Data types can include link volume, occupancy, speed, and travel time field data for specific locations and time periods.

4.7.1. Check what kind of sensor data is available

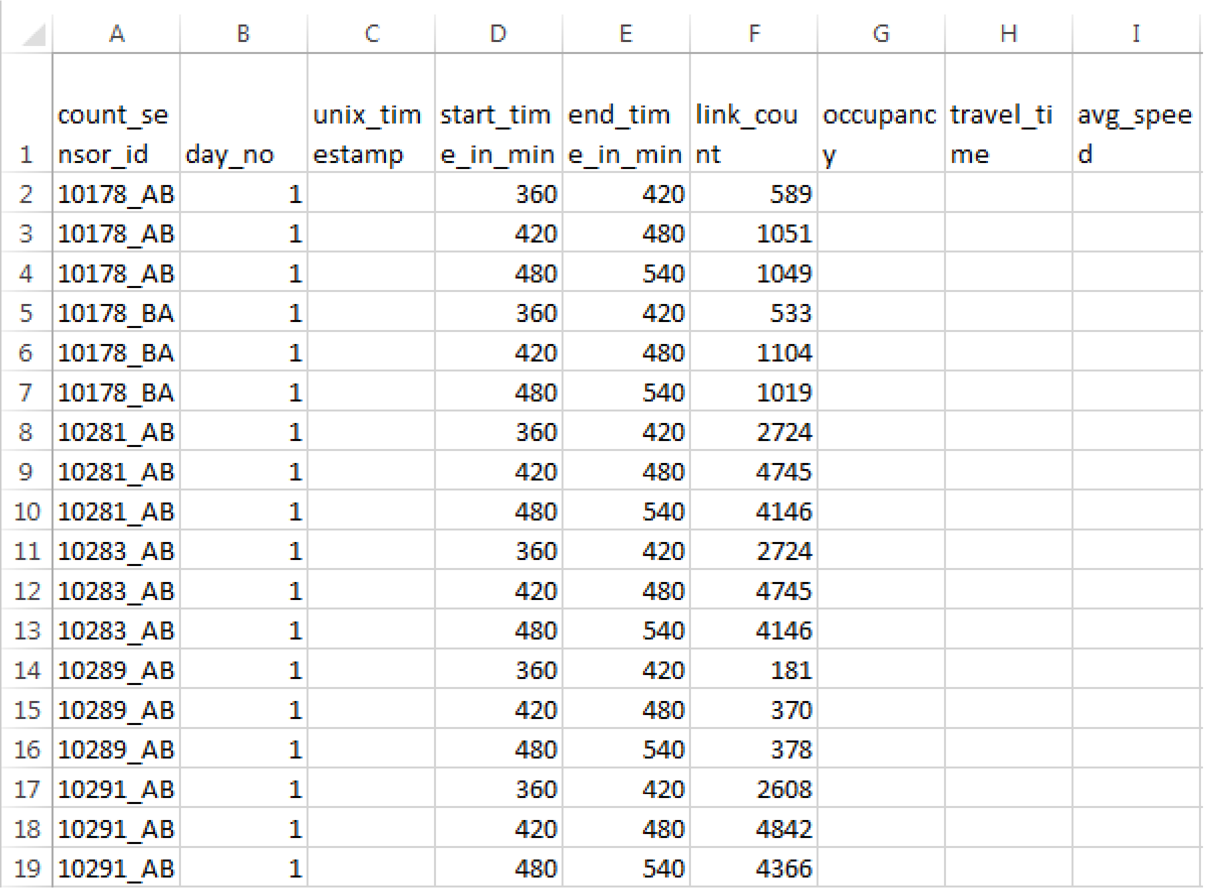
In this project, for the Reno-Sparks network, hourly volumes of major roads and highways in the network were obtained from TRINA.

4.7.2. Prepare sensor_count.csv File

Hourly link volume data should be input in column "link_count" in sensor_count.csv. If other types of sensor data are available, they should be input in corresponding columns. For example, average link speed data should be input in column “avg_speed”, and travel time data should be input in column “travel_time”.

To map sensors to links (in file input_link.csv), one of the following two methods can be used to specify a link with sensors: (i) use the combination of fields “from_node_id” and “to_node_id”, or (ii) use field “count_sensor_id”, which should be first defined in file input_link.csv. It should be noted that if any values in the column of “count_sensor_id” are not defined in file input_link.csv, DTALite will issue warning messages. In this project, the second method was used. Sample values of the fields for sensor data can be seen in Figure 36.

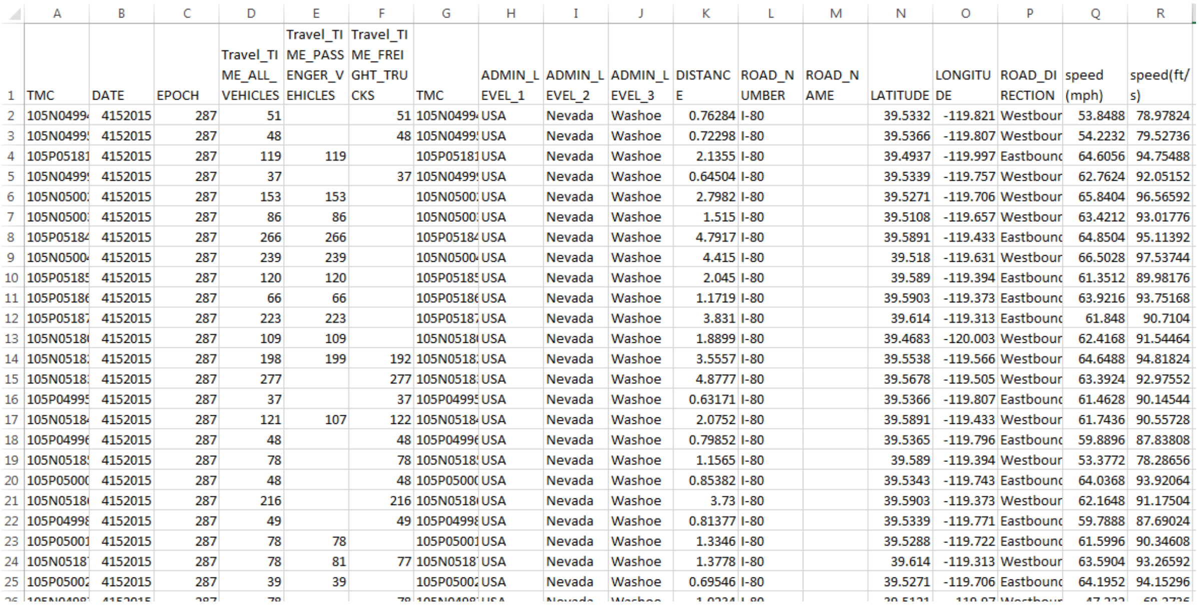
The values of “count_sensor_id” were defined in the input_link.csv file using a format of “10002_AB”, which is the link id (“10002”) joined with a two-letter label (“AB”). For each two-way link in the original TransCAD$^{®}$ network, NeXTA will create two associated links. Therefore, “_AB” and “_BA” should be added after the link id for each link. For one-way links, only “_AB” was added after each link id.



	A	B	C	D	E	F	G	H	I
1	count_sensor_id	day_no	unix_timestamp	start_time_in_min	end_time_in_min	link_count	occupancy	travel_time	avg_speed
2	10178_AB	1		360	420	589			
3	10178_AB	1		420	480	1051			
4	10178_AB	1		480	540	1049			
5	10178_BA	1		360	420	533			
6	10178_BA	1		420	480	1104			
7	10178_BA	1		480	540	1019			
8	10281_AB	1		360	420	2724			
9	10281_AB	1		420	480	4745			
10	10281_AB	1		480	540	4146			
11	10283_AB	1		360	420	2724			
12	10283_AB	1		420	480	4745			
13	10283_AB	1		480	540	4146			
14	10289_AB	1		360	420	181			
15	10289_AB	1		420	480	370			
16	10289_AB	1		480	540	378			
17	10291_AB	1		360	420	2608			
18	10291_AB	1		420	480	4842			
19	10291_AB	1		480	540	4366			

Figure 36. Sensor_count.csv file for a.m. peak

For travel time, data of NPMRDS was obtained. Figure 37 shows a sample of this data.



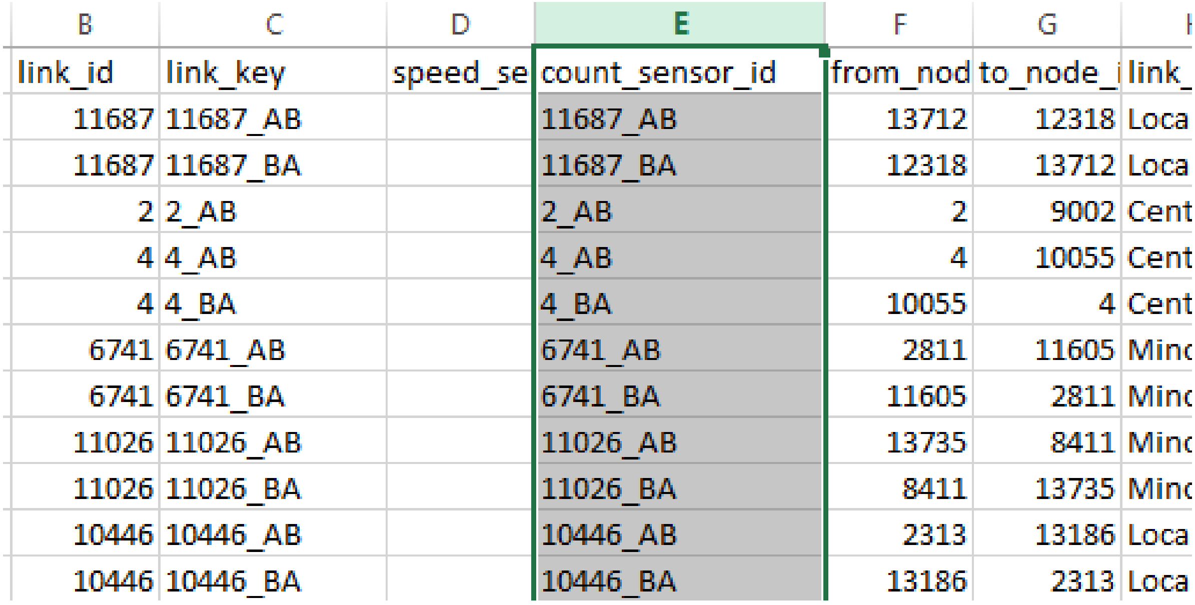
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	TMC	DATE	EPOCH	Travel_TIME_ALL_VEHICLES	Travel_TIME_PASSENGER_VEHICLES	Travel_TIME_FREIGHT_TRUCKS	TMC	ADMIN_LEVEL_1	ADMIN_LEVEL_2	ADMIN_LEVEL_3	DISTANCE	ROAD_NUMBER	ROAD_NAME	LATITUDE	LONGITUDE	ROAD_DIRECTION	speed (mph)	speed(ft/s)
2	105N0499	4152015	287	51		51	105N0499	USA	Nevada	Washoe	0.76284	I-80		39.5332	-119.821	Westbour	53.8488	78.97824
3	105N0499	4152015	287	48		48	105N0499	USA	Nevada	Washoe	0.72298	I-80		39.5366	-119.807	Westbour	54.2232	79.52736
4	105P05181	4152015	287	119	119		105P05181	USA	Nevada	Washoe	2.1355	I-80		39.4937	-119.997	Eastbound	64.6056	94.75488
5	105N0499	4152015	287	37		37	105N0499	USA	Nevada	Washoe	0.64504	I-80		39.5339	-119.757	Westbour	62.7624	92.05152
6	105N0500	4152015	287	153	153		105N0500	USA	Nevada	Washoe	2.7982	I-80		39.5271	-119.706	Westbour	65.8404	96.56592
7	105N0500	4152015	287	86	86		105N0500	USA	Nevada	Washoe	1.515	I-80		39.5108	-119.657	Westbour	63.4212	93.01776
8	105P05184	4152015	287	266	266		105P05184	USA	Nevada	Washoe	4.7917	I-80		39.5891	-119.433	Eastbound	64.8504	95.11392
9	105N0500	4152015	287	239	239		105N0500	USA	Nevada	Washoe	4.415	I-80		39.518	-119.631	Westbour	66.5028	97.53744
10	105P05185	4152015	287	120	120		105P05185	USA	Nevada	Washoe	2.045	I-80		39.589	-119.394	Eastbound	61.3512	89.98176
11	105P05186	4152015	287	66	66		105P05186	USA	Nevada	Washoe	1.1719	I-80		39.5903	-119.373	Eastbound	63.9216	93.75168
12	105P05187	4152015	287	223	223		105P05187	USA	Nevada	Washoe	3.831	I-80		39.614	-119.313	Eastbound	61.848	90.7104
13	105N0518	4152015	287	109	109		105N0518	USA	Nevada	Washoe	1.8899	I-80		39.4683	-120.003	Westbour	62.4168	91.54464
14	105N0518	4152015	287	198	199	192	105N0518	USA	Nevada	Washoe	3.5557	I-80		39.5538	-119.566	Westbour	64.6488	94.81824
15	105N0518	4152015	287	277		277	105N0518	USA	Nevada	Washoe	4.8777	I-80		39.5678	-119.505	Westbour	63.3924	92.97552
16	105P04995	4152015	287	37		37	105P04995	USA	Nevada	Washoe	0.63171	I-80		39.5366	-119.807	Eastbound	61.4628	90.14544
17	105N0518	4152015	287	121	107	122	105N0518	USA	Nevada	Washoe	2.0752	I-80		39.5891	-119.433	Westbour	61.7436	90.55728
18	105P04996	4152015	287	48		48	105P04996	USA	Nevada	Washoe	0.79852	I-80		39.5365	-119.796	Eastbound	59.8896	87.83808
19	105N0518	4152015	287	78		78	105N0518	USA	Nevada	Washoe	1.1565	I-80		39.589	-119.394	Westbour	53.3772	78.28656
20	105P0500	4152015	287	48		48	105P0500	USA	Nevada	Washoe	0.85382	I-80		39.5343	-119.743	Eastbound	64.0368	93.92064
21	105N0518	4152015	287	216		216	105N0518	USA	Nevada	Washoe	3.73	I-80		39.5903	-119.373	Westbour	62.1648	91.17504
22	105P04998	4152015	287	49		49	105P04998	USA	Nevada	Washoe	0.81377	I-80		39.5339	-119.771	Eastbound	59.7888	87.69024
23	105P05001	4152015	287	78	78		105P05001	USA	Nevada	Washoe	1.3346	I-80		39.5288	-119.722	Eastbound	61.5996	90.34608
24	105N0518	4152015	287	78	81	77	105N0518	USA	Nevada	Washoe	1.3778	I-80		39.614	-119.313	Westbour	63.5904	93.26592
25	105P0500	4152015	287	39	39		105P0500	USA	Nevada	Washoe	0.69546	I-80		39.5271	-119.706	Eastbound	64.1952	94.15296

Figure 37: NPMRDS travel time data

Currently, DTALite can use travel time only for validation – not for calibration. Since this data set has different link IDs and link configurations, travel times do not represent the current Reno/Sparks links. Therefore for validation, data set was used manually for some major links.

4.7.3. Modify "count_sensor_id" in the input_link.csv file

Accordingly, in the input_link.csv file, identical values of "count_sensor_id" should be defined as in the sensor_count.csv file (Figure 38).



B	C	D	E	F	G	H
link_id	link_key	speed_se	count_sensor_id	from_nod	to_node_i	link_
11687	11687_AB		11687_AB	13712	12318	Loca
11687	11687_BA		11687_BA	12318	13712	Loca
2	2_AB		2_AB	2	9002	Cent
4	4_AB		4_AB	4	10055	Cent
4	4_BA		4_BA	10055	4	Cent
6741	6741_AB		6741_AB	2811	11605	Minc
6741	6741_BA		6741_BA	11605	2811	Minc
11026	11026_AB		11026_AB	13735	8411	Minc
11026	11026_BA		11026_BA	8411	13735	Minc
10446	10446_AB		10446_AB	2313	13186	Loca
10446	10446_BA		10446_BA	13186	2313	Loca

Figure 38. Change input_link.csv file

4.7.4. Filter unreasonable or bad sensor data

Due to possible data incorrectness or possible inconsistency between the network profile and the sensor data, besides the process of inputting sensor data into the configuration files, there is a necessity of filtering “bad” sensor data. It requires engineering judgment to determine which data are unreasonable or incorrect so as to be ruled out. After analyzing the data set, some errors were caught, which were due to data entry mistakes.

Before running ODME for the initial calibration of the DTA model, these data records were modified.

4.7.5. Result of sensor data configuration

After the above configurations, sensors are associated with links. Figure 11 shows the location of sensors.

4.8. Run ODME Using Field Data for Calibration of the Whole Network

4.8.1. Methodology of ODME in DTALite

The core estimation variable of ODME in DTALite is trip production of each zone. ODME implemented in DTALite is based on a single-level nonlinear optimization model proposed by Lu et al [30]. This section explains this methodology.

This model has the following key features:

- The model is a path flow-based optimization model, which incorporates heterogeneous sources of traffic measurements and does not require explicit dynamic link-path incidences.
- The objective is to minimize (i) the deviation between observed and estimated traffic states and (ii) the deviation between aggregated path flows and target OD flows, subject to the dynamic user equilibrium (DUE) constraint represented by a gap-function-based reformulation.
- A Lagrangian relaxation-based algorithm which dualizes the difficult DUE constraint to the objective function is proposed to solve the model.
- This algorithm integrates a gradient-projection-based path flow adjustment method within a column generation-based framework.
- DTALite, a dynamic network loading (DNL) model which is based on Newell’s simplified kinematic wave theory, is employed in the DUE assignment process to realistically capture congestion phenomena and shock wave propagation.

- This optimization also derives analytical gradient formulas for the changes in link flow and density due to the unit change of time-dependent path inflow in a general network under congestion conditions.

Mathematical model

Given sensor data (i.e. observed link flows and densities) and target (aggregated historical) OD demands, the proposed single-level time-dependent path flow estimation model is a nonlinear program with the path flows $r(w,\tau,p), \forall w,\tau,p$ and least path travel times $\pi = \{\pi(w,\tau), w, \tau\}$ as the decision variables. Denote $c = \{c(w,\tau,p), \forall w,\tau,p\}$, $q = \{q(l,t), \forall l, t\}$ and $k = \{k(l,t), \forall l, t\}$. The objective function seen in Eq. 1, minimizes the weighted sum of the deviation between estimated time-dependent OD demands (or aggregated path flows) and target demands and the deviation between estimated and observed link flows and densities, where β_d, β_q and β_k are the weights reflecting different degrees of confidence on target OD demands and observed link flows and densities, respectively.

P1: Nonlinear program

$$Min\ Z = \beta_d \sum_w \left[\sum_{\tau \in H_d} \sum_p r(w,\tau,p) - \bar{d}(w)\right]^2 + \sum_{l \in S} \sum_{t \in H_0} \{\beta_q[q(l,t) - \bar{q}(l,t)]^2 + \beta_q\left[k(l,t) - \bar{k}(l,t)\right]^2\}$$

(1)

Subject to

$$(c,q,k) = DNLF(r)$$

(2)

$$g(r,\pi) = \sum_w \sum_\tau \sum_p \{r(w,\tau,p)[c(w,\tau,p) - \pi(w,\tau,p)]\} = 0$$

(3)

$$c(w,\tau,p) - \pi(w,\tau) \geq 0, \forall w,\tau,p$$

(4)

$$\pi(w,\tau) \geq 0, \forall p \in P(w,\tau), \forall w,\tau$$

(5)

$$r(w,\tau,p) \geq 0,, \forall w,\tau,p$$

(6)

where

A: set of links

W: set of OD pairs

P: set of paths

S: set of links with sensors, $S \subseteq A$

H_d: set of discretized departure time intervals

H_o: set of discretized observation time intervals

Index:

t: index of simulation time intervals, $t = 0, \ldots, T$. This chapter refers to any particular time interval t as the time t.

τ: index of departure time intervals, $\tau \in H_d$

w: index of OD pairs, $w \in W$

p: index of paths for each OD pair, $p \in P$

l: index of links, $l \in A$

Traffic measurements inputs

$\bar{q}(l,t)$: observed number of vehicles passing through an upstream detector on link l during observation interval t

$\bar{k}(l,t)$: observed density on link l during observation interval t

$\bar{d}(w)$: target demand, which is the total traffic demand for OD pair w over a planning horizon

Estimation variables:

$r(w,\tau,p)$: estimated path flow on path p of OD pair w and departure time interval τ

$c(w,\tau,p)$: estimated path travel time on path p of OD pair w and departure time interval τ

$\pi(w,\tau)$: estimated least path travel time of OD pair w and departure time interval τ

$q(l,t)$: estimated number of vehicles passing through an upstream detector on link l during observation interval t

$k(l,t)$: estimated density on link l during observation interval t

$d(w,\tau)$: estimated demand of OD pair w and departure time interval τ

Solution algorithm

This section describes the Lagrangian relaxation-based heuristic approach for solving the single-level time-dependent path flow estimation model. We propose the following heuristic solution method to efficiently obtain good solutions for problem instances on road networks of practical sizes. The heuristic integrates Lagrangian relaxation and column generation methods to solve the time-dependent path flow estimation model, P1. The gap function constraint (Eq. 3) is relaxed to the objective function (Eq. 1) with a non-negative Lagrange multiplier l. The resulting Lagrangian sub-problem is given as follows.

$$P_2: Min_{r,\pi} L(r,\pi,\lambda) = Z + \lambda\{g(r,\pi)\} \tag{7}$$

Subject to constraints (2), (4), (5) and (6),

where r and π are the vectors of path flows and least path times respectively. For a given l, the solution to P_2 provides a lower bound to P_1. The Lagrangian dual problem is given as follows:

$$P_3: Max_{\lambda} Min_{r,\pi} L(r,\pi,\lambda)$$

$$\text{Subject to } \lambda \geq 0 \tag{8}$$

The heuristic consists of two major algorithmic steps: at each iteration n, (i) given a Lagrange multiplier $\lambda(n)$, find an optimal path assignment $r(n)$ and least path travel times $\pi(n)$ by solving the Lagrangian sub-problem, P2, and (ii) given a vehicle path assignment $r(n)$ and least path travel times $\pi(n)$, update the Lagrange multiplier $\lambda(n+1)$by using the following rule.

$$\lambda(n+1) = \max\{0, \lambda(n) + \alpha(n)\{\sum_w \sum_\tau \sum_p r(w,\tau,p)[c(w,\tau,p) - \pi(w,\tau)]\}\} \tag{9}$$

where $\alpha(n)$ is the step size for updating the Lagrange multiplier.

Accordingly, this heuristic has two loops. The outer loop is for updating the Lagrange multiplier using the rule described in Eq. 9. For each outer loop iteration n (i.e., corresponding to a given Lagrange multiplier $\lambda(n)$), a column generation-based approach is used to solve the Lagrangian sub-problem P_2. This approach forms an inner loop for solving a DUE assignment problem under a restricted feasible solution space. In each inner loop iteration m, a time-dependent shortest path algorithm (Ziliaskopoulos and Mahmassani, [31]) is adopted to generate time-dependent least time paths and to augment the restricted path set. In light of the time-dependent shortest path algorithm, the least path travel times $\pi(m)$ are obtained to satisfy the constraints of Eqs. 4 and 5. Thus, these definitional constraints of the least travel times can be dropped in solving the restricted Lagrangian sub-problem. To solve the restricted sub-problem, a gradient-projection-based descent direction method (Lu et al., 32) is used to update path flows $r(m+1)$, while maintaining the feasibility of non-negativity constraints (Eq. 6). Specifically,

$$r(w,\tau,p)^{m+1} = Max\{0, r(w,\tau,p)^m - \gamma^m[\beta_d \nabla h^d(r)|_{r=r^m} + \beta_q \nabla h^q(r)|_{r=r^m} + \beta_k \nabla h^k(r)|_{r=r^m} + \lambda(n)\nabla g(r,\pi)|_{r=r^m}]\} \qquad (10)$$

where γ^m is the step size, and the gradients, which consist of the first-order partial derivatives with respect to a path flow variable $r(w,\tau,p)$, can be derived as follows.

$$\nabla h^d(r) = \frac{\partial\{\sum_{\tau\in H_d}\sum_{p\in P}(r_{w,\tau,p}) - \bar{d}_w\}}{\partial r(w,\tau,p)} = 2(\sum_{\tau\in H_d}\sum_{p\in P} r(w,\tau,p) - \bar{d}_w) \qquad (11)$$

$$\nabla h^q(r) = \frac{\partial\{\sum_{l\in S}\sum_{t\in H_o}[q_{l,t}(r) - \bar{q}(l,t)\}}{\partial r(w,\tau,p)} = 2(\sum_{l\in S}\sum_{t\in H_o}[q_{l,t}(r) - \bar{q}(l,t)] \times \frac{\partial q(l,t)(r)}{\partial r(w,\tau,p)}) \qquad (12)$$

$$\nabla h^k(r) = \frac{\partial\{\sum_{l\in S}\sum_{t\in H_o}[k_{l,t}(r) - \bar{k}(l,t)\}}{\partial r(w,\tau,p)} = 2(\sum_{l\in S}\sum_{t\in H_o}[k_{l,t}(r) - \bar{k}(l,t)] \times \frac{\partial k(l,t)(r)}{\partial r(w,\tau,p)}) \qquad (13)$$

$$\nabla g(r,\pi) = \frac{\partial g(r,\pi)}{\partial r(w,\tau,p)} = c(w,\tau,p) - \pi(w,\tau) + r(w,\tau,p)\frac{\partial c(w,\tau,p)}{\partial r(w,\tau,p)} \qquad (14)$$

Estimated link flows, densities, and link/path travel times and the corresponding partial derivatives, namely$\nabla h^q(r)$, $\nabla h^k(r)$ and $\frac{\partial c(w,\tau,p)}{\partial r(w,\tau,p)}$ are obtained from the DNL model presented previously.

The steps of this algorithm are presented as follows in Figure 39.

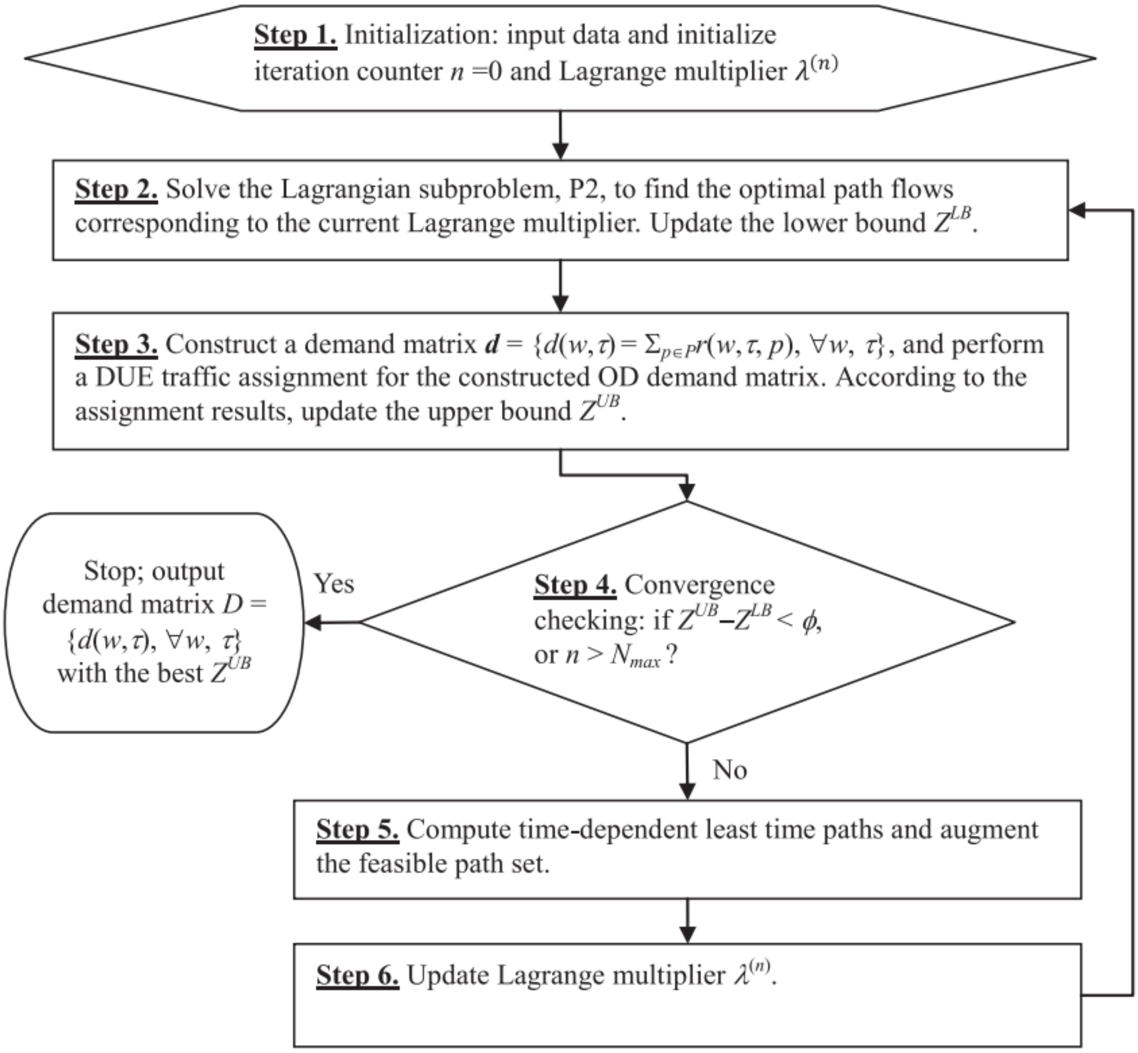

Step 1. Initialization: input data and initialize iteration counter n =0 and Lagrange multiplier $\lambda^{(n)}$
Step 2. Solve the Lagrangian subproblem, P2, to find the optimal path flows corresponding to the current Lagrange multiplier. Update the lower bound Z^{LB}.
Step 3. Construct a demand matrix $\boldsymbol{d} = \{d(w,\tau) = \Sigma_{p\in P} r(w,\tau,p), \forall w, \tau\}$, and perform a DUE traffic assignment for the constructed OD demand matrix. According to the assignment results, update the upper bound Z^{UB}.
Step 4. Convergence checking: if $Z^{UB}-Z^{LB} < \phi$, or $n > N_{max}$?
Yes
Stop; output demand matrix $D = \{d(w,\tau), \forall w, \tau\}$ with the best Z^{UB}
No
Step 5. Compute time-dependent least time paths and augment the feasible path set.
Step 6. Update Lagrange multiplier $\lambda^{(n)}$.

Figure 39: The procedure of proposed algorithm for ODME

Typically, a Lagrangian solution framework requires obtaining exact solutions to relaxed sub-problems. It should be remarked that, analyzing existence and uniqueness of solutions to the DUE problem for multiple OD pairs are very challenging, and the gradient-based algorithm through Eqs. 10-14 cannot guarantee that the relaxed (nonlinear) problem P2 is solved to its optimality. Thus, when no global optimum solution is available for P2, the proposed overall Lagrangian solution algorithm is still a heuristic method in nature.

Solving the proposed single-level dynamic OD estimation model requires the evaluation of the partial derivatives with respect to time-varying path flows, i.e., $\frac{\partial q(l,t)(r)}{\partial r(w,\tau,p)}$, $\frac{\partial k(l,t)(r)}{\partial r(w,\tau,p)}$ and $\frac{\partial c(w,\tau,p)}{\partial r(w,\tau,p)}$. These partial derivatives represent the marginal effects of an additional unit of path inflow on link flow and density, and path travel time. This section delineates the evaluation of these partial derivatives due to path flow perturbation in a congested network, based on cumulative link inflow and outflow curves. The following notation is used throughout this section.

L: the number of links on the path

l: link index $l = 1, 2, \ldots L$

${t_l}'$: the time when an additional unit of perturbation flow arrives at link l

t_l": the time when an additional unit of perturbation flow departs at link l

t_l^{qs}: the time when the queue starts to form on link l

t_l^B: the time when the queue vanishes on link l

t_l^A: $= t_l^B - FFTT(l)$

t_l^{q*}: the time when the queue on link l starts to spillback to its upstream link $l - 1$

n^A: cumulative arrivals at time t_l^A

n': cumulative arrivals at time ${t_l}'$

The following propositions can be directly induced from Figure 40 for deriving the marginal effects on link flow (inflow and outflow) and density.

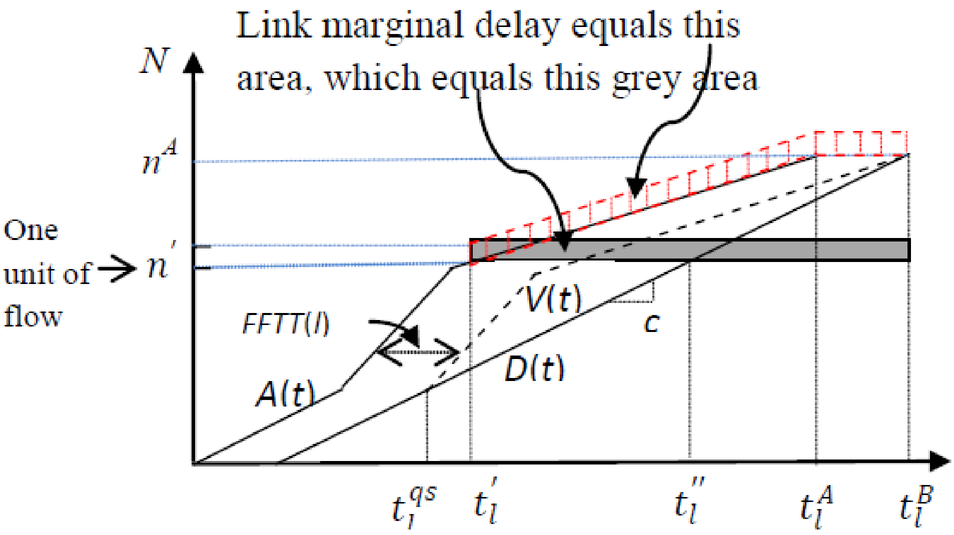


Figure 40: Impact of adding a unit of flow on a congested link

Proposition 1: Under free-flow conditions, an extra unit of flow arriving at the upstream end of link l at time t_l': results in the following: (i) the link inflow and outflow increase by 1 at times t_l' and t_l", respectively, and the flow rates at other time intervals do not change; (ii) the link density increases by 1 from t_l' to t_l"; (iii) the individual travel times are not changed, and $t_l" = t_l' + FFTT(l)$

Proposition 2: Under partially congested conditions and constant link (outflow) capacity c, an extra unit of flow arriving at the upstream end of link l at time t_l' results in the following: (i) the link inflow and outflow increase by 1 at times t_l' and t_l^B, respectively, and the flow rates at other time intervals do not change; (ii) the link density increases by 1 from t_l' and t_l^B ; (iii) the flows arriving between t_l' and t_l^A experience the additional delay $1/c$, because it takes $1/c$ to discharge this perturbation flow.

4.8.2. The Procedure of ODME in NeXTA

The ODME process fixes the original trip distribution ratio of each OD pair and the time profile used to define the percentage of total travel demand departing at some specific time periods. After running the simulation, users can check the calibrated trip production of each zone through ODME_ratio in file ODME_final_result.csv. In addition, the iteration process can also be checked in file ODME_zone_based_log.csv and file ODME_link_based_log.csv.

The key input and output files for ODME and scenario evaluation are listed in Table 14. The details of some files will be explained in the following sections.

Table 14: Key input and output files for ODME

File Group	Input file list	Remarks	Output file list	Remarks
Demand	1. input_demand.csv or input_agent.csv	Demand content for specifying distribution	1. output_agent.csv	Final estimation results represented in terms of agent trajectory format
	2. input_demand_file_list.csv	Demand file specification, specify temporal departure time distribution	2. ODME_zone_based_log.csv 3. ODME_final_result.csv	Zone based production based ratio; Iteration by iteration zone based total production adjustment
Sensor	1. input_link.csv	Store count_sensor_id, speed_sensor_id for referring sensor data	1. ODME_link_based_log.csv	Iteration by iteration link based flow adjustment
	2. sensor_count.csv	Link based sensor count data		
	3. sensor_speed.csv (optional)		2. debug_validation_results.csv	Link based simulated vs. observed results
Scenario	1. input_scenario_settings.csv	ODME and assignment settings	1. output_summary.csv	Iteration-by-iteration UE and ODME statistics
	2. DTASettings.txt	Default settings for sequential ODME run		
	3. scenario_work_zone.csv	Specify the capacity reduction scenarios in the estimation and prediction stages		

The general process of ODME is to adjust the given historical OD demand so that the final assignment results are consistent with observed link traffic measurements, such as, link count, link occupancy, link travel time, etc. In addition to the basic traffic network data (input_node.csv, input_link.csv, input_zone.csv, and input_activitity_location.csv), it also requires (1) an OD demand matrix seed as the initial demand values, (2) observed sensor data for calibration and (3) scenario settings files for algorithm performance. The demand seed could be a zone-to-zone demand file (such as, input_demand.csv) or an activity-based demand file (such as, input_agent.csv) with time profile in input_demand_file_list.csv. The sensor utilization involves sensor location information (input_link.csv and sensor_count.csv) and observed traffic data (sensor_count.csv and/or sensor_speed.csv). The specific settings need to be finished in input_secenario_settings.csv and DTASettings.txt, and if necessary, some scenario files also need to be set up, such as, scenario_work_zone.csv. After ODME and traffic assignment, each vehicle’s travel information is stored in output_agent.csv, which can also be treated as the estimated agent-based travel demand file. In addition, the link-based, zone-based and network-based statistics can be found in ODME_zone_based_log.csv, ODME_final_result.csv, ODME_link_based_log.csv, debug_validation_results.csv and output_summary.csv.

1) Data preparation

As described before, the required basic input data for ODME include the traffic network data (input_node.csv, input_link.csv, input_zone.csv, and input_activitity_location.csv), original or historical travel demand data (input_demand.csv and input_demand_file_list.csv), and observed sensor data (sensor_count.csv).

In this case, the observed link count from "link_count" is used for ODME, and the start_time_in_min and end_time_in_min defines the corresponding observation time period. If optional density data or travel time data are also available, users can also prepare them in "lane_density" or " travel_time_in_min" in this file. More importantly, to map sensors to links (in file input_link.csv), one of the following two methods can be used to specify a link with sensors: (i) "from_node_id" and "to_node_id", or (ii) field "count_sensor_id", which should be first defined in file input_link.csv. If not, warning messages will be issued.

2) Parameters settings and ODME running

The required parameter settings are defined in file input_scenario_settings.csv for varying traffic analysis purposes as previously shown in the Table 13.

In the DTASettings.txt file, you might see the following default settings:

```
[estimation]

number_of_iterations_per_sequential_adjustment=10

time_period_in_min_per_sequential_adjustment=60
```

The time_period_in_min_per_sequential_adjustment defines the time period of each sequential adjustment in the algorithm as 60 min (1 hour). The number_of_iterations_per_sequential_adjustment = 10 means that DTALite takes 10 iterations to adjust its demand during the time period of one sequential adjustment above (60 min or 1 hour). The calibrated demand period is based on minutes, for example a.m. peak period is from 360 to 540 minutes. To reach a reasonable convergence, at least 3-10 global iterations are usually needed, so at least 30 iterations are required for the path flow adjustment for ODME. Meanwhile, with the starting 20 iterations for reaching the user equilibrium state, a total of 50 iterations will be run for the Reno/Sparks network.

In addition, in order to improve the computational efficiency, users can change the value of the max_number_of_threads_to_be_used in DTASettings.txt to perform parallel computing in DTALite.

3) Result analysis

By checking the output_summary.csv file, users can better understand the process of ODME in DTALite. For the first 20 iterations, a standard dynamic user equilibrium method, MAS, is used. It is expected to see the UE gap (Avg User Equilibirum (UE) gap (min) and Relative UE gap (%)) dramatically decrease and finally reach a stable state, which is shown in Figure 41. In addition, Figure 42 gives the general statistics about different traffic measures for the first 20 iterations.

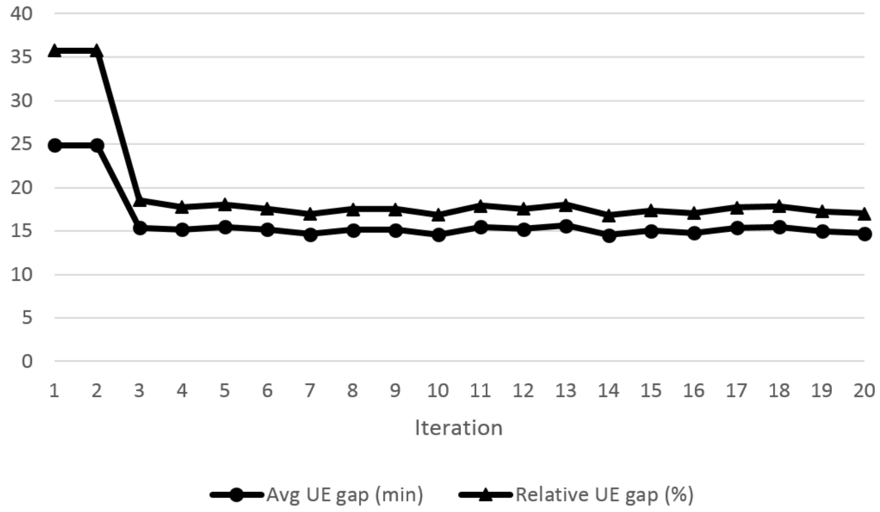
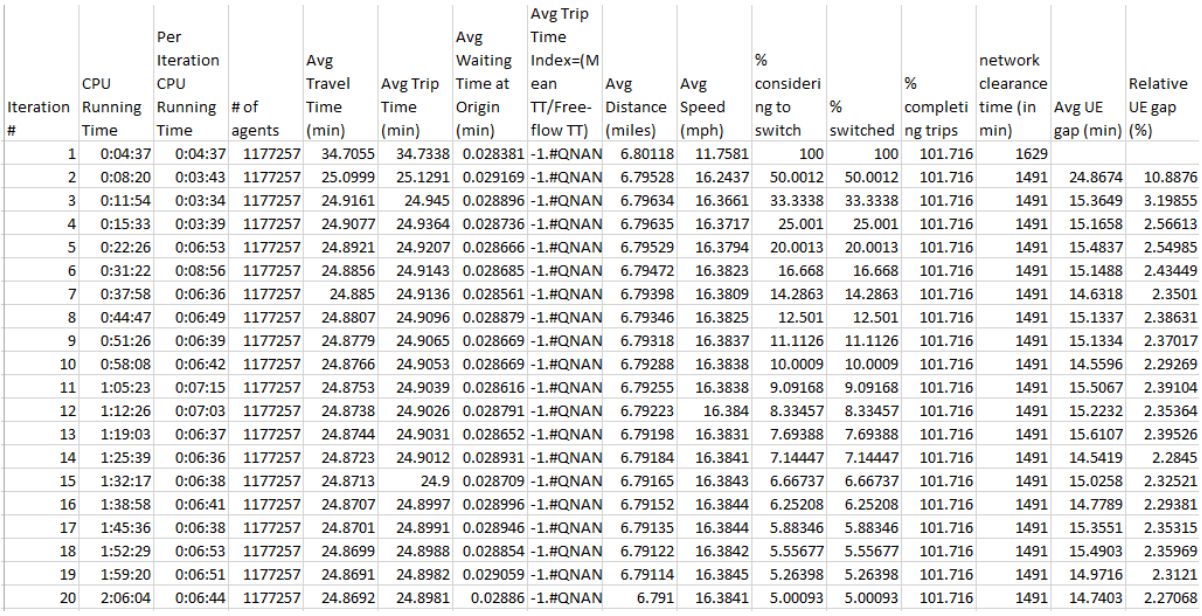


Figure 41: The trend of average UE gap and relative UE gap

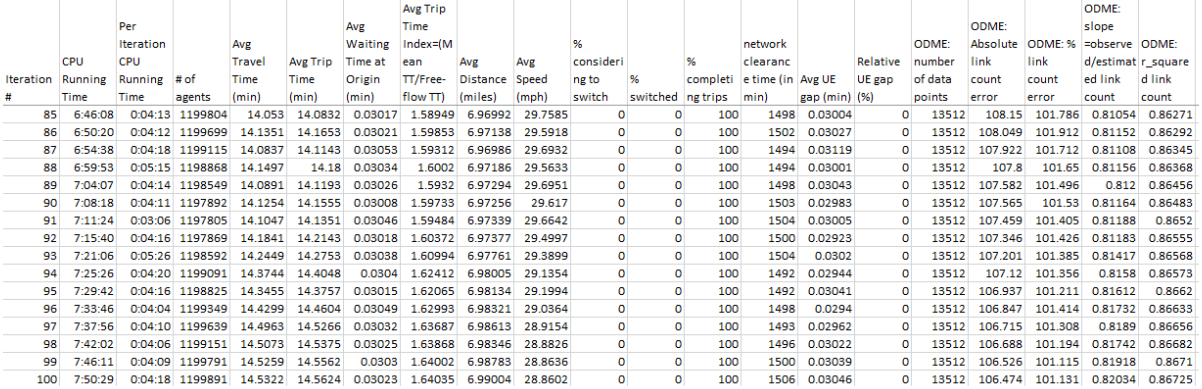


Iteration #	CPU Running Time	Per Iteration CPU Running Time	# of agents	Avg Travel Time (min)	Avg Trip Time (min)	Avg Waiting Time at Origin (min)	Avg Trip Time Index=(Mean TT/Free-flow TT)	Avg Distance (miles)	Avg Speed (mph)	% considering to switch	% switched	% completing trips	network clearance time (in min)	Avg UE gap (min)	Relative UE gap (%)
1	0:04:37	0:04:37	1177257	34.7055	34.7338	0.028381	-1.#QNAN	6.80118	11.7581	100	100	101.716	1629		
2	0:08:20	0:03:43	1177257	25.0999	25.1291	0.029169	-1.#QNAN	6.79528	16.2437	50.0012	50.0012	101.716	1491	24.8674	10.8876
3	0:11:54	0:03:34	1177257	24.9161	24.945	0.028896	-1.#QNAN	6.79634	16.3661	33.3338	33.3338	101.716	1491	15.3649	3.19855
4	0:15:33	0:03:39	1177257	24.9077	24.9364	0.028736	-1.#QNAN	6.79635	16.3717	25.001	25.001	101.716	1491	15.1658	2.56613
5	0:22:26	0:06:53	1177257	24.8921	24.9207	0.028666	-1.#QNAN	6.79529	16.3794	20.0013	20.0013	101.716	1491	15.4837	2.54985
6	0:31:22	0:08:56	1177257	24.8856	24.9143	0.028685	-1.#QNAN	6.79472	16.3823	16.668	16.668	101.716	1491	15.1488	2.43449
7	0:37:58	0:06:36	1177257	24.885	24.9136	0.028561	-1.#QNAN	6.79398	16.3809	14.2863	14.2863	101.716	1491	14.6318	2.3501
8	0:44:47	0:06:49	1177257	24.8807	24.9096	0.028879	-1.#QNAN	6.79346	16.3825	12.501	12.501	101.716	1491	15.1337	2.38631
9	0:51:26	0:06:39	1177257	24.8779	24.9065	0.028669	-1.#QNAN	6.79318	16.3837	11.1126	11.1126	101.716	1491	15.1334	2.37017
10	0:58:08	0:06:42	1177257	24.8766	24.9053	0.028669	-1.#QNAN	6.79288	16.3838	10.0009	10.0009	101.716	1491	14.5596	2.29269
11	1:05:23	0:07:15	1177257	24.8753	24.9039	0.028616	-1.#QNAN	6.79255	16.3838	9.09168	9.09168	101.716	1491	15.5067	2.39104
12	1:12:26	0:07:03	1177257	24.8738	24.9026	0.028791	-1.#QNAN	6.79223	16.384	8.33457	8.33457	101.716	1491	15.2232	2.35364
13	1:19:03	0:06:37	1177257	24.8744	24.9031	0.028652	-1.#QNAN	6.79198	16.3831	7.69388	7.69388	101.716	1491	15.6107	2.39526
14	1:25:39	0:06:36	1177257	24.8723	24.9012	0.028931	-1.#QNAN	6.79184	16.3841	7.14447	7.14447	101.716	1491	14.5419	2.2845
15	1:32:17	0:06:38	1177257	24.8713	24.9	0.028709	-1.#QNAN	6.79165	16.3843	6.66737	6.66737	101.716	1491	15.0258	2.32521
16	1:38:58	0:06:41	1177257	24.8707	24.8997	0.028996	-1.#QNAN	6.79152	16.3844	6.25208	6.25208	101.716	1491	14.7789	2.29381
17	1:45:36	0:06:38	1177257	24.8701	24.8991	0.028946	-1.#QNAN	6.79135	16.3844	5.88346	5.88346	101.716	1491	15.3551	2.35315
18	1:52:29	0:06:53	1177257	24.8699	24.8988	0.028854	-1.#QNAN	6.79122	16.3842	5.55677	5.55677	101.716	1491	15.4903	2.35969
19	1:59:20	0:06:51	1177257	24.8691	24.8982	0.029059	-1.#QNAN	6.79114	16.3845	5.26398	5.26398	101.716	1491	14.9716	2.3121
20	2:06:04	0:06:44	1177257	24.8692	24.8981	0.02886	-1.#QNAN	6.791	16.3841	5.00093	5.00093	101.716	1491	14.7403	2.27068

Figure 42: The statistics of different traffic measurements for the first 20 iterations

In the following 70 iterations, users can check the R_squared values from the iterative adjustment process, which should show an increasing pattern toward reasonable statistics of

0.7, 0.8, or 0.9 in output_summary.csv. The other measures related to ODME include the link count estimation absolute and percentage errors. The summary result of the last 15 iterations is shown in Figure 43, where the R_squared values are marked in the red rectangle.



Iteration #	CPU Running Time	Per Iteration CPU Running Time	# of agents	Avg Travel Time (min)	Avg Trip Time (min)	Avg Waiting Time at Origin (min)	Avg Trip Time Index=(Mean TT/Free-flow TT)	Avg Distance (miles)	Avg Speed (mph)	% considering to switch	% switched	% completing trips	network clearance time (in min)	Avg UE gap (min)	Relative UE gap (%)	ODME: number of data points	ODME: Absolute link count error	ODME: % link count error	ODME: slope =observed/estimated link count	ODME: r_squared link count
85	6:46:08	0:04:13	1199804	14.053	14.0832	0.03017	1.58949	6.96992	29.7585	0	0	100	1498	0.03004	0	13512	108.15	101.786	0.81054	0.86271
86	6:50:20	0:04:12	1199699	14.1351	14.1653	0.03021	1.59853	6.97138	29.5918	0	0	100	1502	0.03027	0	13512	108.049	101.912	0.81152	0.86292
87	6:54:38	0:04:18	1199115	14.0837	14.1143	0.03053	1.59312	6.96986	29.6932	0	0	100	1494	0.03119	0	13512	107.922	101.712	0.81108	0.86345
88	6:59:53	0:05:15	1198868	14.1497	14.18	0.03034	1.6002	6.97186	29.5633	0	0	100	1494	0.03001	0	13512	107.8	101.65	0.81156	0.86368
89	7:04:07	0:04:14	1198549	14.0891	14.1193	0.03026	1.5932	6.97294	29.6951	0	0	100	1498	0.03043	0	13512	107.582	101.496	0.812	0.86456
90	7:08:18	0:04:11	1197892	14.1254	14.1555	0.03008	1.59733	6.97256	29.617	0	0	100	1503	0.02983	0	13512	107.565	101.53	0.81164	0.86483
91	7:11:24	0:03:06	1197805	14.1047	14.1351	0.03046	1.59484	6.97339	29.6642	0	0	100	1504	0.03005	0	13512	107.459	101.405	0.81188	0.8652
92	7:15:40	0:04:16	1197869	14.1841	14.2143	0.03018	1.60372	6.97377	29.4997	0	0	100	1500	0.02923	0	13512	107.346	101.426	0.81183	0.86555
93	7:21:06	0:05:26	1198592	14.2449	14.2753	0.03038	1.60994	6.97761	29.3899	0	0	100	1504	0.0302	0	13512	107.201	101.385	0.81417	0.86568
94	7:25:26	0:04:20	1199091	14.3744	14.4048	0.0304	1.62412	6.98005	29.1354	0	0	100	1492	0.02944	0	13512	107.12	101.356	0.8158	0.86573
95	7:29:42	0:04:16	1198825	14.3455	14.3757	0.03015	1.62065	6.98134	29.1994	0	0	100	1492	0.03041	0	13512	106.937	101.211	0.81612	0.8662
96	7:33:46	0:04:04	1199349	14.4299	14.4604	0.03049	1.62993	6.98321	29.0364	0	0	100	1498	0.0294	0	13512	106.847	101.414	0.81732	0.86633
97	7:37:56	0:04:10	1199639	14.4963	14.5266	0.03032	1.63687	6.98613	28.9154	0	0	100	1493	0.02962	0	13512	106.715	101.308	0.8189	0.86656
98	7:42:02	0:04:06	1199151	14.5073	14.5375	0.03025	1.63868	6.98346	28.8826	0	0	100	1496	0.03022	0	13512	106.688	101.194	0.81742	0.86682
99	7:46:11	0:04:09	1199791	14.5259	14.5562	0.0303	1.64002	6.98783	28.8636	0	0	100	1500	0.03039	0	13512	106.526	101.115	0.81918	0.8671
100	7:50:29	0:04:18	1199891	14.5322	14.5624	0.03023	1.64035	6.99004	28.8602	0	0	100	1506	0.03046	0	13512	106.474	101.131	0.82034	0.86725

Figure 43: The summary result of the last 15 iterations. Last column shows R-squared

The comparison between simulation link measurements and observed link measurements is available in file debug_validation_results.csv. Figure 44 illustrates the comparison result in link counts. This figure is built in Excel based on the data provided in the debug_validation_results.csv file. Users can also check the comparison result in NeXTA through clicking “Tools”→ “Sensor Data Management” → “View Validation Plot for Link Count”, as shown in Figure 45. The black, purple, and blue dots represent freeway, arterial and ramps, respectively.

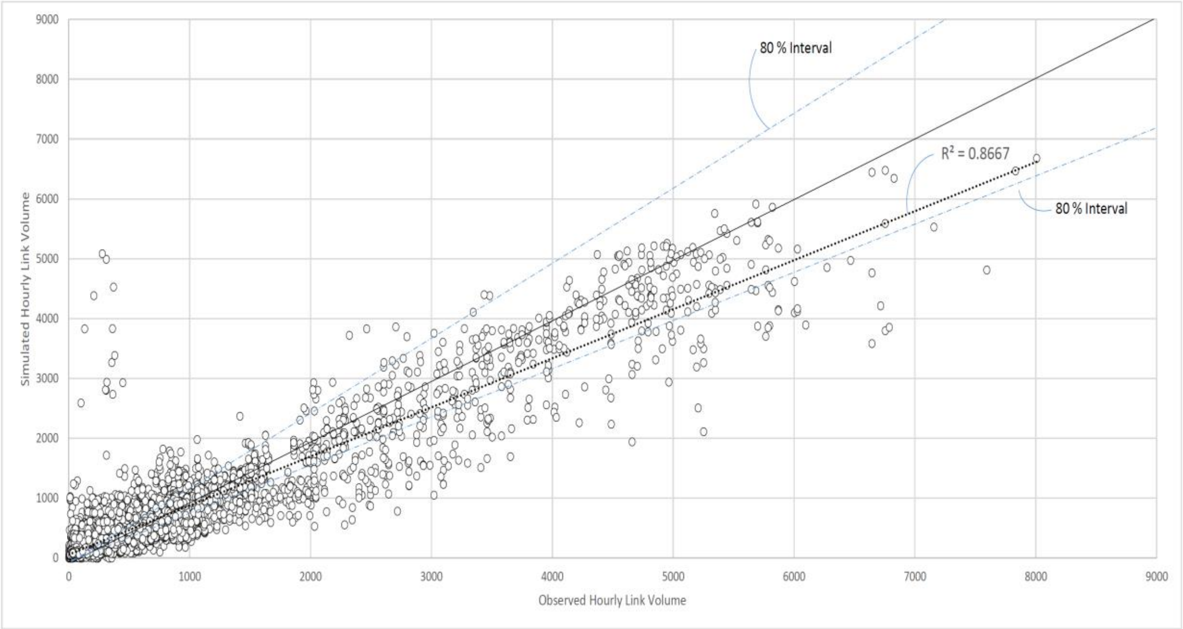


Figure 44: Comparison result between observed link count and simulated link count after ODME in Excel

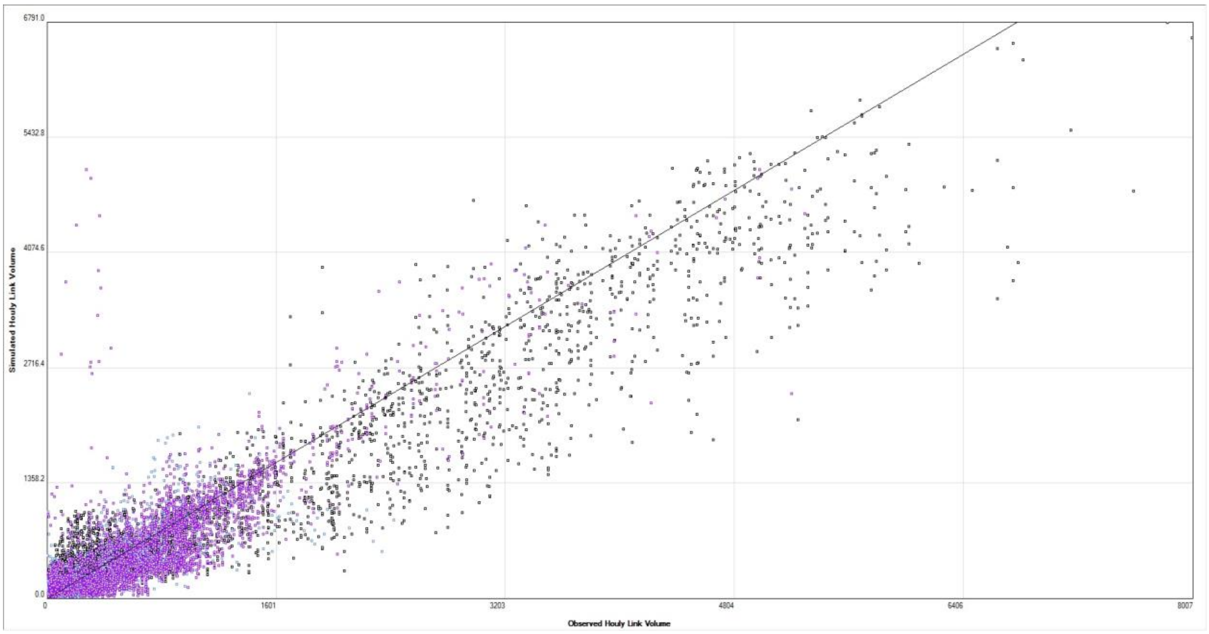


Figure 45: Comparison result between observed link count and simulated link count after ODME in NeXTA

4.9. Further Analysis in Micro-Simulation Software Packages

Travel demand forecasting models such as TransCAD have been used by MPOs and state DOTs for planning future transportation systems. Most travel demand models only provide macroscopic level analysis that is generally not sufficient to capture the operational details needed for microscopic level analysis. For example, no travel demand model can so far provide accurate operational analysis for signalized intersections by taking detailed signal timing and turn-lane pockets into consideration. In these models, freeway operations are generally assessed based on static empirical equations consisting of variables such as number of lanes, free-flow speed, and vehicle composition.

Over the past two decades, computing technology has significantly advanced. Using microscopic traffic simulation models is a common practice when large-scale design projects are involved. Microscopic simulation tools, such as Synchro/SimTraffic, account for the movements of individual vehicles dynamically and stochastically in the network on a second-by-second basis and provide more detailed information for microscopic level analysis. However, constructing a microscopic simulation model from scratch still consumes a significant amount of resources. The reason for adopting the macroscopic level analysis in travel demand models is mainly due to constraints imposed by modeling large-scale transportation networks. For traffic operations analysis and design, travel demand models do not satisfy the needed details.

In practice, there has been a disconnection between planners and traffic engineers. Transportation planners who perform travel demand forecasts rarely get into the operational level. Similarly, traffic engineers who conduct operational analysis using forecasted traffic demands often do not know the underlying principles and constraints of travel demand models. Although planning agencies always desire detailed traffic operational analyses, the significant effort involved in constructing a traffic simulation model often prohibits such detailed simulation modeling.

State and local agencies often conduct corridor studies where detailed traffic operational analyses are desired. Due to major efforts involved in constructing a traffic simulation model, corridor studies are generally kept at the Highway Capacity Manual (HCM) procedure level. Travel demand models contain most of the data needed for constructing traffic simulation models. Additional data can be added to most travel demand models using their data management tools. Expanding travel demand models to microscopic simulation is anticipated to be a major trend in future transportation studies. This section addresses this need.

Though many tools have been developed in this field, they are usually dedicated to a single type of resolution, which cannot satisfy the various modeling needs. Therefore, cross-resolution conversion has become a hot topic in linking traffic planning models and traffic simulation models.

Though each resolution tool has its own advantages and applicable areas, it seems that none of them alone is capable of conducting subarea studies. Therefore, it is of high importance to develop a multi-resolution simulation tool that can fulfill the need to conduct subarea studies. However, it is reasonable to take advantage of the single resolution tools rather than building one from scratch.

The following section provides detailed steps from a TransCAD model to a subarea simulation in Synchro.

The software needed in this process include Caliper's TransCAD, NeXTA/DTALite package, and TrafficWare's Synchro. Users should install these software before attempting to convert any network.

4.9.1. Step 1: DTA model development

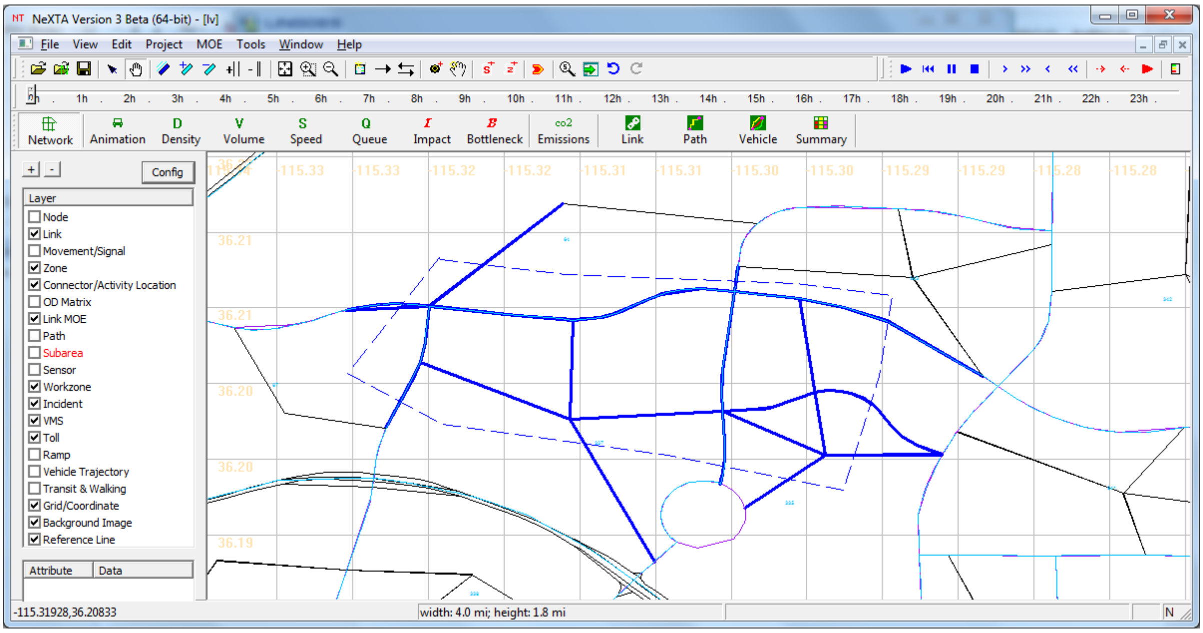
This step was explained at previous sections.

4.9.2. Step 2: subarea extraction and conversion

Before converting the DTA model to Synchro, the user needs to define a subarea or corridor that fits Synchro's simulation ability. The original Reno/Sparks network is too large for Synchro to handle and impractical for microscopic simulation analysis. NeXTA simplifies the subarea creation process by automatically handling extraction of necessary nodes, links, zones, and O-D tables.

Cut a subarea within the larger network

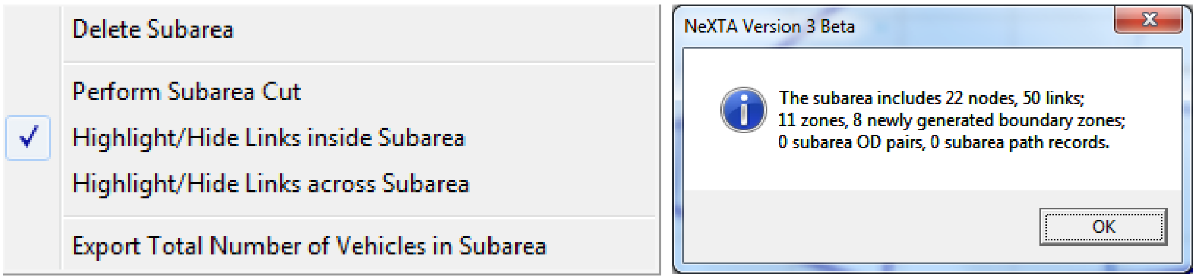
Using the Edit→Create Subarea tool in NeXTA, a subarea boundary can be drawn around the subarea of interest. The boundary must be a closed polygon. Figure 46 shows the defined subarea. The links and nodes within the boundary are highlighted, which allows a visual assessment of the boundary so that adjustments can be made if needed.


NeXTA Version 3 Beta (64-bit) - [lv]
File View Edit Project MOE Tools Window Help
Network Animation Density Volume Speed Queue Impact Bottleneck Emissions Link Path Vehicle Summary
Config
Attribute Data
-115.31928,36.20833
width: 4.0 mi; height: 1.8 mi

Layer
☐ Node
☑ Link
☐ Movement/Signal
☑ Zone
☑ Connector/Activity Location
☐ OD Matrix
☑ Link MOE
☐ Path
☐ Subarea
☐ Sensor
☑ Workzone
☑ Incident
☑ VMS
☑ Toll
☐ Ramp
☐ Vehicle Trajectory
☐ Transit & Walking
☑ Grid/Coordinate
☑ Background Image
☑ Reference Line

Figure 46: Illustration of Subarea Boundary

Right click in the main display frame and then select Perform Subarea Cut (Figure 47a); the tool will automatically remove all of the network objects (nodes and links) outside of the subarea boundary and extracted links, nodes, zones, O-D pairs, and subarea path records. A message window will pop out when subarea cut is finished (Figure 47b).


Delete Subarea
Perform Subarea Cut
✓ Highlight/Hide Links inside Subarea
Highlight/Hide Links across Subarea
Export Total Number of Vehicles in Subarea
NeXTA Version 3 Beta
OK

The subarea includes 22 nodes, 50 links;
11 zones, 8 newly generated boundary zones;
0 subarea OD pairs, 0 subarea path records.

(a) The "Perform Subarea Cut" Button

(b) Subarea Cut Result Window

Figure 47: Subarea Cut in NeXTA

The subarea needs to be saved as a new project in a separate folder.

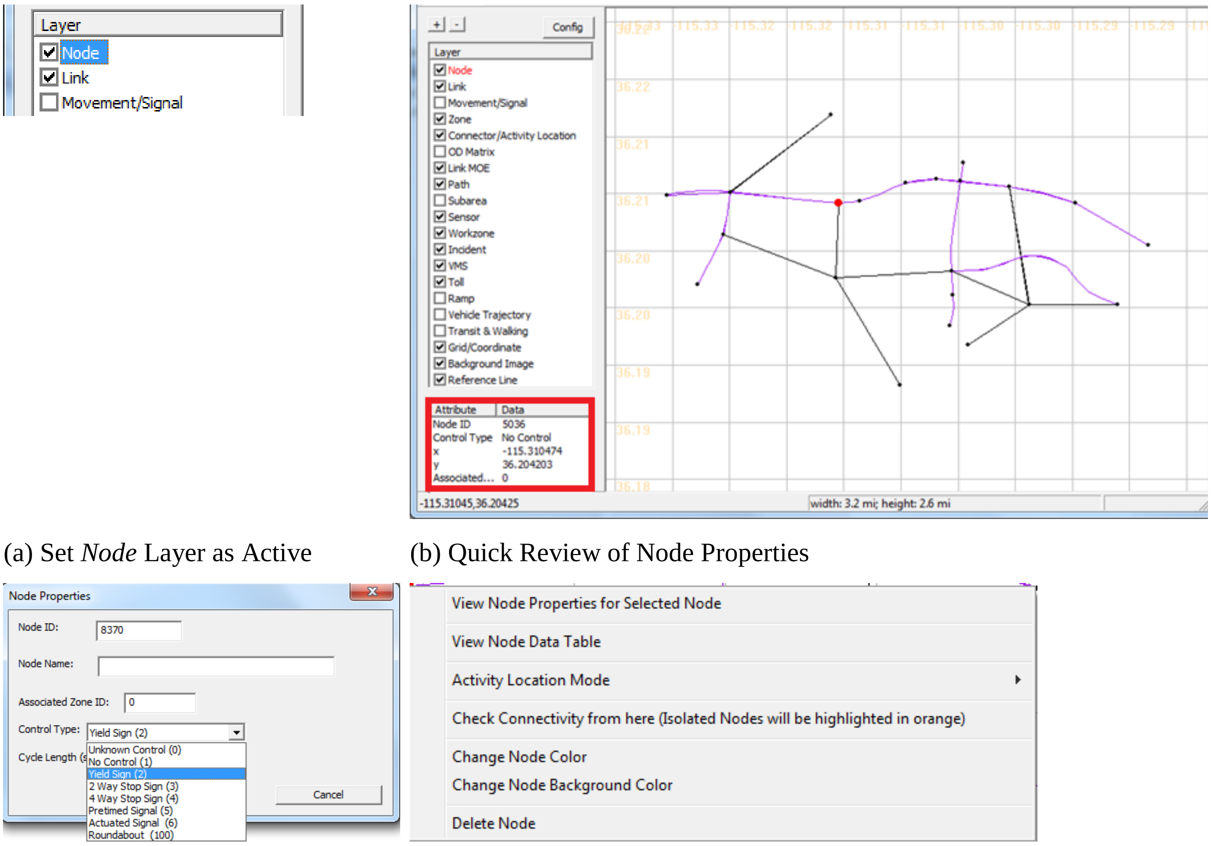
Generate physical zone centroids on road network

The Generate Physical Zone Centroids on Road Network Tool under Tools→Network Tools in NeXTA converts the zonal connectors to side streets within the network. This tool replaces zone centroids with additional nodes so that no paths can be routed through a zone centroid. While DTALite cannot use paths through zone centroids, other AMS software tools such as Synchro and PTV VISSIM do not make such distinctions. Executing this command ensures that the resulting network is compatible with Synchro.

Modify node properties

Depending on the level of details in the TransCAD model, modification of node properties, such as node control type and cycle length, may be needed. The current Reno/Sparks network does not contain node control type, therefore in the conversion process, the user needs to specify node control type for all the nodes within the subarea.

To check and modify node properties, the user needs to turn on *Node* layer (by checking the box) the layer manager frame and make it active (by clicking on *Node*), as shown in Figure 48a. Check node attributes by left clicking on any node. The attributes will display in the lower left frame (Figure 48b). To change the attributes of a node, right click on the node and then select View Node Properties for Selected Node in the menu (Figure 48c). In the Node Properties windows, users can change Control Type to reflect field practice (Figure 48d).



Attribute	Data
Node ID	5036
Control Type	No Control
x	-115.310474
y	36.204203
Associated...	0

Node ID: 8370

Node Name:

Associated Zone ID: 0

Control Type: Yield Sign (2)

Cycle Length (s

(a) Set *Node* Layer as Active

(b) Quick Review of Node Properties

(d) Node Properties Window

(c) Right-click Menu for a Node

Figure 48: Modify Node Properties in NeXTA

Re-run DTALite for the subarea

To ensure a consistent vehicle path assignment within the subarea, dynamic traffic assignment should be run again on the subarea network. This can be done by clicking the Run Simulation button ⮚. Simulation settings can be reviewed and users can choose preferred models. After clicking the button, DTALite will run and perform dynamic traffic assignment for the subarea.

Subarea calibration (optional step)

If field data availability (e.g. link volumes) allows, it is recommended to perform model calibration for the subarea, using ODME. The objective of ODME is to better match simulated link volumes to observed field data, as was explained in the previous sections.

4.9.3. Step 3: Export subarea to Synchro

After dynamic traffic assignment (and ODME process) is done for the subarea, the subarea is ready to be exported to Synchro for further analysis. Since a typical TransCAD model does not contain signal information, signal phasing and timing can be approximated using QEM. The procedure for exporting a subarea network for microscopic analysis is as follows.

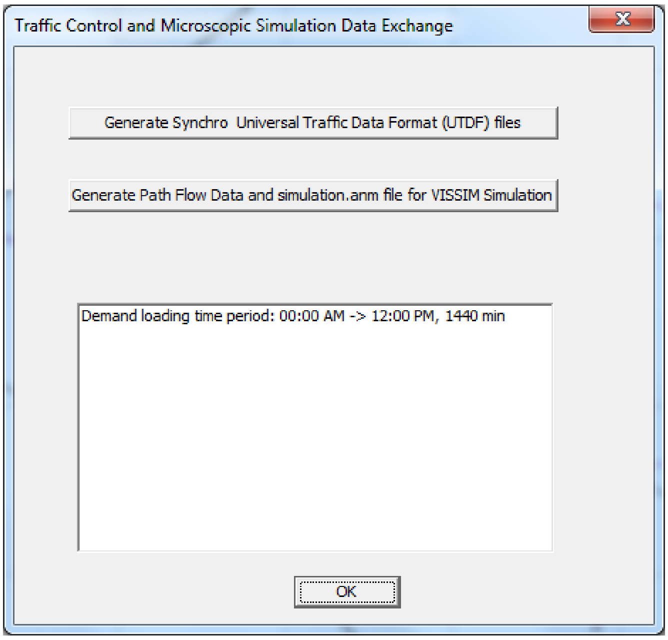
Use QEM to estimate initial signals

Using QEM was explained in Section 3.3.

Export to Universal Traffic Data Format (UTDF) files

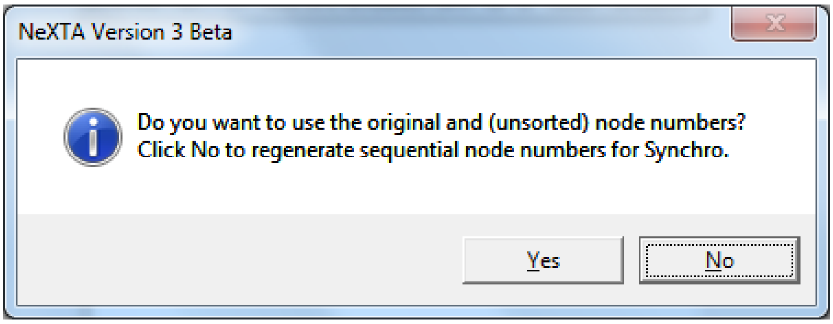
NeXTA is capable of writing its network data in UTDF that is compatible with Synchro. When QEM analysis finishes, re-open the subarea file in NeXTA.

Go to menu *File – Export – Microscopic Network and Traffic Control Data*, and choose "*Generate Synchro Universal Traffic Data Format files*" (Figure 49a). In the next pop-up dialog (Figure 49b), click "No" to generate sequential node numbers.


Traffic Control and Microscopic Simulation Data Exchange
Generate Synchro Universal Traffic Data Format (UTDF) files
Generate Path Flow Data and simulation.anm file for VISSIM Simulation
OK

Demand loading time period: 00:00 AM -> 12:00 PM, 1440 min

(a)



Do you want to use the original and (unsorted) node numbers?
Click No to regenerate sequential node numbers for Synchro.

(b)

Figure 49: Export Subarea to UTDF Format

Conversion files will be stored in a folder named Exporting_Synchro_UTDF under the subarea project folder.

Import UTDF files into Synchro

The import should be carried out using the import tool under Synchro menu ***Transfer* — *Data Access* — *Version 6 data Access*** (Figure 50).

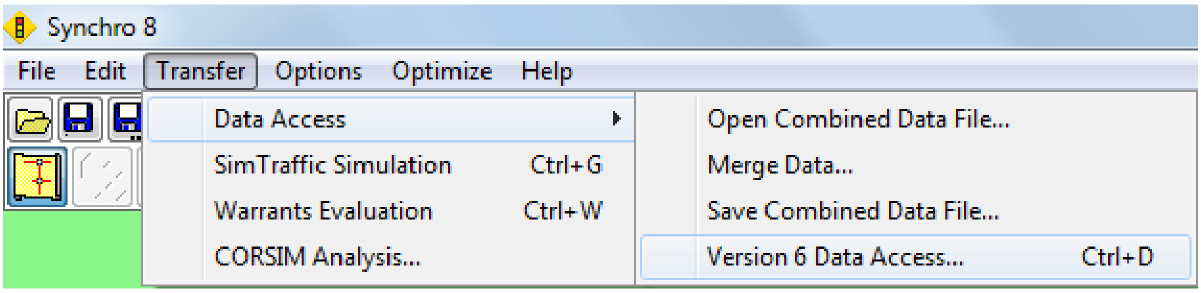

Synchro 8
File
Edit
Transfer
Options
Optimize
Help
Data Access
SimTraffic Simulation Ctrl+G
Warrants Evaluation Ctrl+W
CORSIM Analysis...
Open Combined Data File...
Merge Data...
Save Combined Data File...
Version 6 Data Access... Ctrl+D

Figure 50: Import the Subarea into Synchro

The imported network in Synchro is shown in Figure 51.

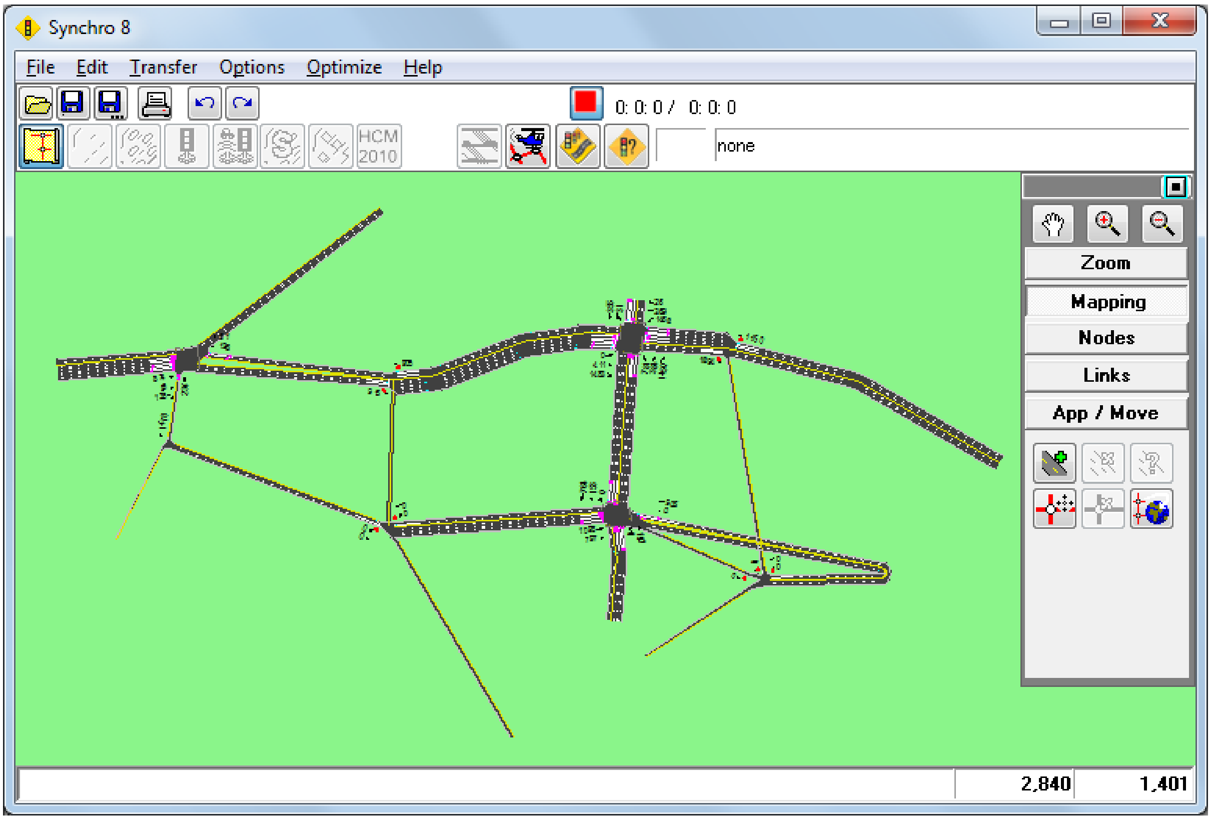


Figure 51: Screen shot of imported network in Synchro

Additional adjustments in Synchro

Manual adjustments may be needed in Synchro before running simulation in SimTraffic. These adjustments, depending on the specific network, may include but are not limited to:

1) Geometry modification, e.g. node-link connection, lane configuration, turning pockets configuration, and side street modification;

2) Turning volumes balancing, and

3) Signal optimization.

These adjustments help the network reflect real traffic conditions as closely as possible, so the resulting simulation would best present traffic performance.

5. MODEL VALIDATION AND CALIBRATION

In the previous chapter, the model was run for initial demand calibration using ODME. However, it is possible to improve the model by using other mechanisms.

The process of validation (verification) compares the model outputs to the observed traffic conditions such as traffic counts and speeds to assess the quality of model outputs. Model calibration involves the identification of a set of DTA model inputs and parameters that results in model outputs that are reasonably close to those field observations. Traffic data can come from various sources with varying detection technologies such as loop detectors, acoustic sensors or video-based detections. Automatic vehicle identification technologies commonly used in toll collection are also useful for O-D calibration in addition to point-to-point count and travel time data.

The measured traffic data can represent a wide range of quantities, some of which are [3]:

- Vehicle counts (by link or by lane) measured at detector locations;
- Average vehicle speeds at detector locations;
- Average link or segment density or detector occupancy;
- Queue lengths;
- Link or sub-route travel times; and
- Intersection turning movement counts.

Since a DTA model's outputs vary in time, the data used to calibrate and validate the DTA must also be dynamic. Thus, the various data listed above must be collected at a relatively fine temporal resolution (such as every 5 or 15 min). The data collection interval must also be compatible with the desired modeling time interval selected for the particular application. For example, if the project goal is to model the hourly variations in demand and network performance, then hourly data should be collected at a minimum. Theoretically, the finer time resolution that the data are in, the better the calibrated model represents the real-world situation. However, one needs to exercise several cautions as noted below [3].

Data sets collected from different sources may be expected to show some degree of inconsistency between them. Inconsistency can occur for various reasons. Data from different days, weeks, months, or years may reflect different demand levels, trip patterns and network–infrastructure conditions such as work zones. Even on the same day, some sensors may introduce measurement errors due to malfunction or failure. Another issue is to understand the source of the data. Hourly traffic counts generated from daily counts using hourly multipliers are not as accurate as actual hourly counts from the field. It is thus worthwhile to do some consistency checking in addition to the usual data cleaning procedures. The ideal case is to be able to make direct comparisons based on overlap between two data sets if they cover some of the same links or turning movements. Basic statistical tests, such as linear regression analyses, can be used to quantify the goodness of fit between two overlapping subsets. Even a relatively small sample of overlapping data can give a general indication of the consistency between the data sets as a whole. Due to the day-to-day variability of traffic conditions, care needs to be exercised when using traffic data for model validation and calibration. One commonly used approach is to use traffic data that is averaged over a number of days, and those days should be selected in such a way that they are representative of the O-D demands in the model or scenario. Note, however, that the presence of unrecorded major disruptions such as severe incidents or weather effects across days can cause significant bias if the data from these days are simply averaged. In some cases, traffic simply fluctuates to a large extent from day to day, and simply taking the average would smooth out the congestion (worse case) and thereby would leave out valuable information [3].

The process of initially validating and subsequently calibrating a DTA model can be broken down into two sequential analysis stages: qualitative and quantitative. The qualitative analysis stage (also referred to as preliminary validation) is what typically starts after the very first model runs, when there may still be many errors in the input data to be found. In these situations, it may be of little practical value to begin comparing the model outputs to empirical data, especially if the model is not converging to a stable solution. Once the model has been improved to a certain extent, the quantitative analysis (also referred to as calibration) starts. Quantitative analysis is based on a direct comparison of model outputs and empirical data, and investigating the outliers in order to further refine the model.

5.1. Preliminary Qualitative Analysis

There are some indicators that can show errors in the model. Generally, errors in the coding or inputting of network and traffic signal data are found to produce outputs showing more congestion rather than less. Because of the phenomenon of congestion spillback, a queue grows in space and engulfs vehicles that do not directly contribute to the original cause of the queue (they will turn off the road before reaching the downstream bottleneck). In extreme cases, queues that are initially separate grow into one, causing congestion to grow even faster and spread out in many directions. This can cause an entire section of a network to be engulfed in heavy congestion [3]. One example was explained in the previous chapter. Originally, it was thought that adding intersection links (Figure 25) would increase the accuracy of the model. However, as mentioned before, it caused unrealistic congestion due to the first-in first-out rule employed in DTALite. There is a need to identify the initial bottlenecks behind the heavy congestion and to fix input errors that result in unrealistic capacity reductions. At the same time, care should be taken to retain realistic capacity values, as the bottlenecks may also be caused by unrealistic O-D demand inputs or route choice models. In some situations, it may help to advance the simulation start time to allow for a more gradual evolution of congestion patterns. A simple diagnosis strategy is to conduct a DTA run with correct network and control settings but with a uniformly reduced O-D demand level to create relatively free-flow conditions, and observe if unusual congestion still develops at certain locations. Tracking such congestion patterns and locations usually leads to the discovery of coding errors. Figure 52 shows a sample error that was caught during this process. As it can be seen in this figure, some ramps of I-80 to I-580 interchanges do not have simulation volume. By checking the AMS_movement.csv file, it was discovered that the prohibited flag in this file was set 1 for these links (Figure 53). Another artifact of over-congested model results, and in particular if gridlock has set in, is that the assignment algorithm will often not converge to a stable solution. If the assignment is not stable, it is also of questionable value to be comparing the model outputs to empirical data, because the results may still have been varying significantly from iteration to iteration when the model run was stopped. Figure 41 shows that the Reno/Sparks model converges perfectly.

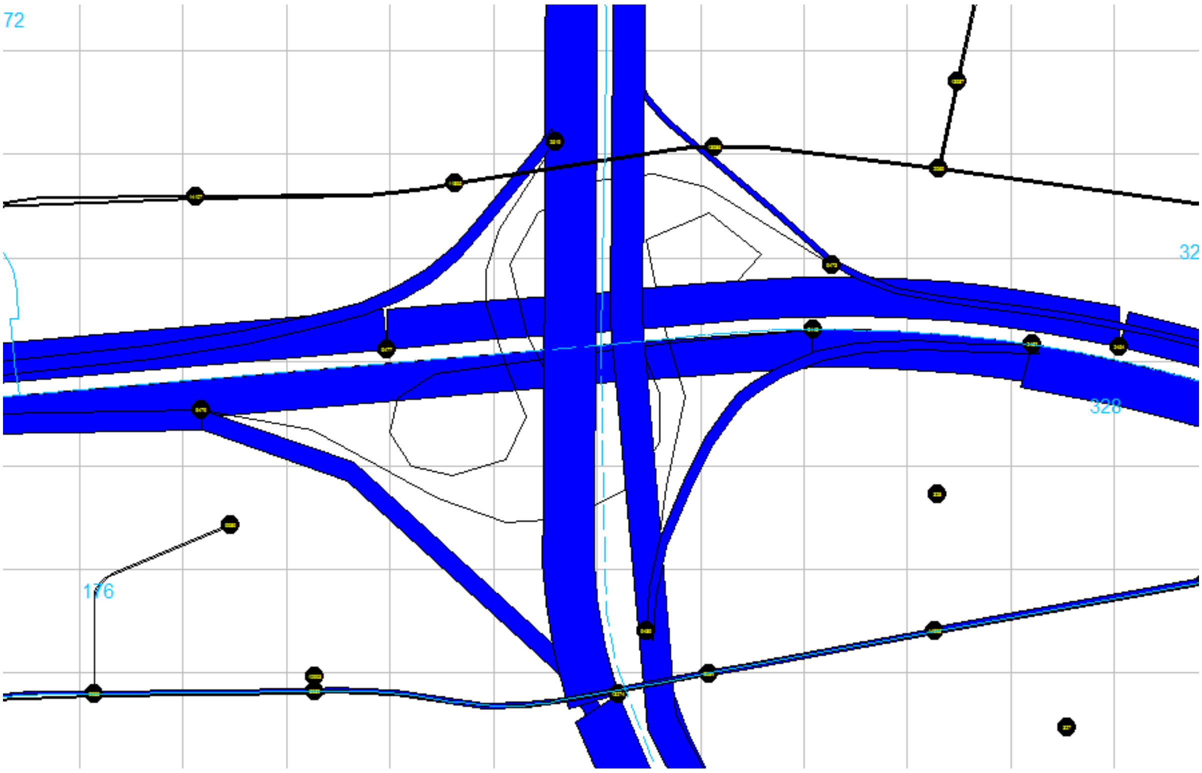
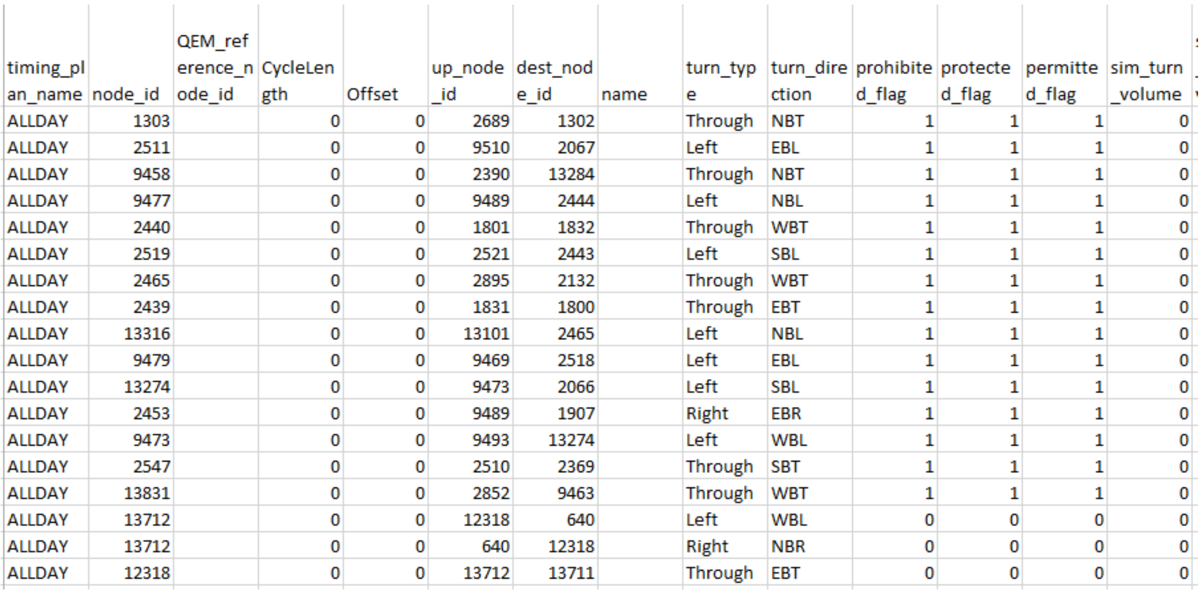


Figure 52: I-80 to I-580 interchange error



timing_plan_name	node_id	QEM_reference_node_id	CycleLength	Offset	up_node_id	dest_node_id	name	turn_type	turn_direction	prohibited_flag	protected_flag	permitted_flag	sim_turn_volume
ALLDAY	1303		0	0	2689	1302		Through	NBT	1	1	1	0
ALLDAY	2511		0	0	9510	2067		Left	EBL	1	1	1	0
ALLDAY	9458		0	0	2390	13284		Through	NBT	1	1	1	0
ALLDAY	9477		0	0	9489	2444		Left	NBL	1	1	1	0
ALLDAY	2440		0	0	1801	1832		Through	WBT	1	1	1	0
ALLDAY	2519		0	0	2521	2443		Left	SBL	1	1	1	0
ALLDAY	2465		0	0	2895	2132		Through	WBT	1	1	1	0
ALLDAY	2439		0	0	1831	1800		Through	EBT	1	1	1	0
ALLDAY	13316		0	0	13101	2465		Left	NBL	1	1	1	0
ALLDAY	9479		0	0	9469	2518		Left	EBL	1	1	1	0
ALLDAY	13274		0	0	9473	2066		Left	SBL	1	1	1	0
ALLDAY	2453		0	0	9489	1907		Right	EBR	1	1	1	0
ALLDAY	9473		0	0	9493	13274		Left	WBL	1	1	1	0
ALLDAY	2547		0	0	2510	2369		Through	SBT	1	1	1	0
ALLDAY	13831		0	0	2852	9463		Through	WBT	1	1	1	0
ALLDAY	13712		0	0	12318	640		Left	WBL	0	0	0	0
ALLDAY	13712		0	0	640	12318		Right	NBR	0	0	0	0
ALLDAY	12318		0	0	13712	13711		Through	EBT	0	0	0	0

Figure 53: Coding error in AMS_movement.csv file

Table 15 presents the changes of some Measures of Effectiveness (MOEs) according to different time in minutes. For illustration purposes, the changes in speed, volume and queues are presented in Figures 54 to 65. Figures 54 to 57 illustrate the changes of average speed, Figures 58 to 61 show the difference between observed (pink hatched) and simulation (blue) volume, and Figures 62 to 65 show the queue (red links) sections at a.m. peak hours.

Table 15: A sample of network MOEs across the simulation time

Time in min	Cumulative in Flow Count	Cumulative Out Flow Count	Flow per Minute	Average Trip Time in Minute
7:06	198087	12114	23500	10.6393
7:07	199308	12090	23940	10.4276
7:08	200536	12007	22900	10.1679
7:09	201740	12049	24920	10.0951
7:10	202958	12030	23980	10.5174
7:11	204111	11976	21980	9.94238
7:12	205348	11995	25120	10.9174
7:13	206519	11978	23080	10.2278
7:14	207693	11986	23640	9.98826
7:15	208851	12149	26420	10.9645

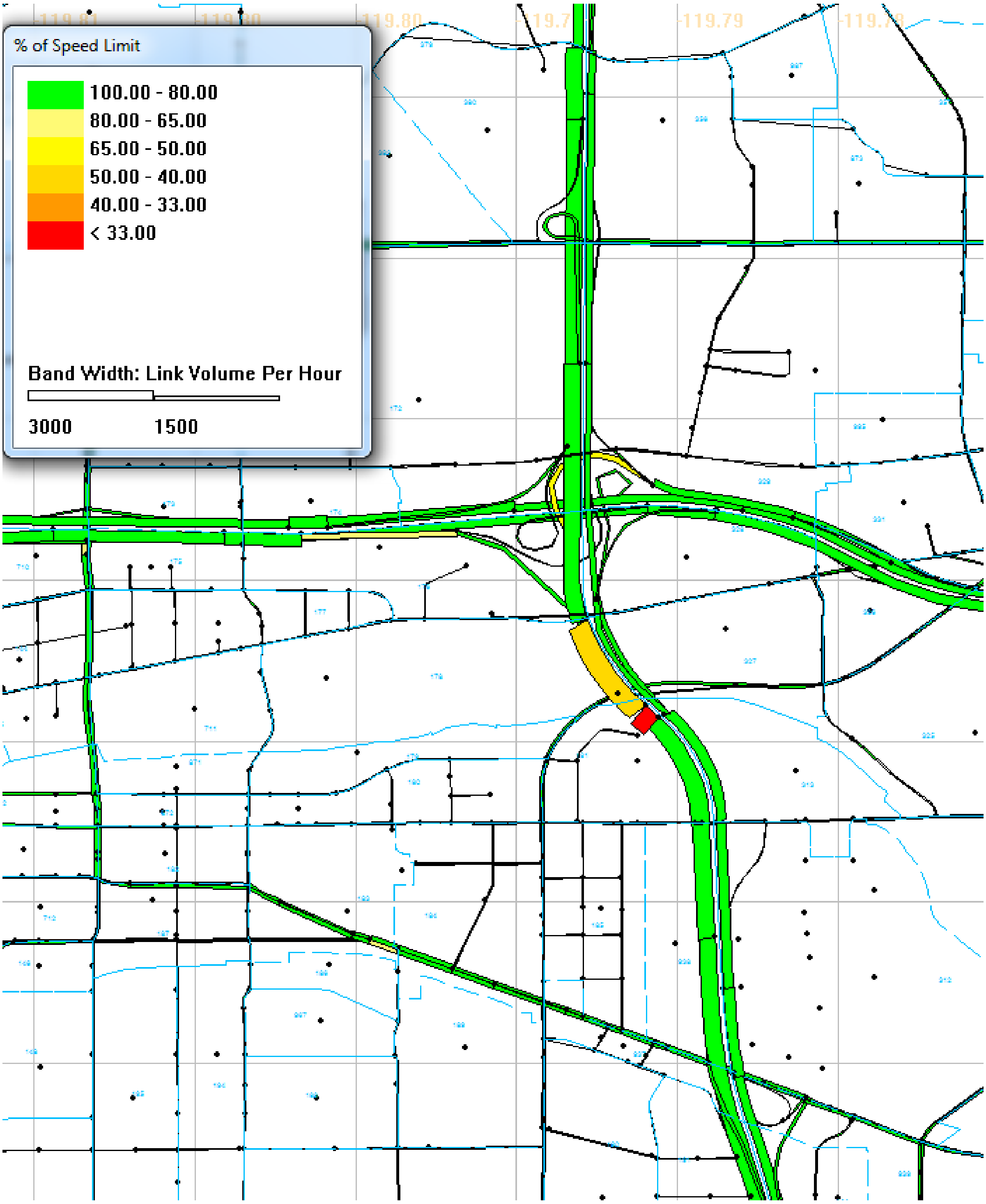


Figure 54: Changes in average speed on different road sections at 6:00 a.m.

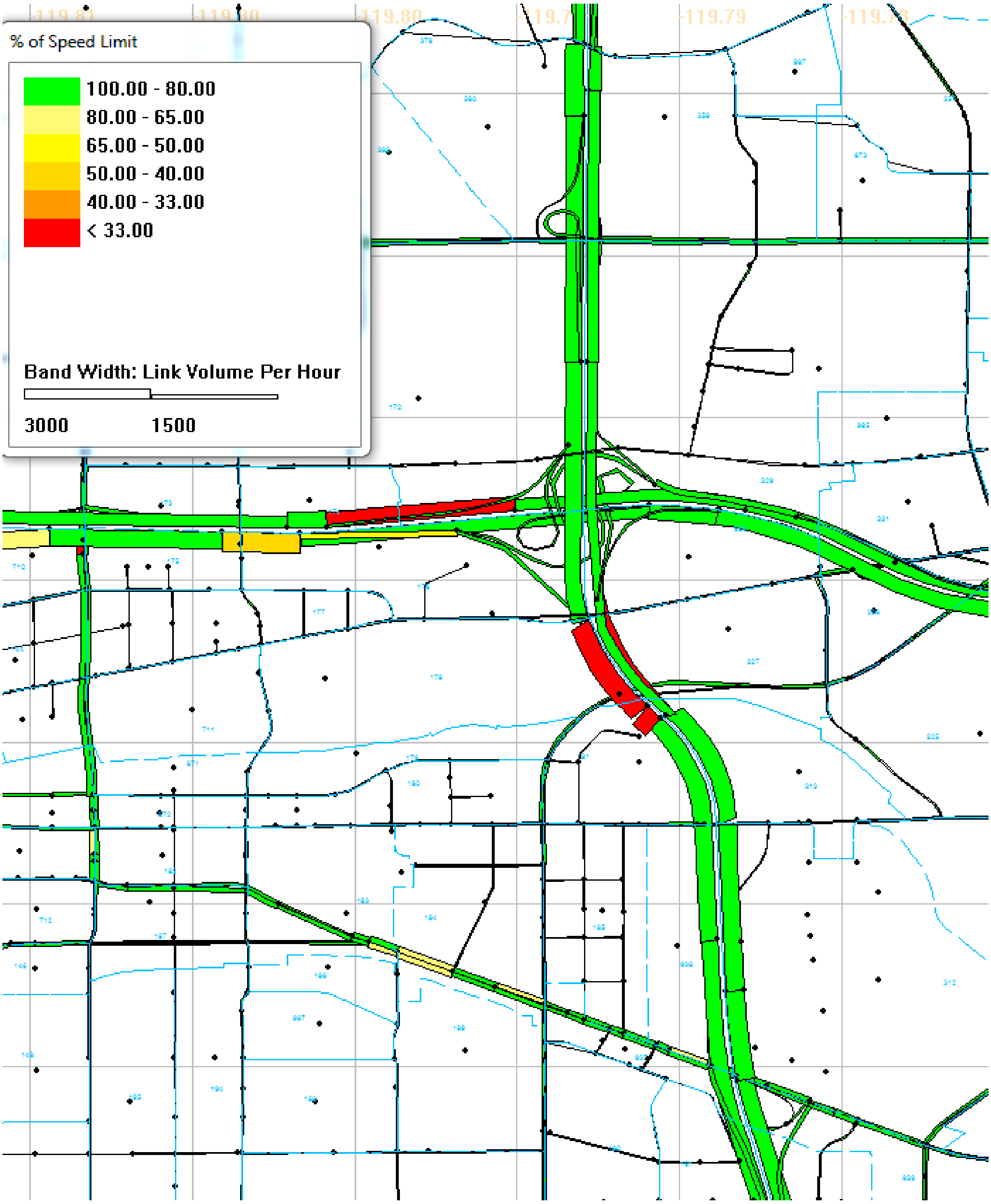


Figure 55: Changes in average speed on different road sections at 7:00 a.m.

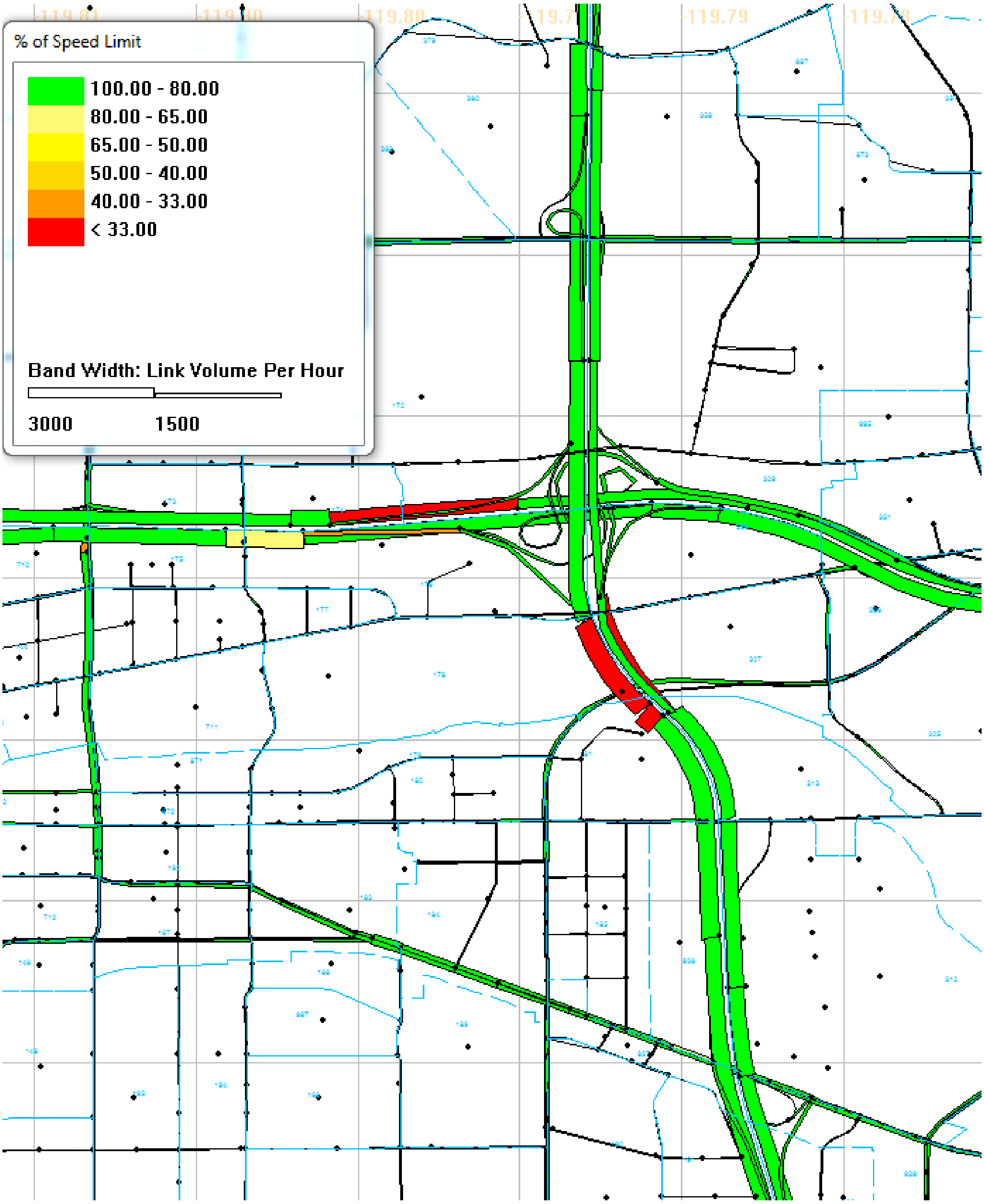


Figure 56: Changes in average speed on different road sections at 8:00 a.m.

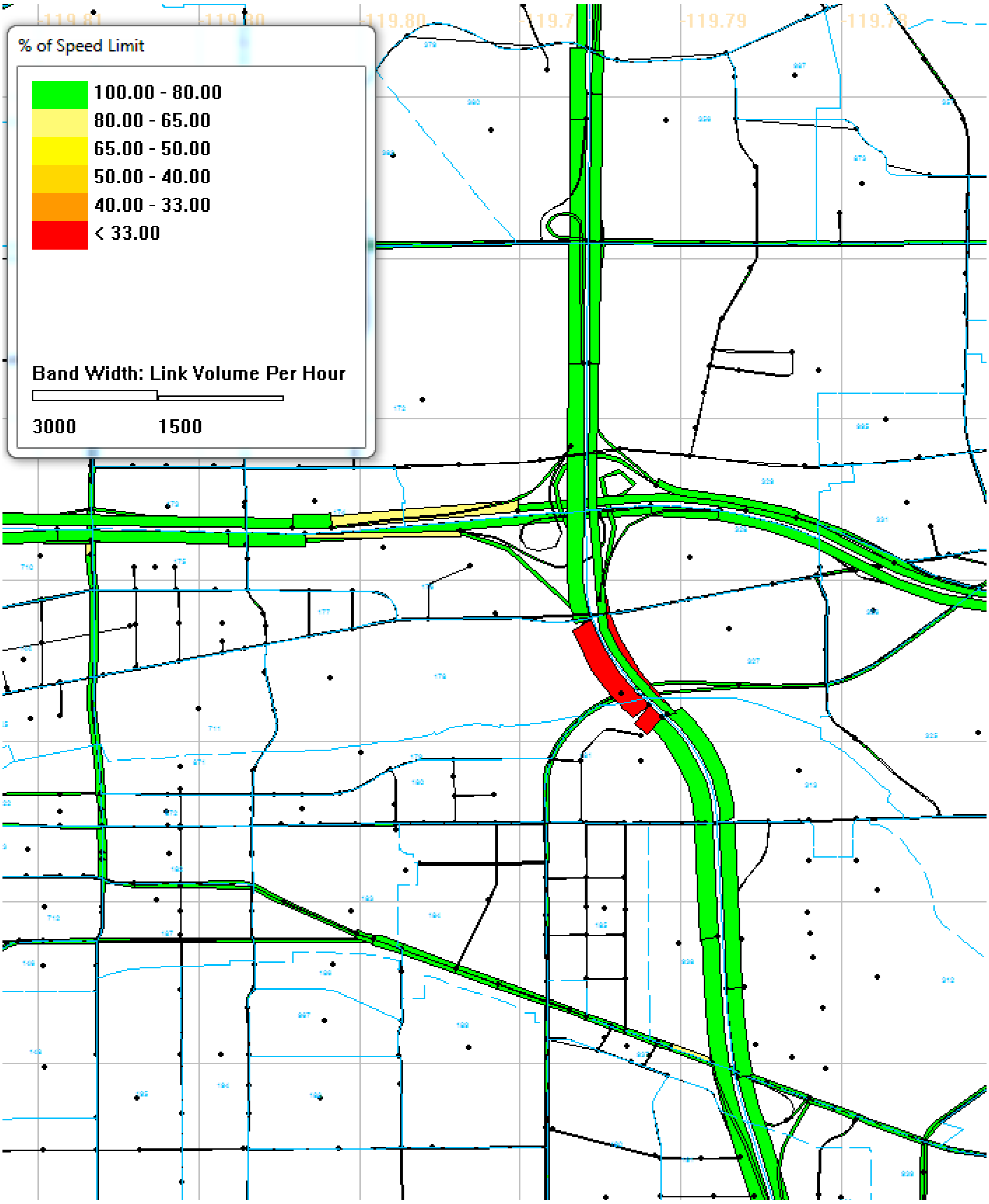


Figure 57: Changes in average speed on different road sections at 9:00 a.m.

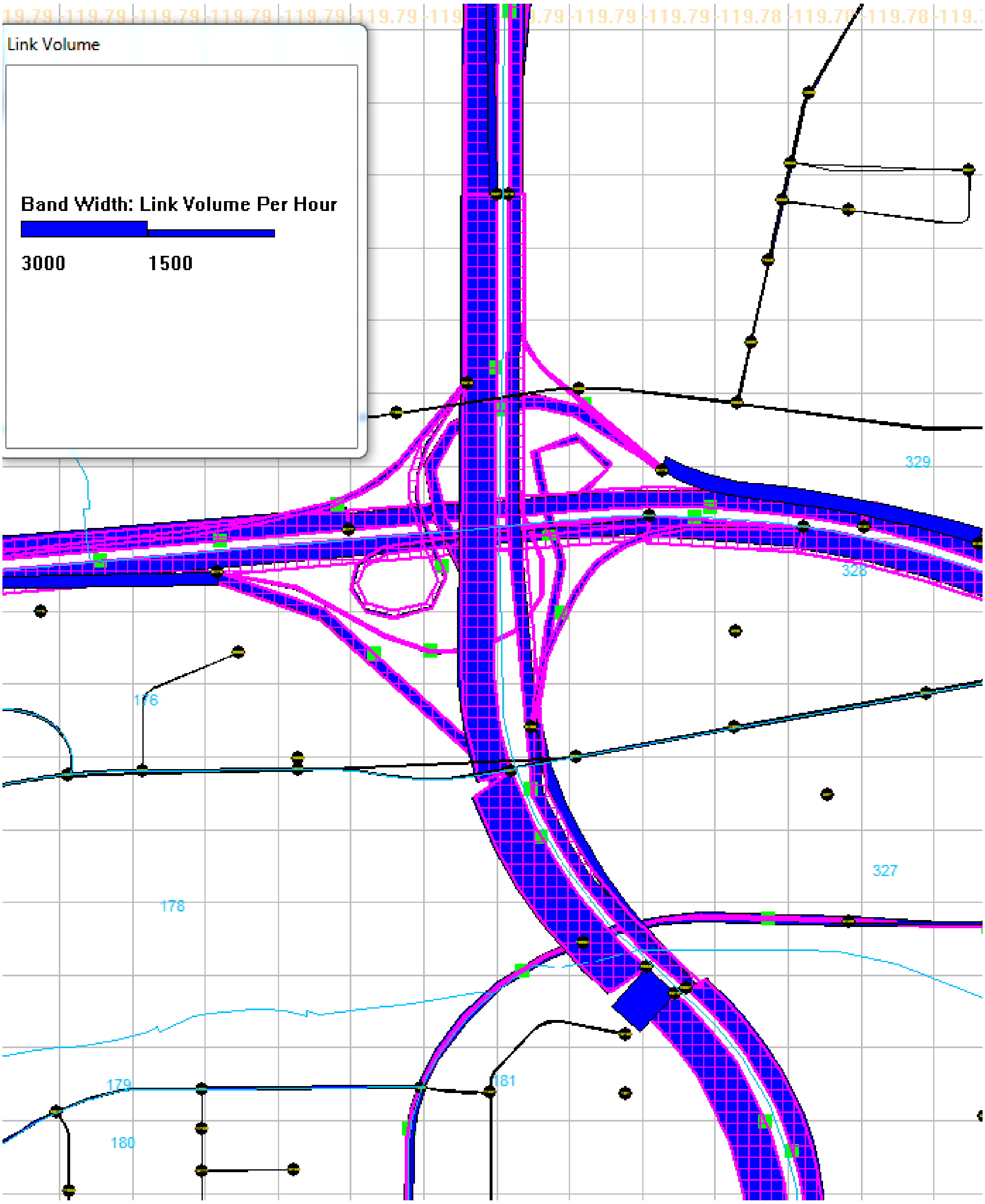


Figure 58: Differences between observed (pink hatched) and simulation (blue) volume on different road sections at 6:00 a.m.

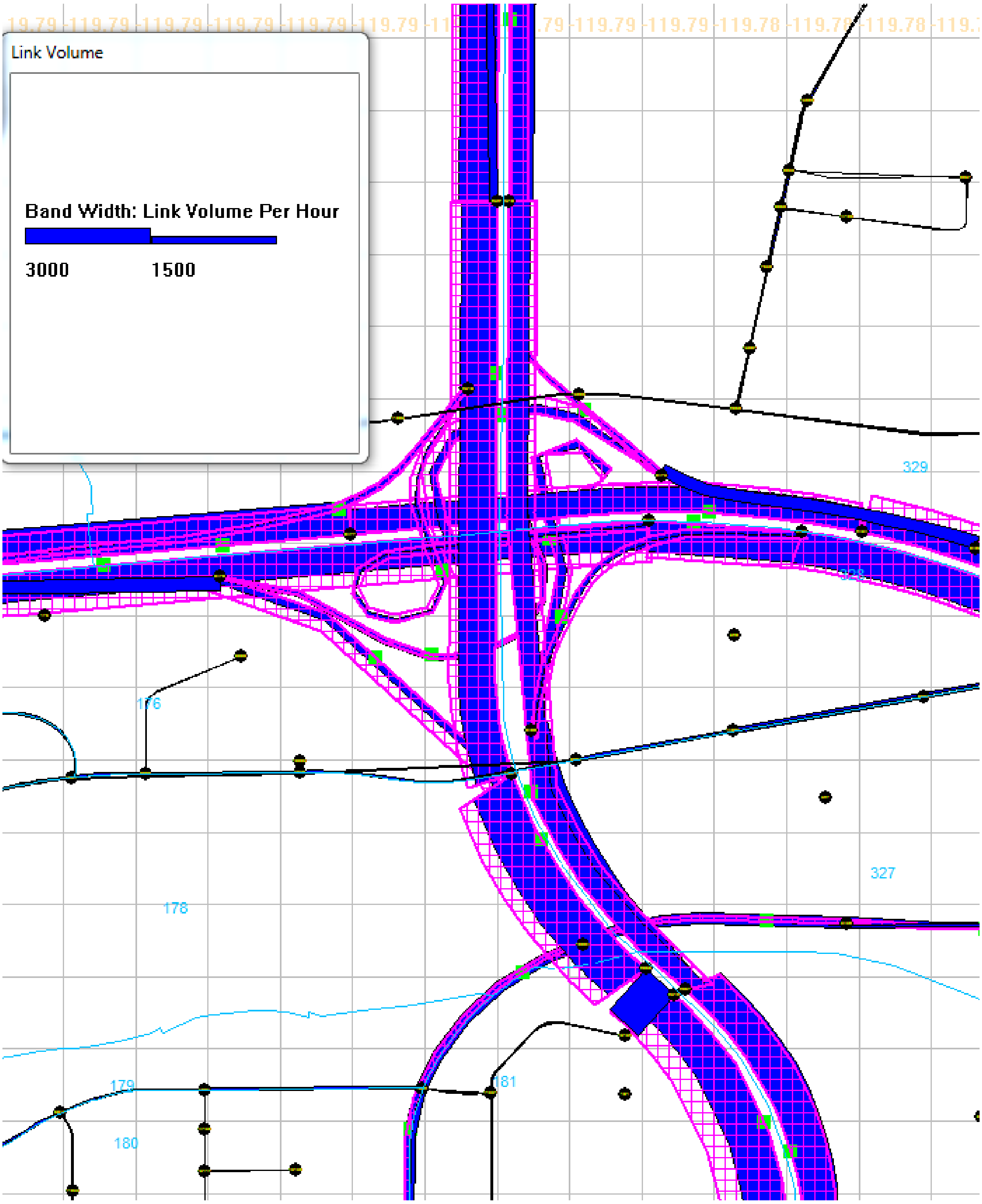


Figure 59: Differences between observed (pink hatched) and simulation (blue) volume on different road sections at 7:00 a.m.

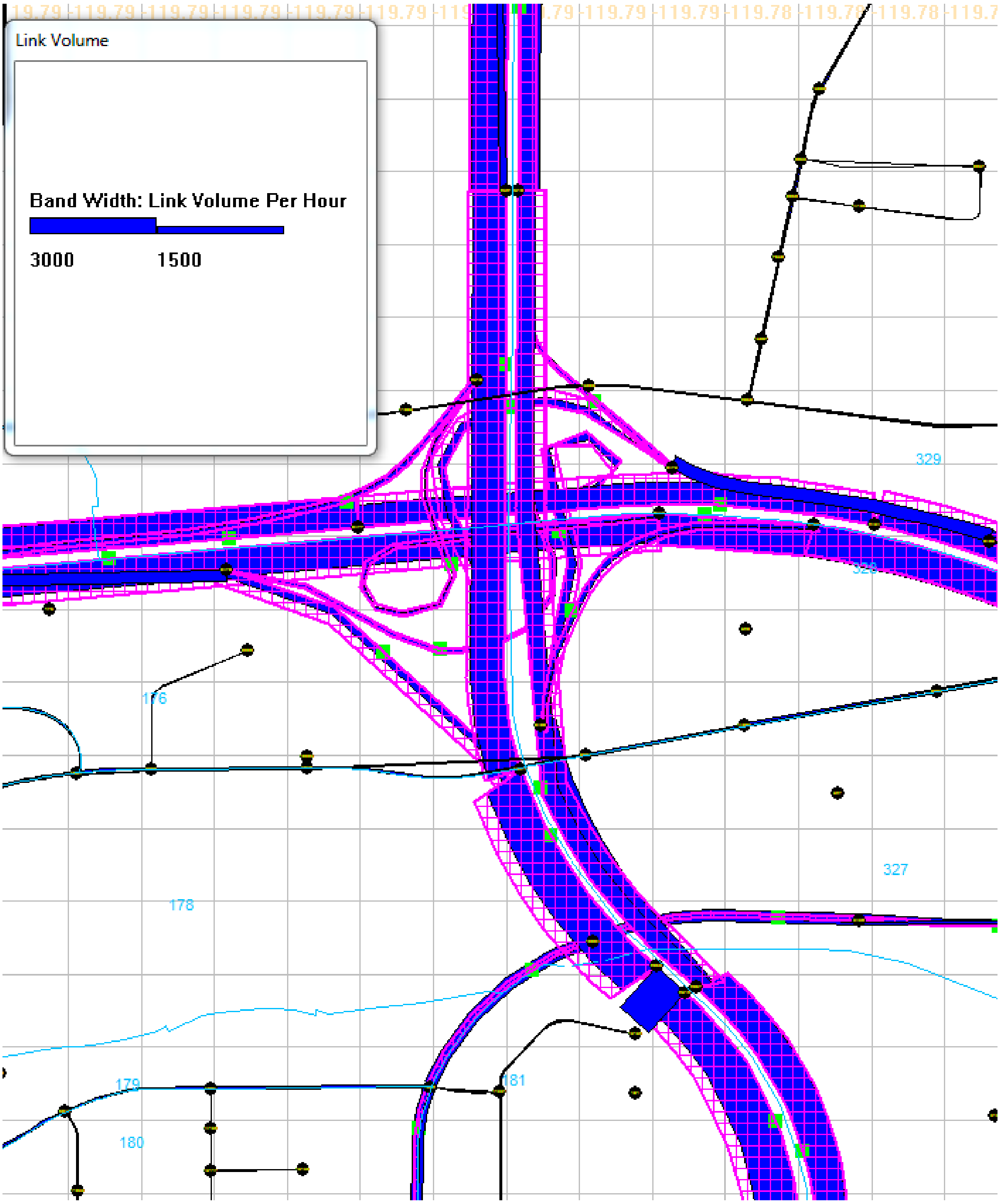


Figure 60: Differences between observed (pink hatched) and simulation (blue) volume on different road sections at 8:00 a.m.

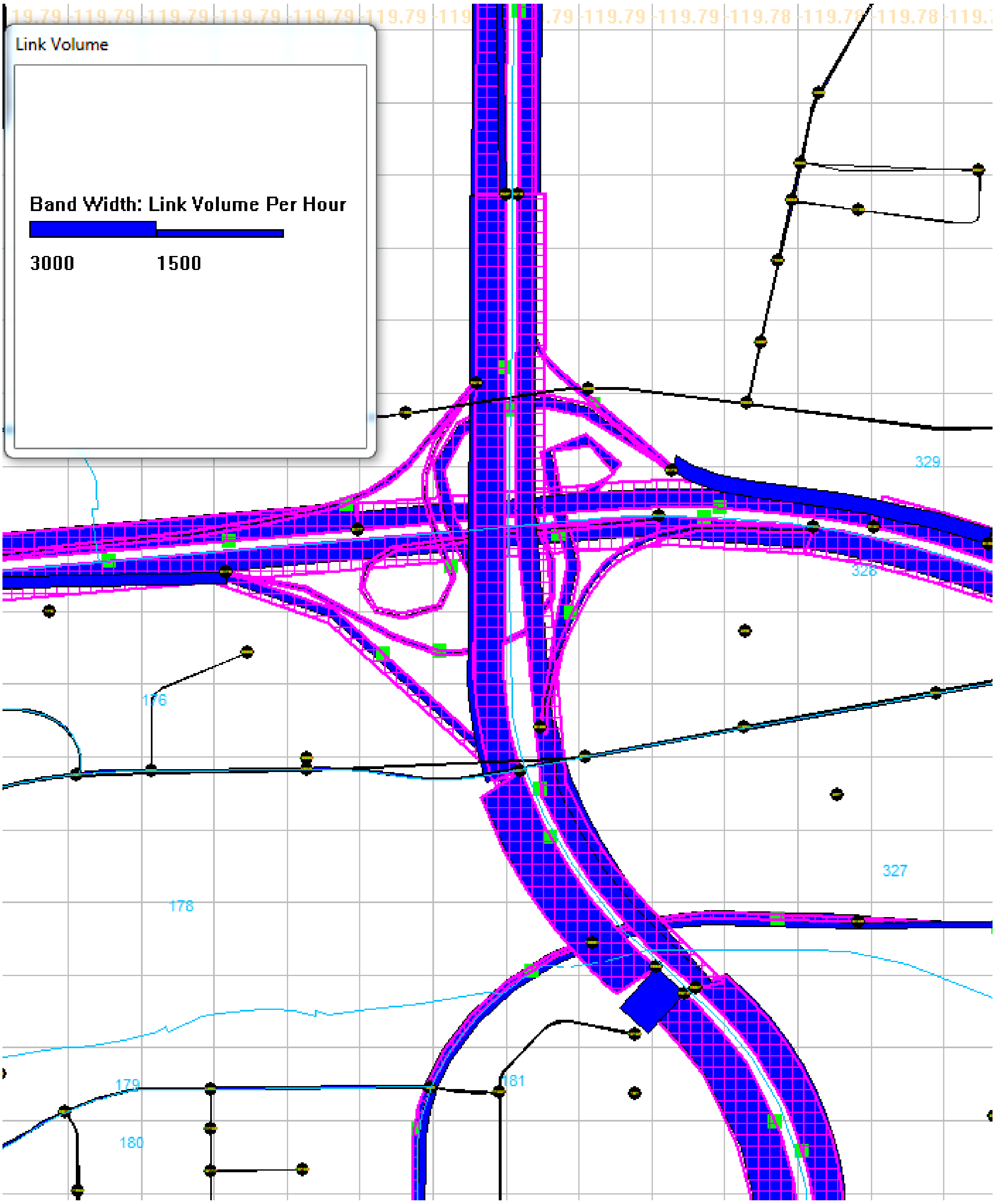


Figure 61: Differences between observed (pink hatched) and simulation (blue) volume on different road sections at 9:00 a.m.

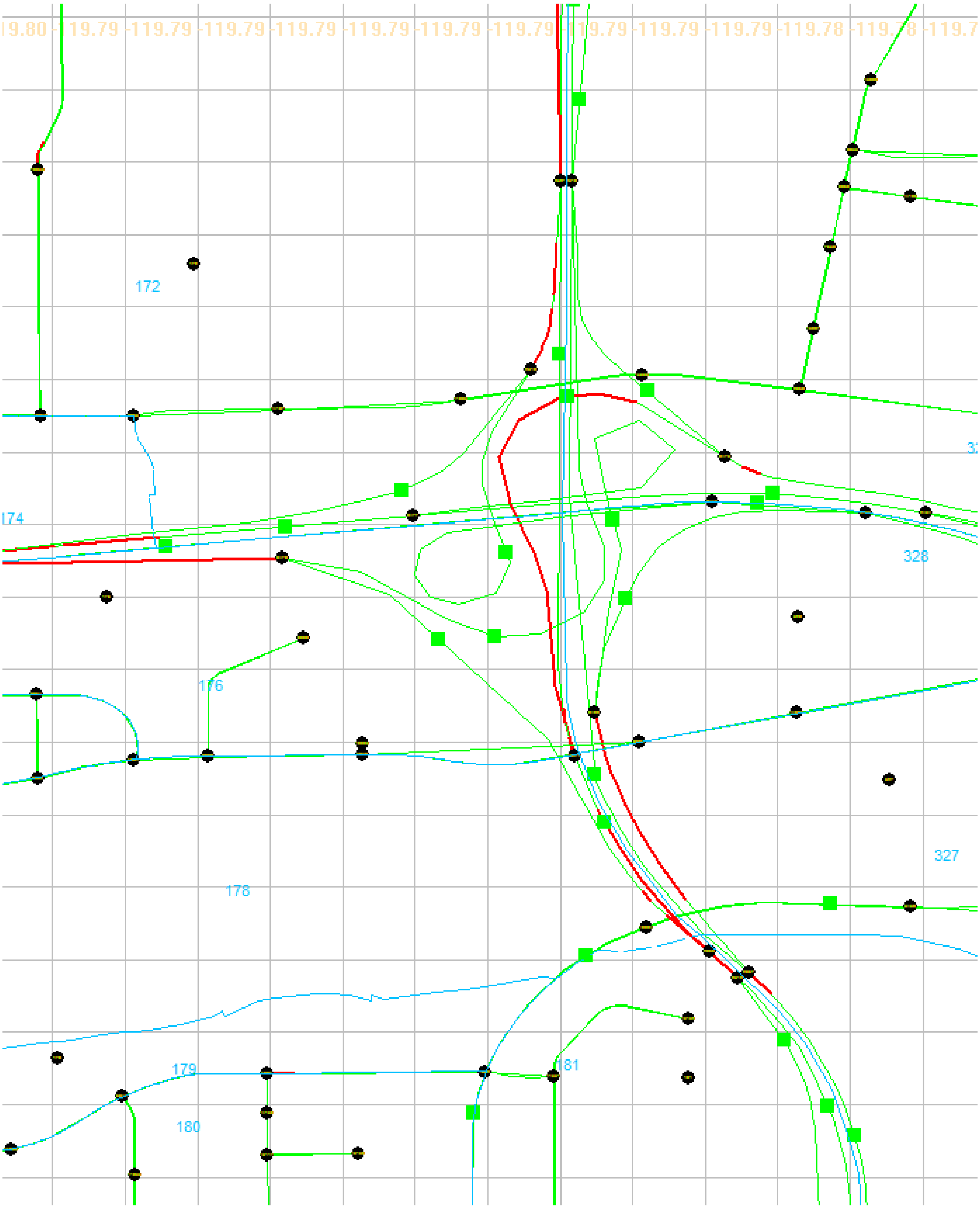


Figure 62: Illustration of queues (red color denotes a long queue) on different road sections at 6:00 a.m.

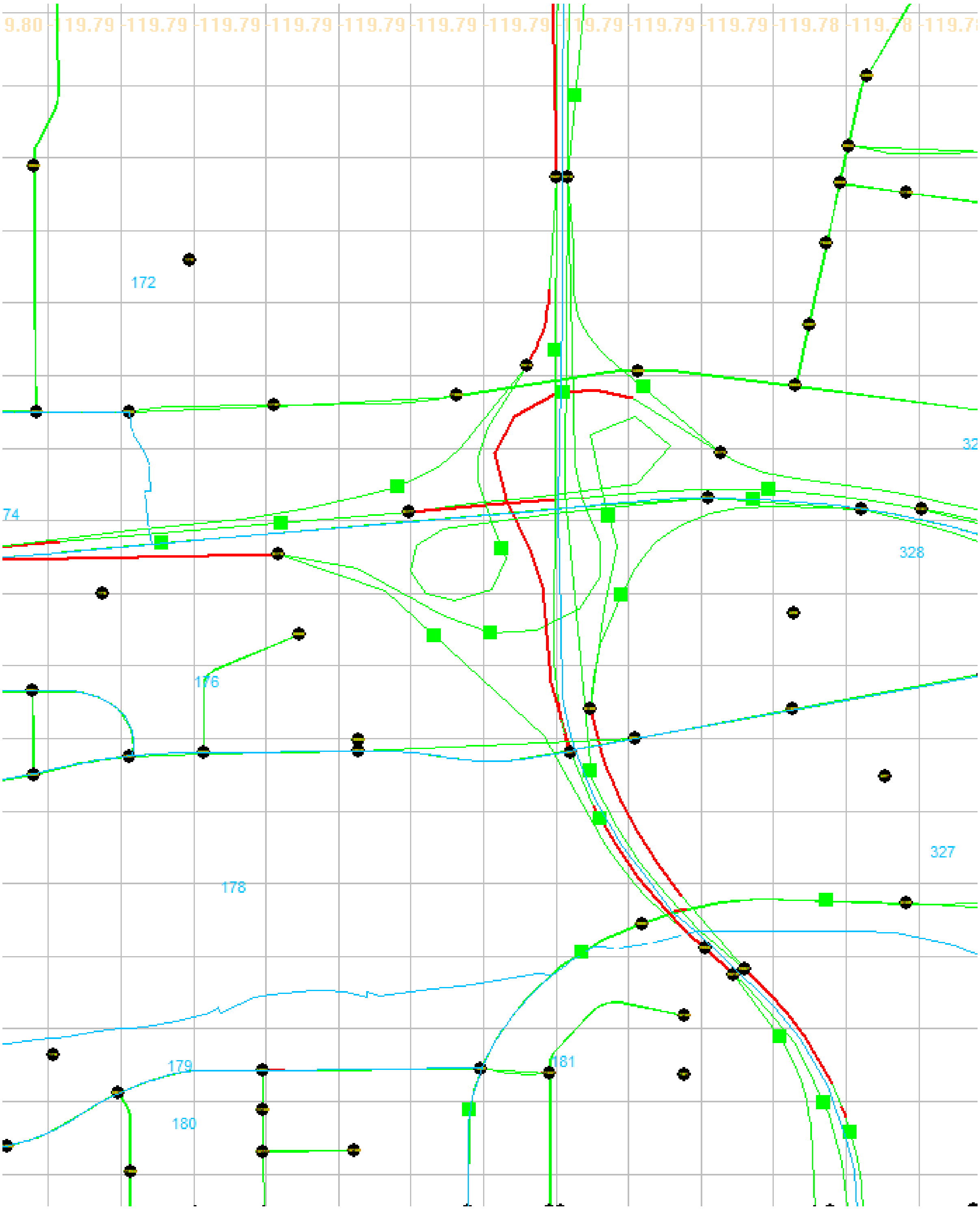


Figure 63: Illustration of queues (red color denotes a long queue) on different road sections at 7:00 a.m.

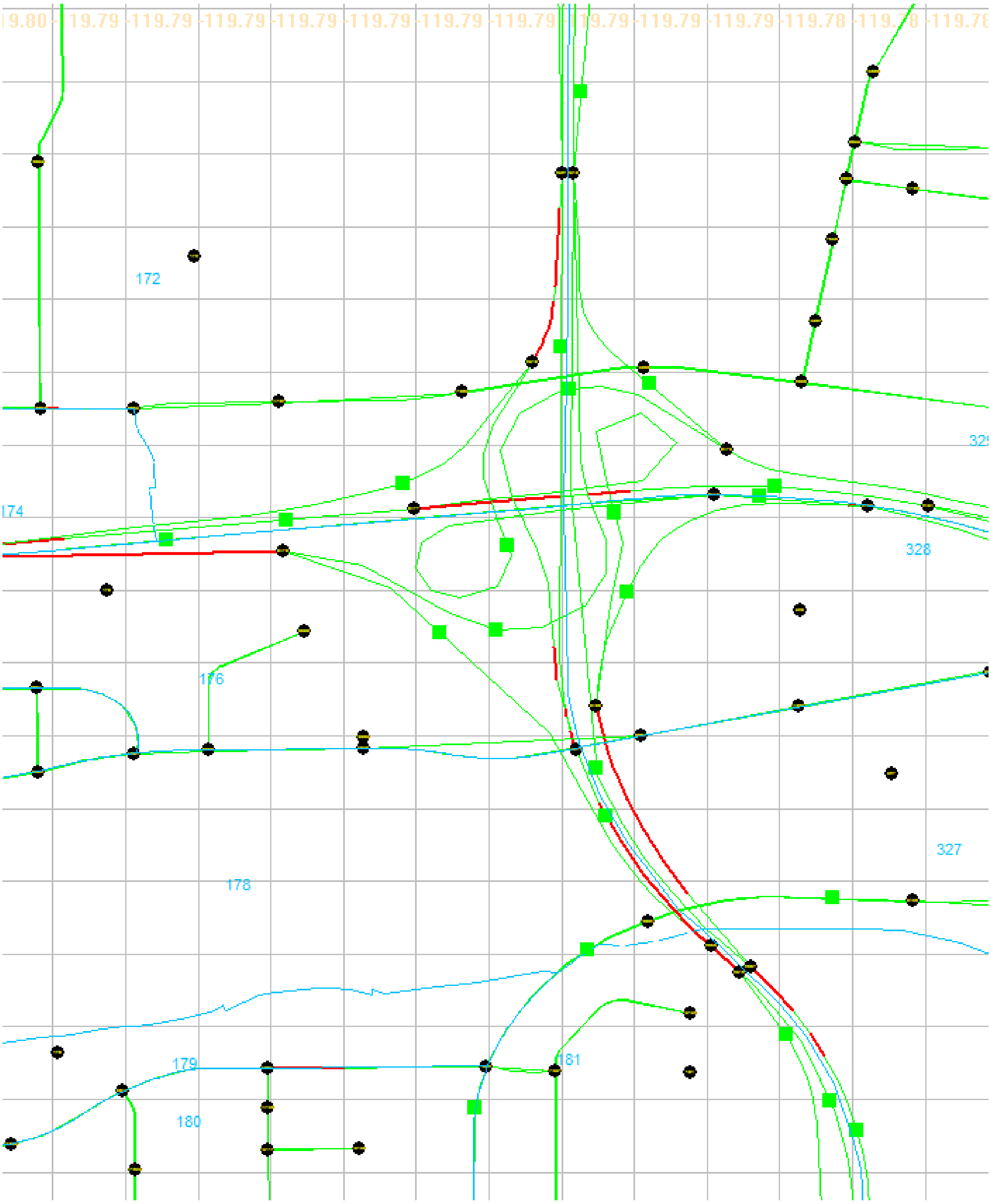


Figure 64: Illustration of queues (red color denotes a long queue) on different road sections at 8:00 a.m.

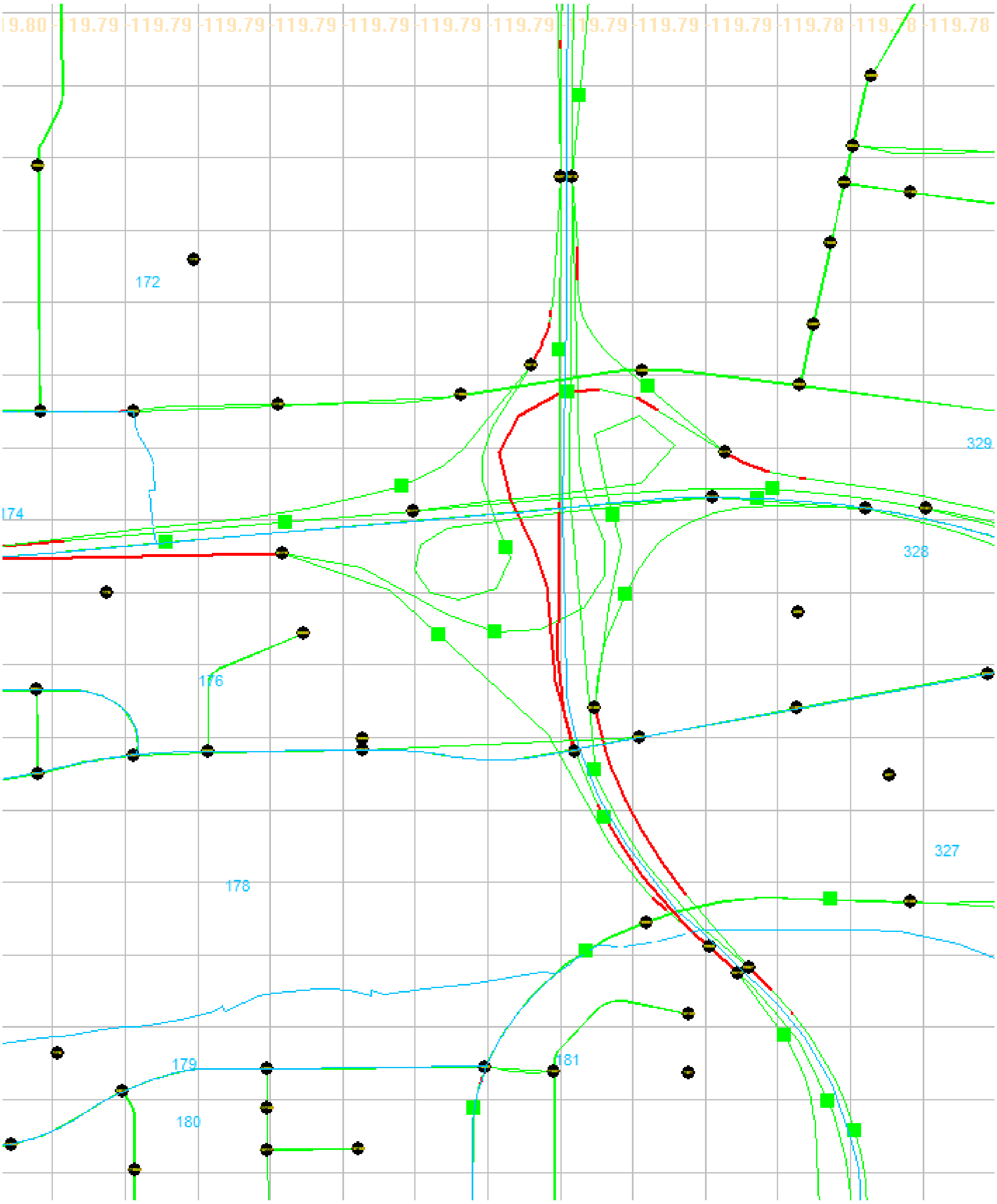


Figure 65: Illustration of queues (red color denotes a long queue) on different road sections at 9:00 a.m.

The purpose of the qualitative analysis stage is primarily to achieve model results that exhibit a stable solution, are free of gridlock and, if possible, in which the overall congestion pattern at least resembles the actual conditions on the street. The qualitative analysis illustrated in these figures does not show any obvious error in the model, therefore the quantitative analysis described in the next section can be started.

5.2. Quantitative Analysis

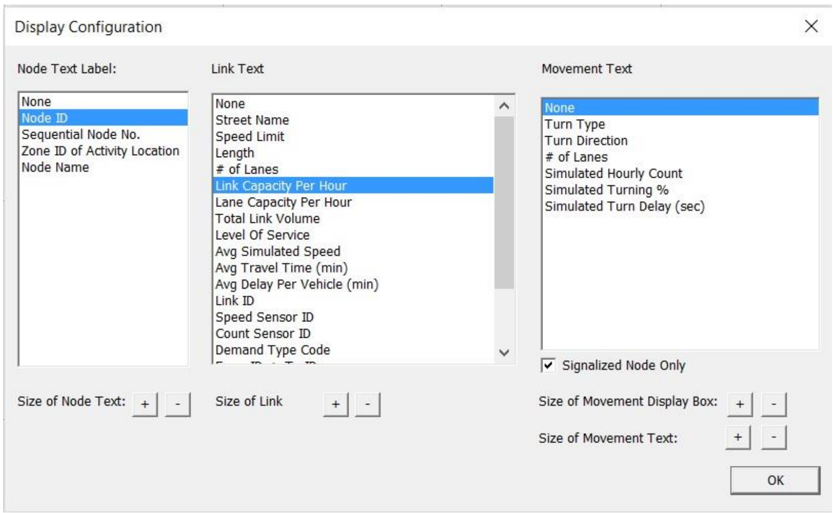
This stage of the calibration process is based on direct comparisons between model results and empirical observations. Various statistical measures may be used to quantify the goodness of fit between the DTA output and the observed data, but the actual process of improving the fit by adjusting input data is essentially the same as used in the qualitative analysis stage. It is based on an understanding of traffic phenomena and causes of congestion along with common sense and modeling judgment [3]. The following section provides a general process of deductive analysis used for improving the Reno/Sparks model. In its simplest form, the quantitative analysis work consists of investigating one outlier at a time in order to determine if it is not the result of an error in the input data. In this context, an error might simply reflect a need for a minor adjustment of some input value. The most important data to consider for calibration, especially in the initial stages of this work, are traffic counts.

Once the inputs have been corrected or adjusted to the point where the outputs are deemed acceptable, the calibration process is complete. There is an expression about modeling that is good to keep in mind when calibrating: “All models are wrong (i.e., imperfect), but some are useful” [33]. The challenge is to make the model good enough to be useful, namely, to make useful predictions [3].

In addition to time dependent link MOE visualizations that were introduced in the previous section (such as speed, queue and volume), NeXTA provides other means of visualization interfaces for checking the traffic states such as density and volume, which are described briefly as follows.

1) Network Level Text Display

By following the steps in Figure 66, traffic states for all links in the network can be viewed. For example, check the “Link Capacity Per Hour”.



STEPS

Step 1: Click on the "Config" button at the top of the GIS layers panel

Step 2: "Link Capacity Per Hour" in "link text"

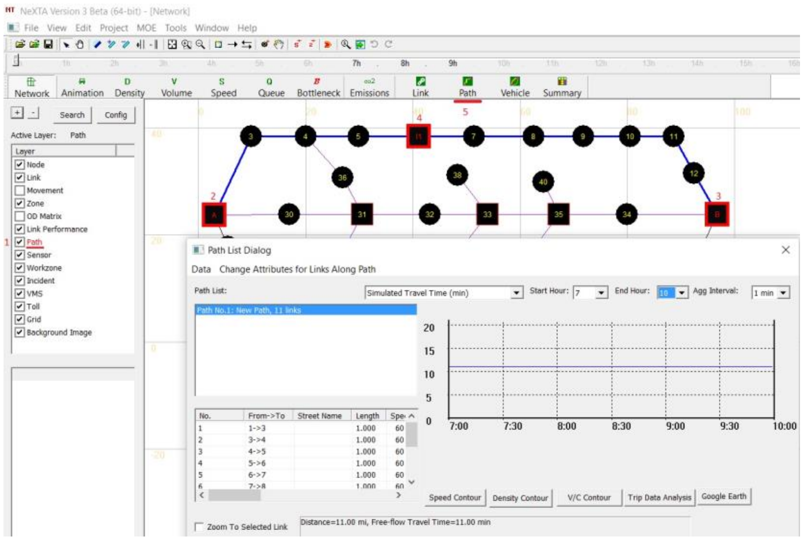
Step 3: Slide the "time bar" to the simulation time at 7am to 9am

Step 4: View the link capacity of any links in the network.

Figure 66: Procedure of analyzing the bottleneck

2) Path Level Dynamic Contour Plot

The method of selecting a path is shown in Figure 67. By following the steps in this figure, define a path, and then plot "Density Contour", "Speed Contour", "V/C Contour", as shown in Figure 68.



No.	From->To	Street Name	Length	Spe
1	1->3		1.000	60
2	3->4		1.000	60
3	4->5		1.000	60
4	5->6		1.000	60
5	6->7		1.000	60
6	7->8		1.000	60

Distance=11.00 mi, Free-flow Travel Time=11.00 min

STEPS

Step 1: Select the "path" layer in the GIS layers panel

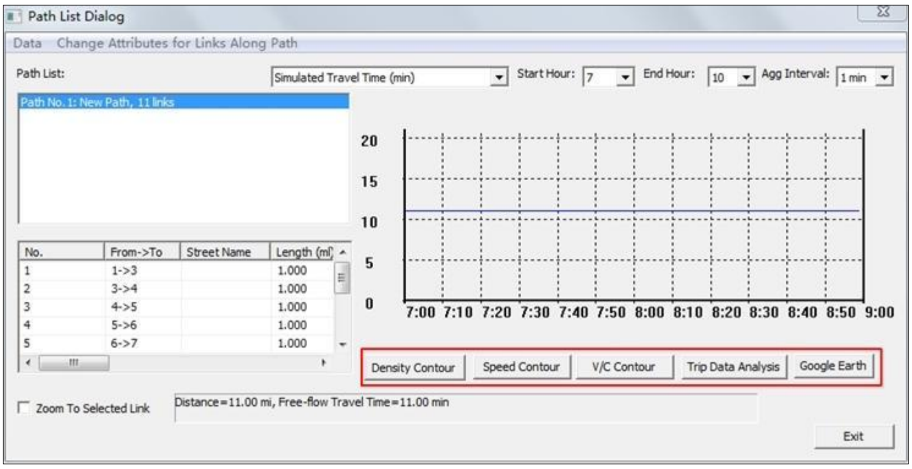
Step 2: Click on node 1-> right click menu to choose "Direction from here"

Step 3: Click node 2-> right click menu to choose "Direction to here"

Step 4: Select any node (e.g. node 6) of the path 1-> right click to choose "Add Intermediate Destination here".

Step 5: To view the traffic states of selected path, click on the "Path" button.

Figure 67: Defining a path



Path List: Simulated Travel Time (min) Start Hour: 7 End Hour: 10 Agg Interval: 1 min

Path No. 1: New Path, 11 links

No.	From->To	Street Name	Length (mi)
1	1->3		1.000
2	3->4		1.000
3	4->5		1.000
4	5->6		1.000
5	6->7		1.000

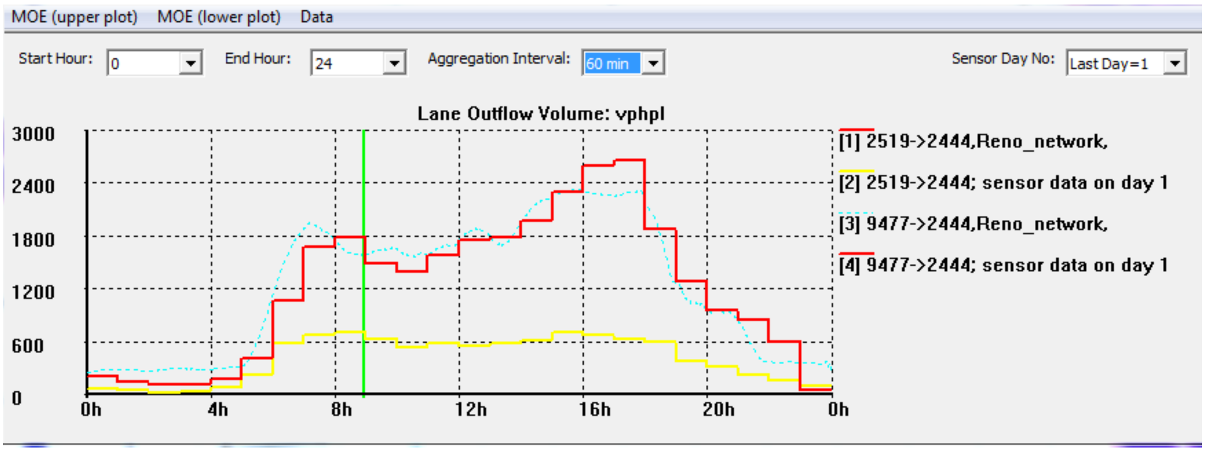
☐ Zoom To Selected Link

Distance=11.00 mi, Free-flow Travel Time=11.00 min

Figure 68: Path list dialog box

3) Link Level MOE Display

After simulation, select the “Link performance” layer of the GIS layer panel and then click one link to see the link MOE dialog shown in Figure 69.


MOE (upper plot)
MOE (lower plot)
Data
Lane Outflow Volume: vphpl
3000
2400
1800
1200
600
0
0h
4h
8h
12h
16h
20h
0h
[1] 2519->2444,Reno_network,
[2] 2519->2444; sensor data on day 1
[3] 9477->2444,Reno_network,
[4] 9477->2444; sensor data on day 1

Start Hour: 0 End Hour: 24 Aggregation Interval: 60 min Sensor Day No: Last Day=1

Figure 69: Interface of the MOE of links 2519->2444 and 9477->2444

By clicking the “MOE (upper plot)” in the menu bar, different traffic state variables shown in Figure 70 can be examined. One can use the key combination “control +click” to select multiple links into the link MOE display.

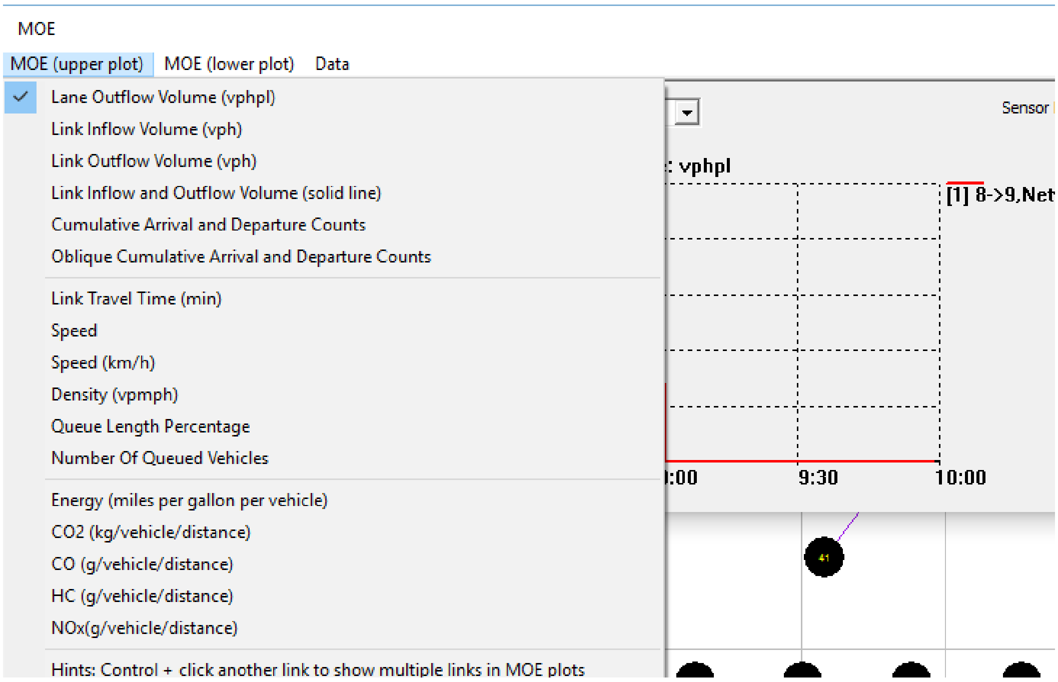
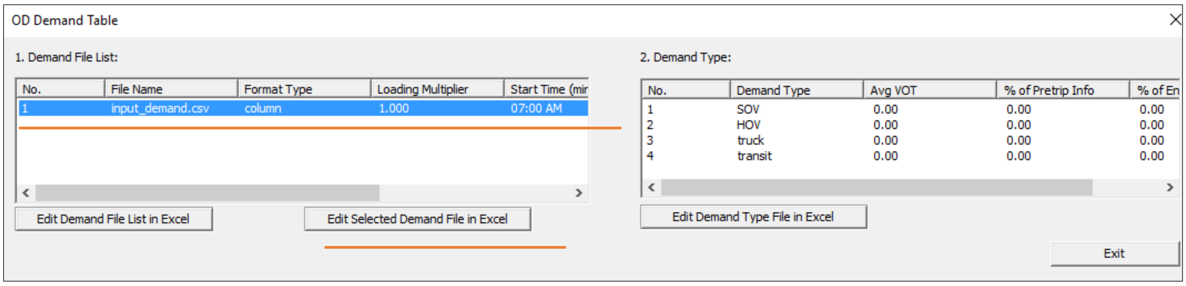

MOE
MOE (upper plot)
MOE (lower plot)
Data
✓ Lane Outflow Volume (vphpl)
Link Inflow Volume (vph)
Link Outflow Volume (vph)
Link Inflow and Outflow Volume (solid line)
Cumulative Arrival and Departure Counts
Oblique Cumulative Arrival and Departure Counts
Link Travel Time (min)
Speed
Speed (km/h)
Density (vpmph)
Queue Length Percentage
Number Of Queued Vehicles
Energy (miles per gallon per vehicle)
CO2 (kg/vehicle/distance)
CO (g/vehicle/distance)
HC (g/vehicle/distance)
NOx(g/vehicle/distance)
Hints: Control + click another link to show multiple links in MOE plots
Sensor
: vphpl
[1] 8->9,Net
:00
9:30
10:00
41

Figure 70: Detailed content of Link MOE

4) OD demand matrix

For checking the OD demand, there are two methods:

- Method 1: check the input_demand_file_list.csv and input_demand.csv in table 1;
- Method 2: through the NeXTA's interface as shown in Figure 71.


OD Demand Table
1. Demand File List:
2. Demand Type:
Edit Demand File List in Excel
Edit Selected Demand File in Excel
Edit Demand Type File in Excel
Exit

No.	File Name	Format Type	Loading Multiplier	Start Time (mir
1	input_demand.csv	column	1.000	07:00 AM

No.	Demand Type	Avg VOT	% of Pretrip Info	% of En
1	SOV	0.00	0.00	0.00
2	HOV	0.00	0.00	0.00
3	truck	0.00	0.00	0.00
4	transit	0.00	0.00	0.00

STEPS
Step 1: Click on "Project" in the menu bar
Step 2: "2. Demand Database"
Step 3: Click on the row of "1. Demand File List".
Step 4: Click on the "Edit Selected Demand File in Excel". The corresponding OD demand file ("input_demand.csv") will be open in Excel.

Figure 71: Checking the OD demand

5.3. Calibration Procedure

After running ODME and correcting the model inputs and coding errors, still some differences may remain between sensor data and simulation volumes. This section presents a general methodology for investigating discrepancies between model outputs and field data. Figures 72 to 75 show discrepancies at some important links in the Reno/Sparks network.

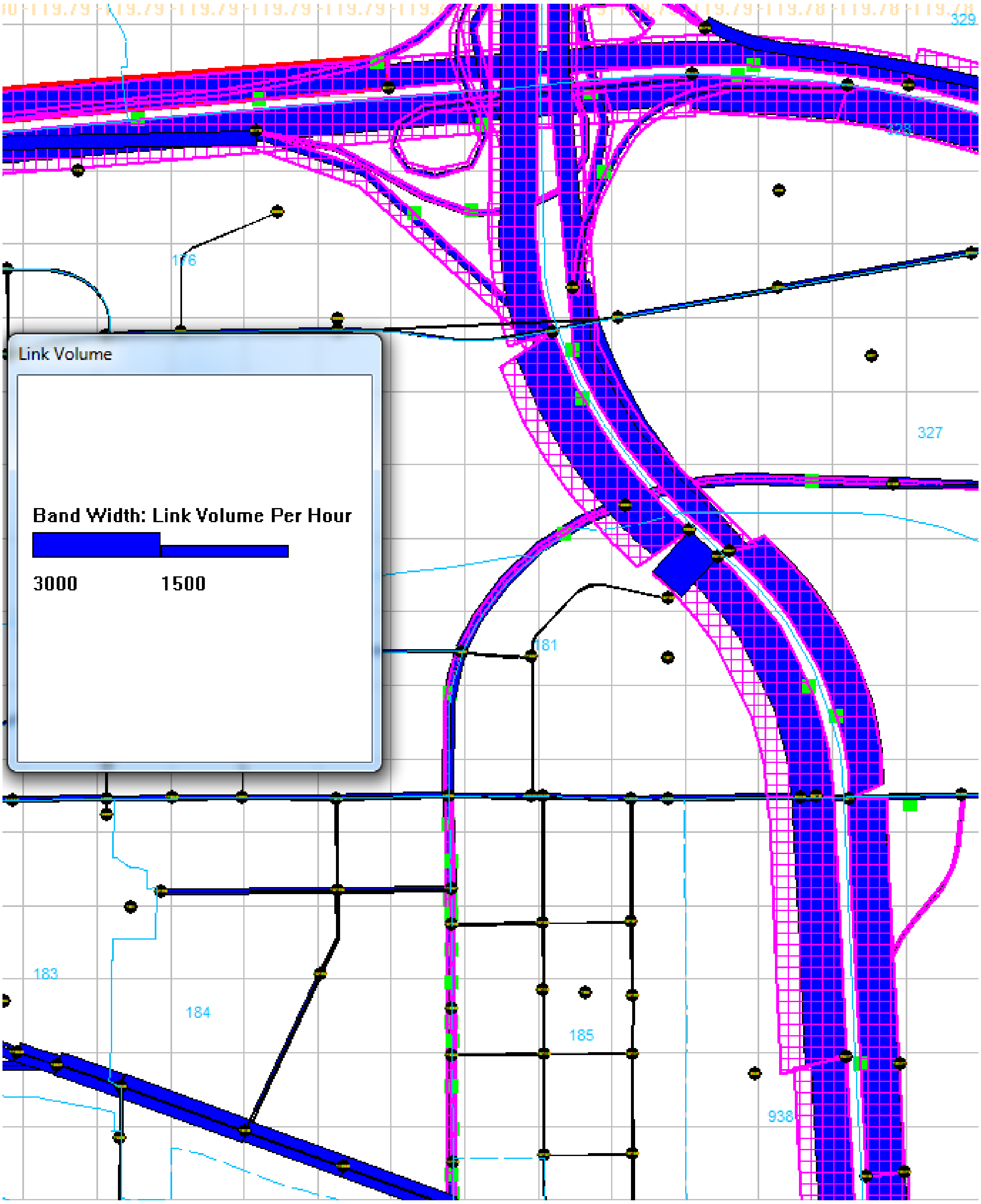


Figure 72: I580 southbound at 7 a.m., sensor counts are bigger than simulation volume

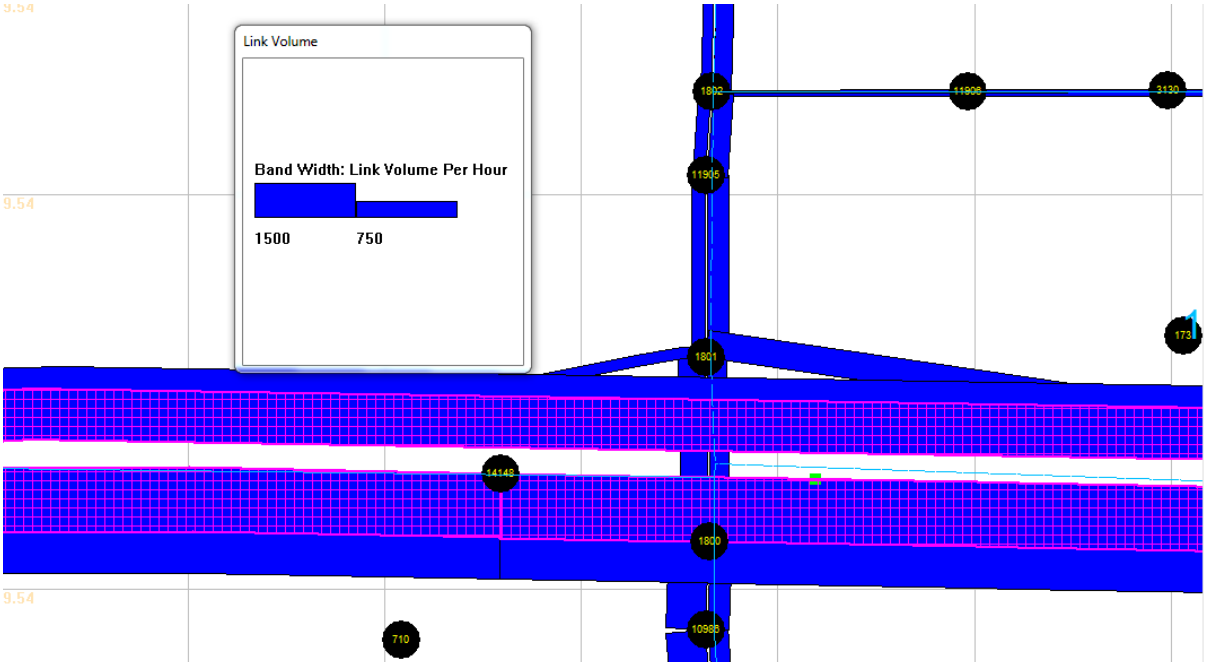


Figure 73: I80 east bound and westbound at 6:50 a.m., simulation volumes are bigger than sensor counts

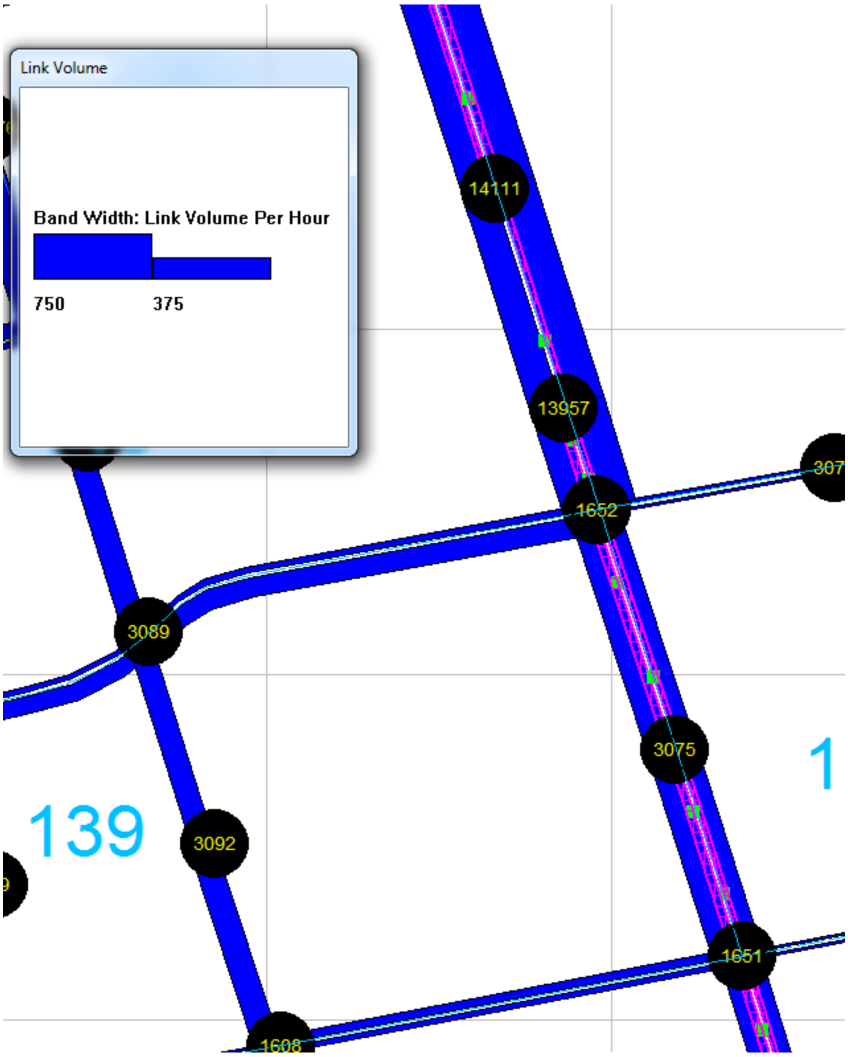


Figure 74: Virginia St., at 7:00 a.m., simulation volumes are bigger than sensor counts

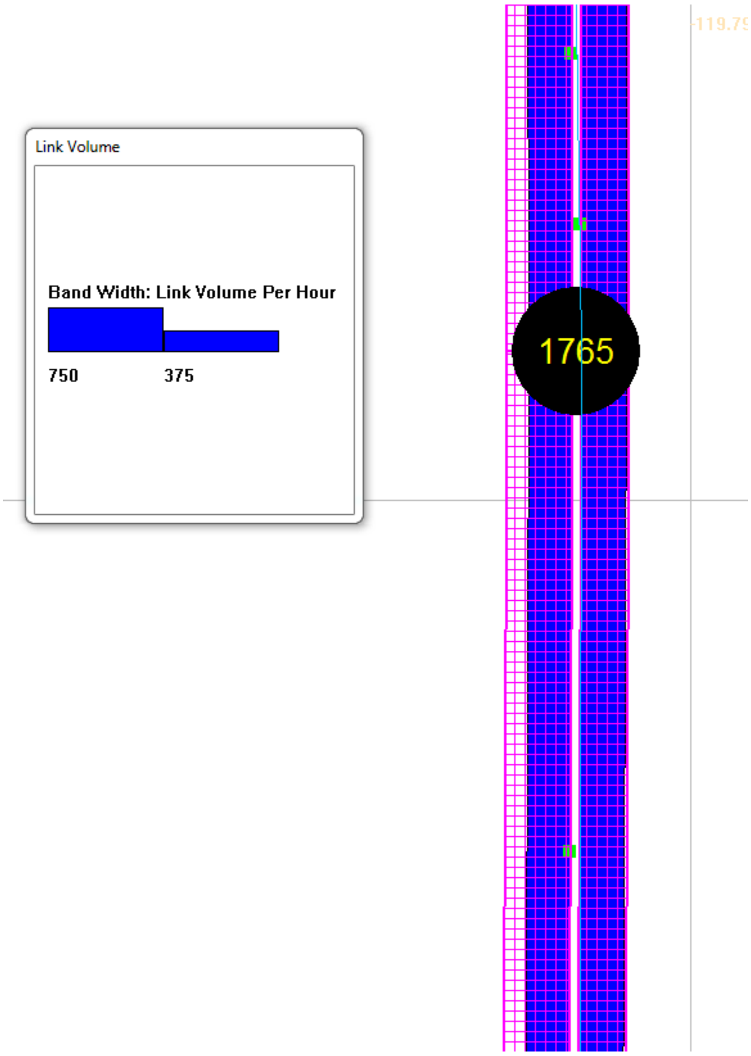
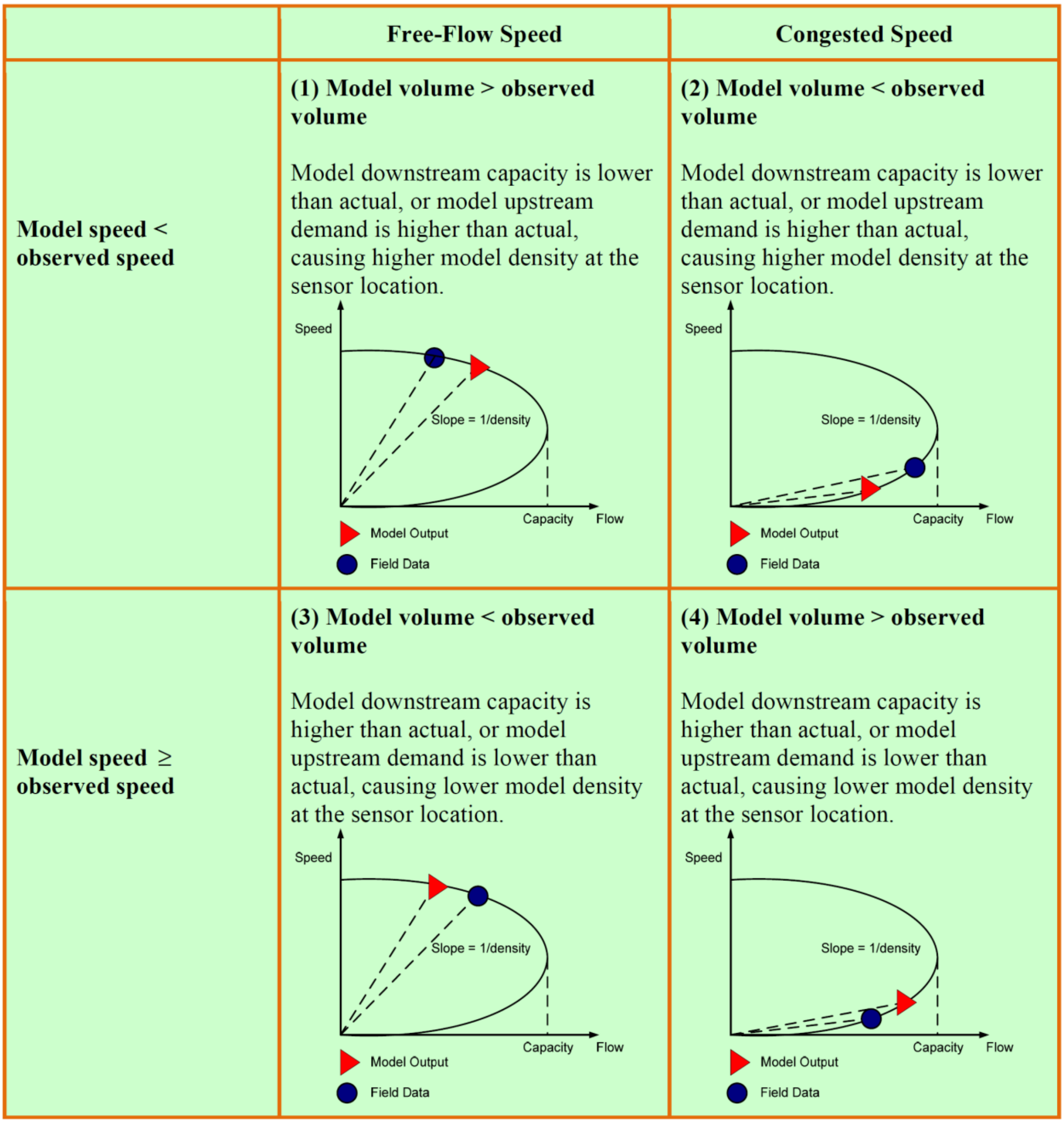


Figure 75: Kietzke St., southbound at 7:40 a.m., sensor counts are bigger than simulation counts

The underlying logic is applicable to the qualitative context as well as the quantitative context of the calibration process, as presented before. In addition to link counts, other empirical data such as speed are also important for understanding the source of discrepancy.

Before starting to interpret discrepancies between model volumes and traffic counts, it is imperative to understand a fundamental property of real traffic that is respected by dynamic models (but not by static models). As illustrated in Figure 76, when the model volume is higher than the observed volume, one needs to understand whether this occurs under a free-flow or congested regime by checking the speed data. If the speeds are in the state of free-flow, it means that the model link density is higher than actual density (see Condition 1 in Figure 76; density can be depicted from the reciprocal of the slope of the line connecting the traffic point and the origin, a smaller slope means a higher density). Higher model link density can be caused either by lower model downstream capacity or higher model upstream demand. If the speeds are in the congested regime, then the model density is lower than the observed (see Condition 4 in

Figure 76), caused by either higher model downstream capacity or lower model upstream demand [3].



	Free-Flow Speed	**Congested Speed**
Model speed < observed speed	**(1) Model volume > observed volume** Model downstream capacity is lower than actual, or model upstream demand is higher than actual, causing higher model density at the sensor location.	**(2) Model volume < observed volume** Model downstream capacity is lower than actual, or model upstream demand is higher than actual, causing higher model density at the sensor location.
Model speed ≥ observed speed	**(3) Model volume < observed volume** Model downstream capacity is higher than actual, or model upstream demand is lower than actual, causing lower model density at the sensor location.	**(4) Model volume > observed volume** Model downstream capacity is higher than actual, or model upstream demand is lower than actual, causing lower model density at the sensor location.

Figure 76: Network and demand effects contributions to discrepancies between model outputs and field data [3]

In the opposite case in which the model volume is lower than the observed counts, two separate conditions need to be examined as in Conditions 2 and 3 in Figure 76, depending on the prevailing speed regimes.

A discrepancy between model volumes and counts at a particular observation location (the plural here is used to refer to time-varying data) is essentially due to an imbalance in the model between capacity and demand. There are three basic factors that contribute to this imbalance: (a) local network capacity and control timing parameters; (b) local traffic demand because of the assignment process, in the form of route flows; and, (c) global demand as represented by the O-D matrix. Each of these three influences can be a possible source of error. Their respective contributions to a specific outlier can be determined by following a simple process of investigation and elimination. This process is discussed in detail later in this section but can be summarized as follows [3]:

1. Compare model volume and observed volume. If model volume is higher than the observed volume, Conditions 1 and 4 in Figure 76 apply. The next step is to check speed. If speed is congested speed, then Condition 4 applies. The discrepancy can be attributed to either higher downstream capacity or lower upstream inflow demand. To further isolate the cause, the diagnostic actions could include checking the local capacity and signal timing parameters upstream and downstream (as may be applicable) of the observation location. A similar process also applies to other volume discrepancy situations.

2. If the local capacities are not (or are no longer) a major contributing factor, the next step is to check the assignment, which is responsible for the local demand.

3. If the local capacities are correct and the assignment is acceptable, the only remaining possibility is the global O-D demand: it may have to be revised. Since the other two items were fine, OD demand was modified to improve the discrepancies in Reno/Sparks network volumes. However, the first two items are also explained as follows in Section 5.3.1and Section 5.3.2.

The next three sections will further discuss these influencing factors in order: capacity, assignment and demand.

5.3.1. Network connectivity and capacity effects

It should be noted that volume on a link or movement in a dynamic model will often be much lower than the simple theoretical capacity estimate (given, for example, by the product of the effective green, number of lanes, and per lane saturation flow rate for arterials). In general, a DTA model will account for many factors, and understanding the causes behind the model volumes is a key component of the calibration process [3].

The following steps should be checked:

a. Activity Location (node, zone)

The file input_activity_location.csv can check each node ID associated with the zone ID. These nodes represent the activity location.

i. Node (input_node.csv)

ii. Zone (input_zone.csv)

- Each zone has at least one activity location.
- Zone number used in OD demand table files must be defined in input_zone.csv first.

b. Link (input_link.csv)

The fields link type, length, speed, capacity and number of lanes for each link must be defined.

i. Checking the link connectivity:

- How the different link types are connected/related. For example, Freeway can connect to arterial using ramp. Problematic links will check in this part.
- How each link type is set up in the system based on capacity, speed and number of lanes

ii. Link Type (input_link_type.csv)

- capacity_adjustment_factor
- travel_time_bias_factor

The value of this field can be changed based on preference of using different link types ($0 \leq$ bias factor ≤ 1). When it is 1 for all the link types, it doesn't matter which route the driver will choose. However, if the priority is choosing freeway, this value must change. Also, due to connectivity of freeways to arterials with ramps, the value of ramps should change, too.

- approximate_cycle_length_in_second
- saturation_flow_rate_in_vhc_per_hour_per_lane

Determining the saturation flow rate can be a somewhat complicated matter and depends on roadway and traffic conditions. Therefore, the user can assume an ideal value for saturation flow rate.

c. Movement (input_movement.csv)

To model traffic movement in the different scenarios such as work zone, incident, and toll, the user can prohibit movement from specified nodes. In this case, the user needs to set profibited_flag to 1 to forbid vehicles to pass from that node.

i. Prohibited movements

- Shows whether there is any movement from selected node or not
- Can accept value 0 or 1 where 1 means to forbid vehicles from passing on that node

d. Output_summary (Validate travel model)

After running the simulation, all the results summarize in the file output_summary. This file includes information on network MOE, link type, number of agents, traffic assignment, ODME, and Link MOE.

- Validation of traffic assignment
- Validation of link performance
- Link type statistics (checking average lane capacity, speed limit and length for each link type).

5.3.2. Assignment procedure and its effect

Evaluating the impact of the assignment on a particular traffic-count outlier essentially consists of qualitative analysis techniques, as there is generally no empirical information about travelers' routes through the network. However, knowledge of the network is often sufficient to make reasonable and educated judgments that can go a long way, particularly for using route analysis to find more coding errors in the capacity-side (network and traffic control) inputs. Route analysis generally consists of two basic approaches: select-link route analysis (Figure 67) and O-D route analysis (Figure 71). As a general rule, the capacity-side data should be verified as much as possible (as discussed in the previous section) before commencing route analysis. Select-link analysis is commonly used in DTA modeling. This tool generates all routes including a link or turning movement (or, if desired, a specified combination of links and movements).

If the select-link routes seem reasonable and it is certain that the capacity-side data are correct, the problem is almost certainly due to excessive demand for one or more of the O-D pairs in question. Coming to this conclusion from select-link route information is not as unlikely as it may first appear. Depending on the location of the link or movement in question, the select-link analysis may in fact produce relatively few O-D pairs, and those O-D pairs may have relatively few reasonable alternative routes. If there is a route that is obviously unreasonable, or at least questionable, the next step is to visualize the full set of routes for the corresponding O-D pair. This process will likely indicate much more reasonable routes that are also being used, but which are probably carrying too little flow. In many cases, it will be immediately obvious that there is too much congestion on one of the reasonable routes, and it may turn out that result is due to an undiscovered capacity-side coding error. As a result of the objective of the assignment model—i.e., to minimize each traveler's travel time—the additional travel time due to the excessive congestion pushes the travelers to use the unconventional route(s). If no particular causes of excessive travel time on the more reasonable routes are identified, and if it is certain that the capacity-side data are correct, the only remaining option is that there is excessive O-D demand for one or more of the O-D pairs in question. Nevertheless, it is desirable first to eliminate unreasonable or unrealistic routes from the choice set to ensure that future scenario runs do not run into similar problems. While performing this task, care should be taken to include alternative routes that may become reasonable when regularly used links are disabled through incidents, work zones, etc. If using movement-based (intersection turning) counts for the analysis, and if the movement flow is shown to be near capacity in both models and actual data, the upstream congestion could still be rather different. This is a good example of why it is often said that speed or queue length data are essential in addition to traffic count data [3]. The next section will explain assignment and demand validation and modification together.

5.3.3. Demand modification and its effects

As discussed in the Assignment Effects section above, it is possible in some situations to draw definitive conclusions about excessive demand for certain O-D pairs from a rigorous investigation of traffic-count outliers, preferably in conjunction with empirical data about link speeds or queue lengths. In these cases, how much to adjust a specific O-D demand value is largely a matter of common sense and modeling judgment. Automated processes and algorithms for adjusting dynamic O-D matrices using empirical data are an active area of academic research. Some existing DTA models, in conjunction with optimization formulations and solvers, allow systematic adjustment of O-D matrices for multiple vehicle classes on large networks with thousands of zones. Nevertheless, manual O-D adjustment is also commonly used in calibrating a DTA model. It should also be noted that, unlike the well-defined physical properties of roads and traffic signals, there is often a significant degree of uncertainty and day-to-day variability in the O-D demand data.

The followings explain the procedure explained before with more details about the Reno/Sparks model.

Dynamic Traffic Assignment was run with: MSA (40 days) + ODME (160 days). The following steps were taken:

Step1. Data preparation as described for the following:

- Traffic network data (node, link, zone, activity_location)
- Travel Demand data (demand, demand type, demand_file_list)
- Sensor data (sensor_count, sensor_speed)

Step 2. The 2nd step is to setup files input_scenario_setting.csv and input_demand_file_list.csv with the data shown in Tables 16 and 17.

Table 16: input_scenario_setting.csv

scenario_no	**scenario_name**	**number_of_assignment_days**	**number_of_statistics_reporting_days**	**traffic_flow_model**	**signal_representation_model**	**traffic_assignment_method**	**demand_multiplier**
1	1	100	1	1	0	3	100
random_seed	**ODME_start_iteration**	**ODME_max_percentage_deviation_wrt_hist_demand**	**ODME_step_size,ue_gap_calculation_method**	**calibration_data_start_time_in_min**	**calibration_data_end_time_in_min**	**routing_movement_delay_mode**	
40	200	30	0.05	0	300	600	

Table 17: input_demand_file_list.csv

scenario_no	**file_sequence_no**	**file_name**	**format_type**	**number_of_lines_to_be_skipped**
0	7	trips_am(da).csv	matrix	1
loading_multiplier	**subtotal_in_last_column**	**demand_type_in_3rd_column**	**start_time_in_min**	**end_time_in_min**
1.5	0	0	300	600

For demand data, the followings should be considered:

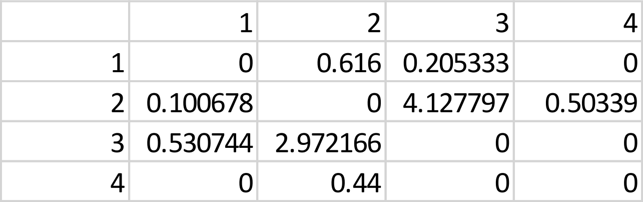
1. Specify the demand file name like “trips_am(da).csv” and format, in this case format is matrix.

2. Specify loading_multiplier for different demand files.

Loading_multiplier is a local multiplication factor applied to the number of trips in the demand file. It is usually provided for each demand loading time and it is applied to each row of the demand file.

Table 18 shows a sample of the OD table (matrix format) that stored number of trips between 6 a.m. to 9 a.m.

Table 18: A sample of OD Demand Table



	1	2	3	4
1	0	0.616	0.205333	0
2	0.100678	0	4.127797	0.50339
3	0.530744	2.972166	0	0
4	0	0.44	0	0

In this case, the loading multiplier is 1.5 (Table 19). The number of trips from zone 2 to zone 3 is 4.1 vehicles. Therefore, the total demand from zone 2 to zone 3 is 1.5 × 4.1 = 6.15 vehicles (loading multiplier × demand of corresponding zones). The 6.15 vehicles will be randomly rounded up or down according to the uniform distribution to convert to an integer number in the final simulation.

3. Specify the loading start time and end time for the demand file (300 to 600, or 5 a.m. to 10 a.m.)

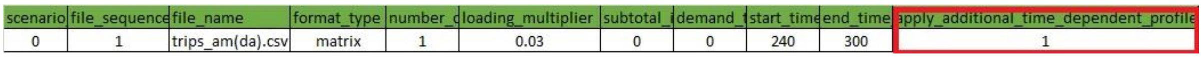
Table 19: input_demand_file_list.csv. Loading multiplier is set 1.5 for a.m. peak

scenario_	file_seque	file_name	format_ty	number_d	loading_multiplier	subtotal_i	demand_t	start_time	end_time	apply_additional_time_dependent_	number_d	demand_t
0	1	trips_am(da).csv	matrix	1	0.03	0	0	240	300	1	1	1
0	2	trips_am(sr2).csv	matrix	1	0.03	0	0	240	300	1	1	2
0	3	trips_am(sr3).csv	matrix	1	0.03	0	0	240	300	1	1	2
0	4	trips_am(local_bus).csv	matrix	1	0.03	0	0	240	300	1	1	4
0	5	trips_am(multiUnitTrucks).csv	matrix	1	0.03	0	0	240	300	1	1	3
0	6	trips_am(singleUnitTrucks).csv	matrix	1	0.03	0	0	240	300	1	1	3
0	7	trips_am(da).csv	matrix	1	1.5	0	0	300	600	1	1	1
0	8	trips_am(sr2).csv	matrix	1	1.5	0	0	300	600	1	1	2
0	9	trips_am(sr3).csv	matrix	1	1.5	0	0	300	600	1	1	2
0	10	trips_am(local_bus).csv	matrix	1	1.5	0	0	300	600	1	1	4
0	11	trips_am(multiUnitTrucks).csv	matrix	1	1.5	0	0	300	600	1	1	3
0	12	trips_am(singleUnitTrucks).csv	matrix	1	1.5	0	0	300	600	1	1	3
0	13	trips_am(da).csv	matrix	1	0.22	0	0	600	660	1	1	1
0	14	trips_am(sr2).csv	matrix	1	0.22	0	0	600	660	1	1	2
0	15	trips_am(sr3).csv	matrix	1	0.22	0	0	600	660	1	1	2
0	16	trips_am(local_bus).csv	matrix	1	0.22	0	0	600	660	1	1	4
0	17	trips_am(multiUnitTrucks).csv	matrix	1	0.22	0	0	600	660	1	1	3
0	18	trips_am(singleUnitTrucks).csv	matrix	1	0.22	0	0	600	660	1	1	3

4. Apply time-dependent profile

Set the field apply_additional_time_dependent_profile to 1, then a time-dependent departure time profile will be read from the time profile.

Table 20: Setting up time-dependent profile in file input_demand_file_list.csv



scenario	file_sequence	file_name	format_type	number_d	loading_multiplier	subtotal_i	demand_t	start_time	end_time	apply_additional_time_dependent_profile
0	1	trips_am(da).csv	matrix	1	0.03	0	0	240	300	1

To calculate a time-dependent profile, suppose we have demands for each hour as shown in Table 21, so the distribution rate for each hour can be calculated from this formula $\frac{Hourly\ demand}{Total\ demand}$. For instance, demand from 5 a.m. to 6 a.m. is 170841 vehicles and total demand from 5 a.m. to 10 a.m. is 2305445, so the hourly loading multiplier can calculate $\frac{170841}{2305445} = 0.074$. Because the time profile has 15 min time intervals, it will be divided by 4. Table 22 shows the time dependent profile.

Table 21: Demand File Distribution from 5 am to 10 am

	5:00 to 6:00	6:00 to 7:00	7:00 to 8:00	8:00 to 9:00	9:00 to 10:00
Demand	170841	380877	631721	582174	539832
Hourly Loading multiplier	0.074	0.165	0.274	0.253	0.234
15 min time interval loading multiplier	0.019	0.041	0.069	0.063	0.059
Total demand from 5 am to 10 am	2305445				

Table 22: Time Dependent Profile pattern

'04:00	'04:15	'04:30	'04:45	'05:00	'05:15	'05:30	'05:45	'06:00	'06:15	'06:30	'06:45	'07:00	'07:15	'07:30	'07:45	'08:00	'08:15	'08:30	'08:45	'09:00	'09:15	'09:30	'09:45	'10:00	'10:15	'10:30	'10:45
0.007	0.007	0.007	0.007																								
0.007	0.007	0.007	0.007																								
0.007	0.007	0.007	0.007																								
0.007	0.007	0.007	0.007																								
0.007	0.007	0.007	0.007																								
0.007	0.007	0.007	0.007																								
				0.019	0.019	0.019	0.019	0.041	0.041	0.041	0.041	0.069	0.069	0.069	0.069	0.063	0.063	0.063	0.063	0.059	0.059	0.059	0.059				
				0.019	0.019	0.019	0.019	0.041	0.041	0.041	0.041	0.069	0.069	0.069	0.069	0.063	0.063	0.063	0.063	0.059	0.059	0.059	0.059				
				0.019	0.019	0.019	0.019	0.041	0.041	0.041	0.041	0.069	0.069	0.069	0.069	0.063	0.063	0.063	0.063	0.059	0.059	0.059	0.059				
				0.019	0.019	0.019	0.019	0.041	0.041	0.041	0.041	0.069	0.069	0.069	0.069	0.063	0.063	0.063	0.063	0.059	0.059	0.059	0.059				
				0.019	0.019	0.019	0.019	0.041	0.041	0.041	0.041	0.069	0.069	0.069	0.069	0.063	0.063	0.063	0.063	0.059	0.059	0.059	0.059				
				0.019	0.019	0.019	0.019	0.041	0.041	0.041	0.041	0.069	0.069	0.069	0.069	0.063	0.063	0.063	0.063	0.059	0.059	0.059	0.059				
																								0.057	0.057	0.057	0.057
																								0.057	0.057	0.057	0.057
																								0.057	0.057	0.057	0.057
																								0.057	0.057	0.057	0.057
																								0.057	0.057	0.057	0.057
																								0.057	0.057	0.057	0.057

Loading multiplier and time dependent multiplier factors are two separated multipliers, because they can accept different values. If the loading multiplier is 2, it doesn’t mean the summation of the ratio in the time profile pattern should be 2. The time-dependent multiplier is a cumulative factor for which the total is equal to 1.

Step 3. Run the simulation. Conduct traffic assignment 1 hour before as a warm up and 1 hour after to smooth the distribution. Run the simulation for 200 days, 40 days historical data (reaching UE) and 160 days ODME. The following cases with different loading multipliers were run. The results are shown for each case.

Case 1: loading multiplier 1

$R^2 = 0.79$, Y = 0.6709x

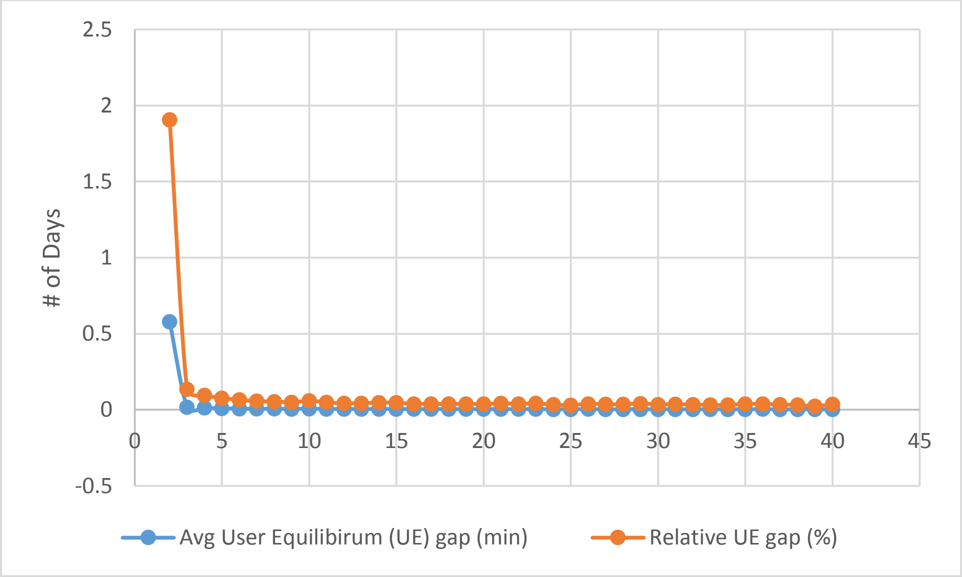
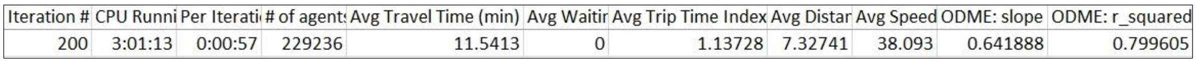


Figure 77: Trend of the Average User Equilibrium Gap and Relative User Equilibrium Gap (Loading multiplier =1)

Table 23: Last day Info in ouput_summary.csv (Loading multiplier =1)



Iteration #	CPU Runni	Per Iterati	# of agent	Avg Travel Time (min)	Avg Waitir	Avg Trip Time Index	Avg Distar	Avg Speed	ODME: slope	ODME: r_squared
200	3:01:13	0:00:57	229236	11.5413	0	1.13728	7.32741	38.093	0.641888	0.799605

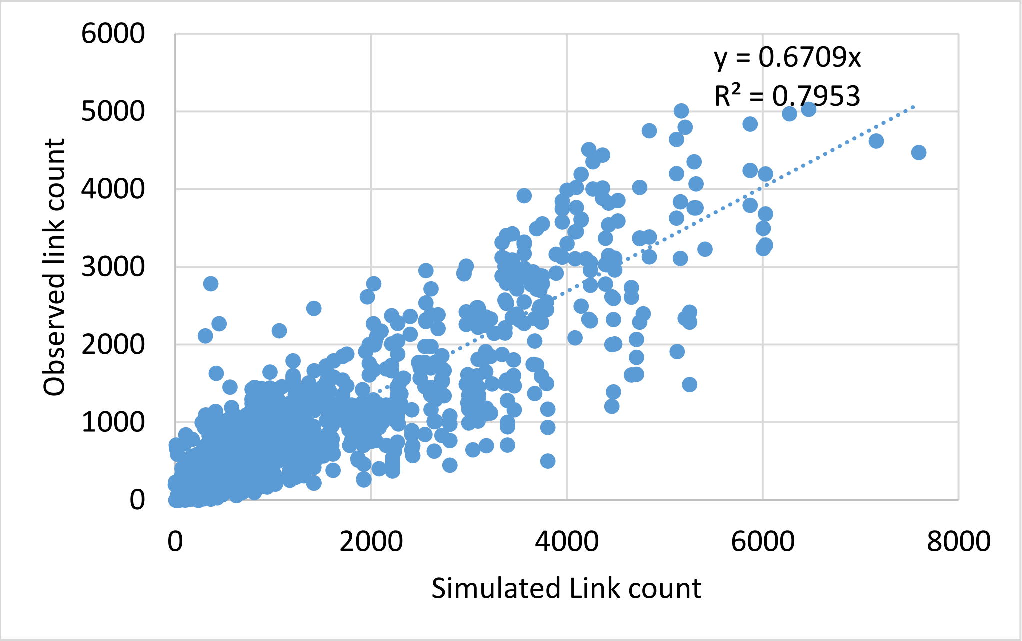


Figure 78: Observed vs simulated link volume (loading multiplier=1)

Case 2: loading multiplier 1.2

$R^2 = 0.84$, Y =0.7592x

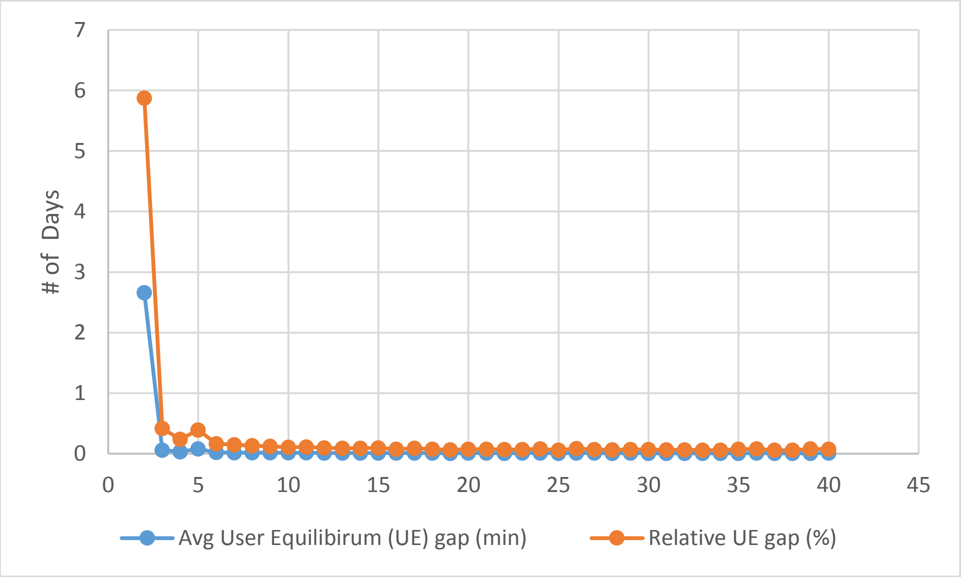
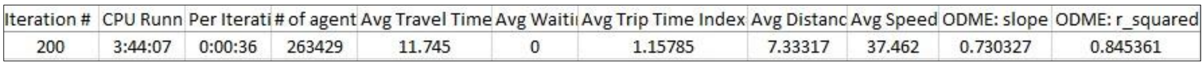


Figure 79: Trend of the Average User Equilibrium Gap and Relative User Equilibrium Gap (Loading multiplier =1.2)

Table 24: Last day Info in ouput_summary.csv (Loading multiplier =1.2)



Iteration #	CPU Runn	Per Iterati	# of agent	Avg Travel Time	Avg Waitii	Avg Trip Time Index	Avg Distanc	Avg Speed	ODME: slope	ODME: r_squared
200	3:44:07	0:00:36	263429	11.745	0	1.15785	7.33317	37.462	0.730327	0.845361

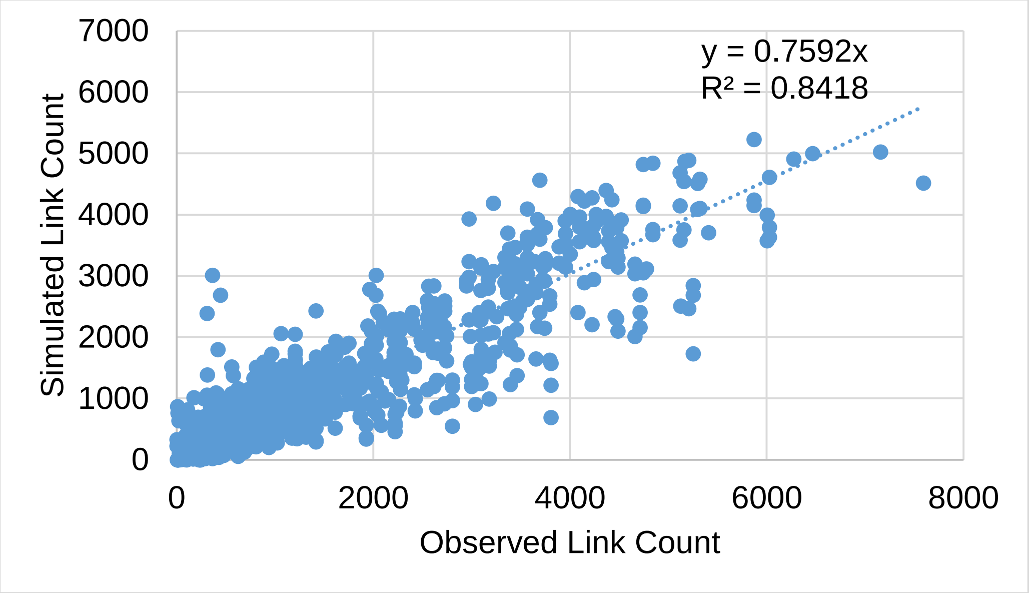


Figure 80: Observed vs simulated link volume (loading multiplier=1.2)

Case 3: loading multiplier 1.3

$R^2 = 0.85$, Y = 0.787x

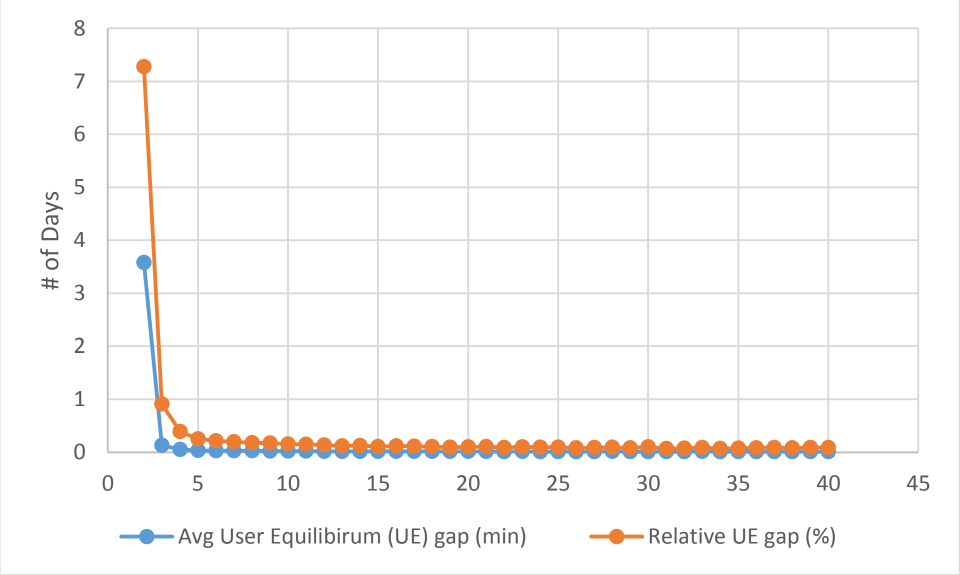
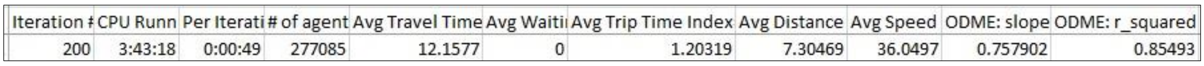


Figure 81: Trend of the Average User Equilibrium Gap and Relative User Equilibrium Gap (Loading multiplier =1.3)

Table 25: Last day Info in ouput_summary.csv (Loading multiplier =1.3)



Iteration #	CPU Runn	Per Iterati	# of agent	Avg Travel Time	Avg Waiti	Avg Trip Time Index	Avg Distance	Avg Speed	ODME: slope	ODME: r_squared
200	3:43:18	0:00:49	277085	12.1577	0	1.20319	7.30469	36.0497	0.757902	0.85493

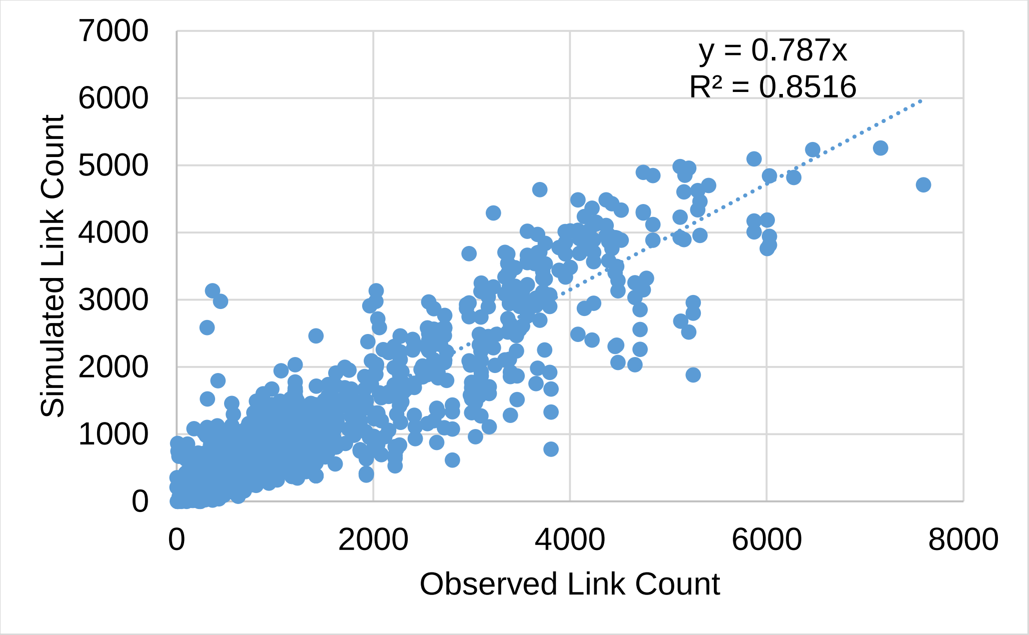


Figure 82: Observed vs simulated link volume (loading multiplier=1.3)

Case 4: loading multiplier 1.5

R^2 = 0.8756, Y =0.8252x

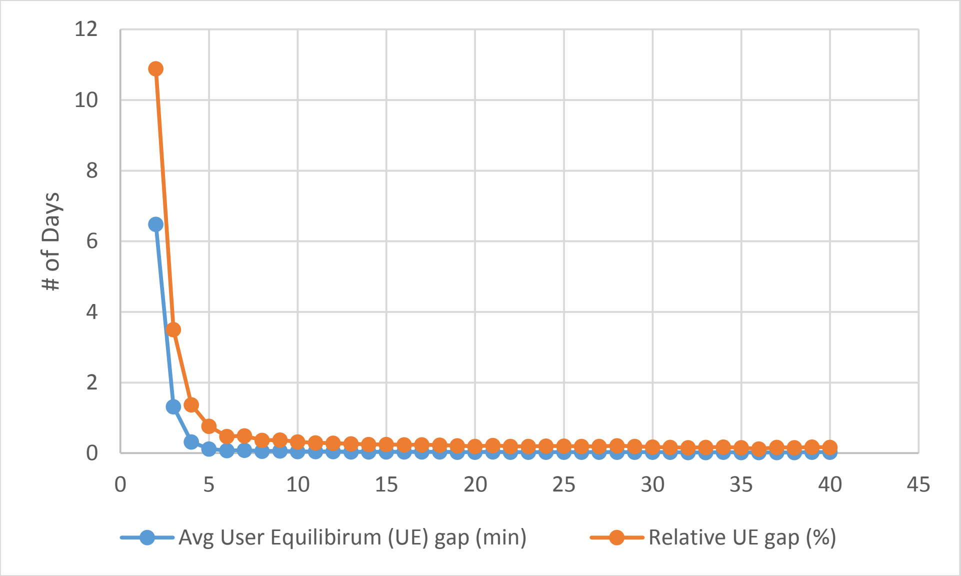
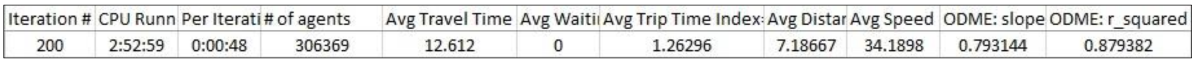


Figure 83: Trend of the Average User Equilibrium Gap and Relative User Equilibrium Gap (Loading multiplier =1.5)

Table 26: Last day Info in ouput_summary.csv (Loading multiplier =1.5)



Iteration #	CPU Runn	Per Iterati	# of agents	Avg Travel Time	Avg Waiti	Avg Trip Time Index	Avg Distar	Avg Speed	ODME: slope	ODME: r_squared
200	2:52:59	0:00:48	306369	12.612	0	1.26296	7.18667	34.1898	0.793144	0.879382

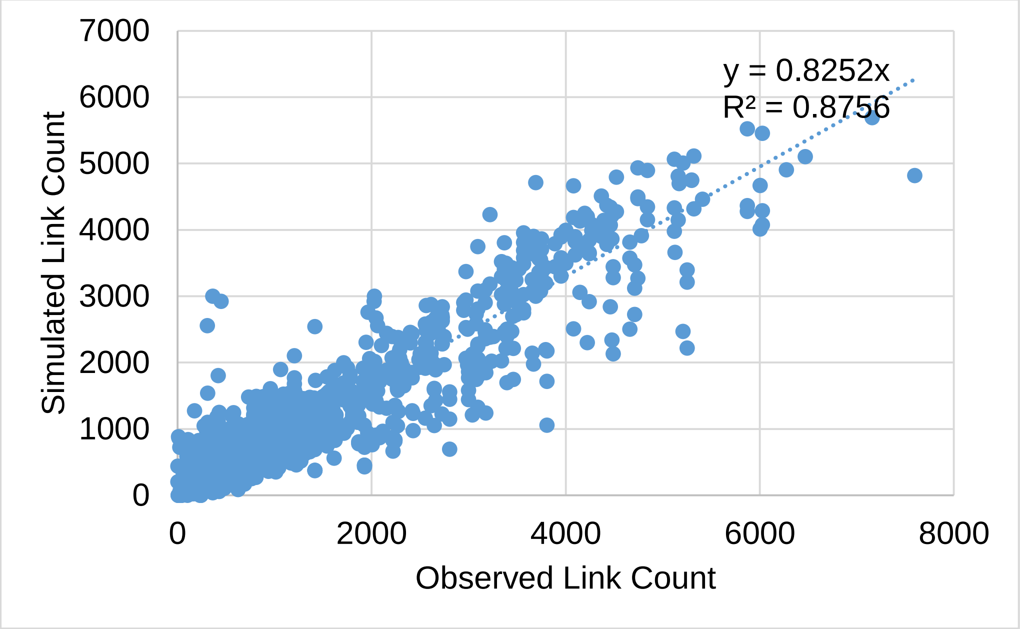


Figure 84: Observed vs simulated link volume (loading multiplier=1.5)

Case 5: loading multiplier 1.6

$R^2 = 0.8815$, Y = 0.8392

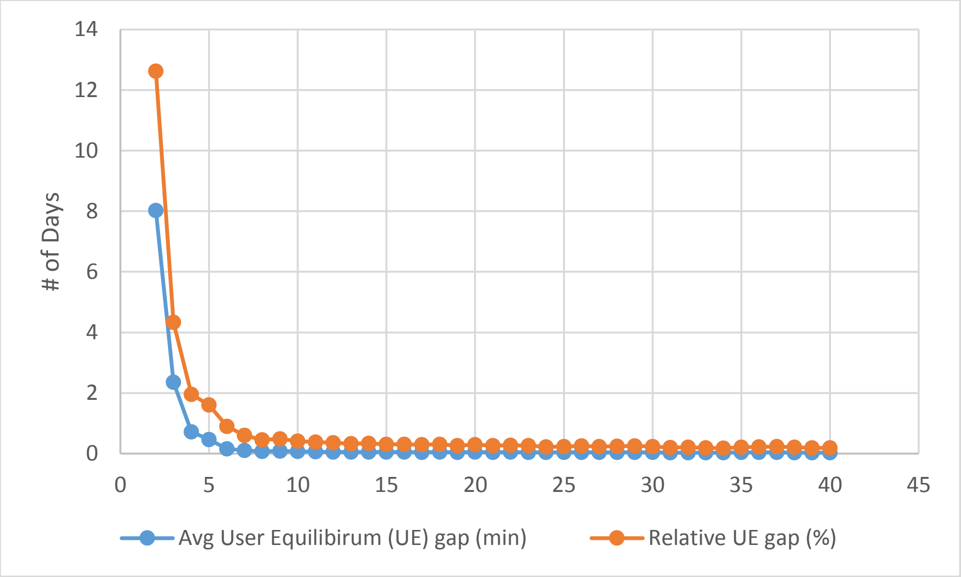
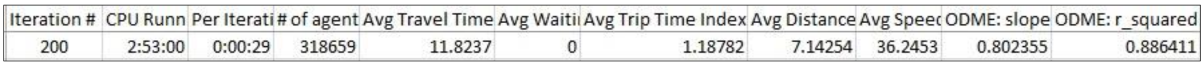


Figure 85: Trend of the Average User Equilibrium Gap and Relative User Equilibrium Gap (Loading multiplier =1.6)

Table 27: Last day Info in ouput_summary.csv (Loading multiplier =1.6)



Iteration #	CPU Runn	Per Iterati	# of agent	Avg Travel Time	Avg Waitii	Avg Trip Time Index	Avg Distance	Avg Speec	ODME: slope	ODME: r_squared
200	2:53:00	0:00:29	318659	11.8237	0	1.18782	7.14254	36.2453	0.802355	0.886411

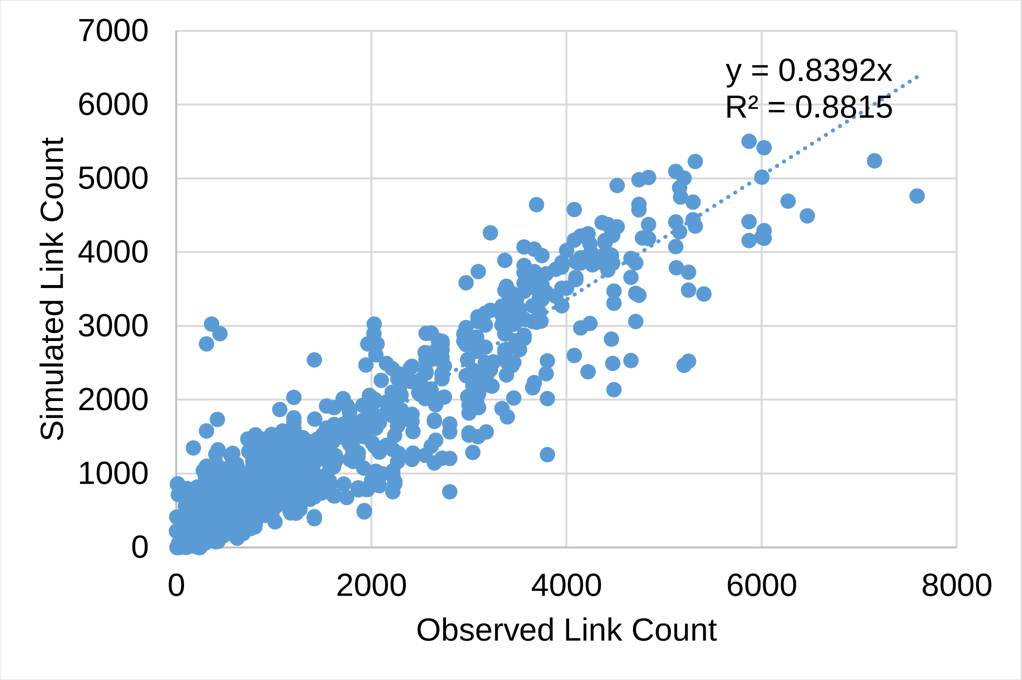


Figure 86: Observed vs simulated link volume (loading multiplier=1.6)

It should be noted that given the same total demand volume (number of cars), a time sliced matrix that exhibits a time-varying demand profile (Figure 87) will inevitably generate more congestion than a flat (static) demand that was imported from TransCAD for the Reno/Sparks model. This is because it is the peak of the demand profile that will represent the peak loading conditions, which may result in congestion or even gridlock. This is why many links were showing simulated volumes larger than sensor counts before applying the loading multiplier and time dependent profile pattern (Table 22). The same number of cars spread out evenly in time will have the lowest possible peak loading conditions.

Time-varying O-D inputs were estimated from sensor counts. This topic continues to receive significant research attention, as there are typically far too many O-D flow variables to allow manual adjustments within a reasonable time frame. Those links with observed data will also constitute only a subset of all possible O-D pairs. O-D calibration–adjustment is simple in concept. A starting set of O-D matrices is loaded into the DTA model. At convergence, its outputs are compared against the observed data (mostly counts). An objective function is evaluated to quantify this fit (usually the total count differences over all observed links), and its value is used in some systematic way to adjust the O-D flows automatically. The adjusted O-D matrices are loaded into the DTA model again and run to reach convergence [3].

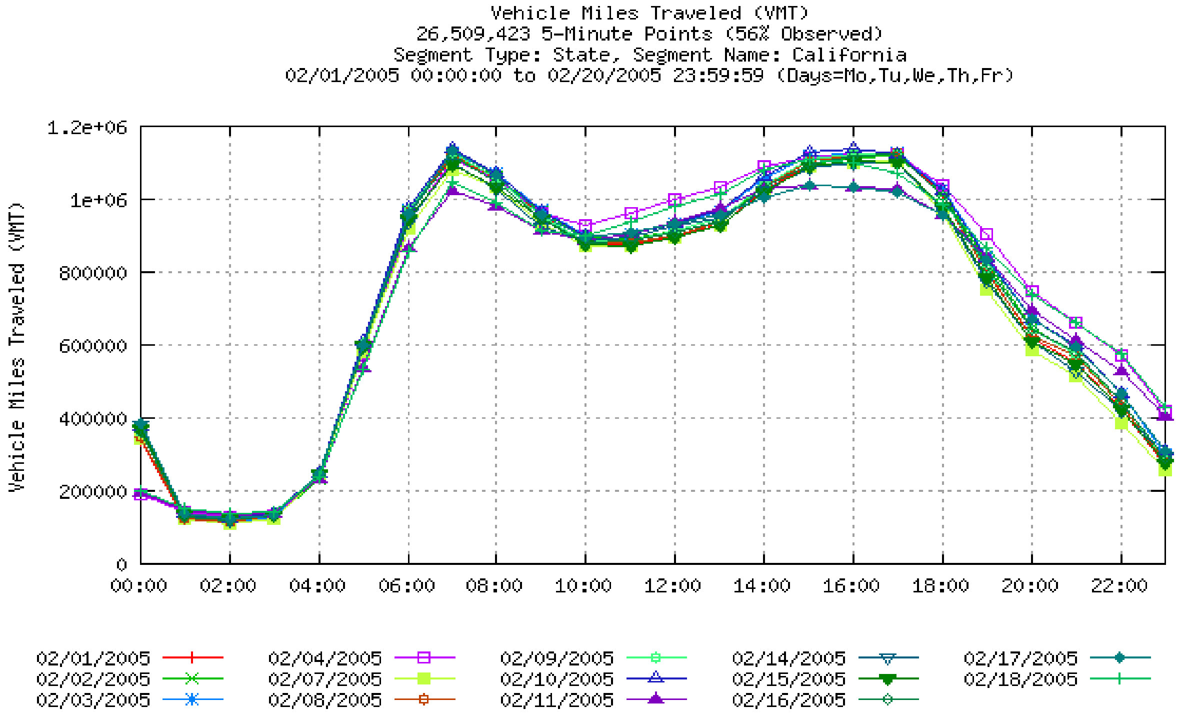


Figure 87: A time varying demand profile

The following should be checked for validating demand accuracy and modification:

a. Demand type (input_demand_type.csv)

- Used to define the characteristics for different demand types. Demand types are single operation vehicle (SOV), high operation vehicle (HOV), and trucks by default, but one can define additional types.
- Average_VOT: Average value of time for each demand type defines in this file.
- Fields of percentage of pretrip/enroute info will affect routing behavior

b. Demand table format (column, matrix, agent-base)

i. Column

Including 3 columns of from zone, to zone, and demand value

ii. Matrix

iii. Agent-base

c. Time-dependent demand loading patterns: user can define the loading multiplier for each 15-minute time interval. At first, the user should set apply_additional_time_dependent_profile to 1 in file demand_file_list.

d. Configuration of input_demand_file_list.csv

e. Compare simulated results with sensor data in 24-hour time period

Are the number of vehicles simulated reasonable?

f. Static traffic assignment: in static traffic assignment, focus on total vehicle miles traveled (VMT), total vehicle hours of travel (VHT)

g. Dynamic traffic assignment

Average travel time and average distance

6. SCENARIO ANALYSIS

For performing scenario analysis, such as work zone, incident, ramp metering, etc., the travel demand should be calibrated in advance. In this section, the integration of the two parts above is realized in DTALite through just one simulation, the process of which can be illustrated in Figure 88.

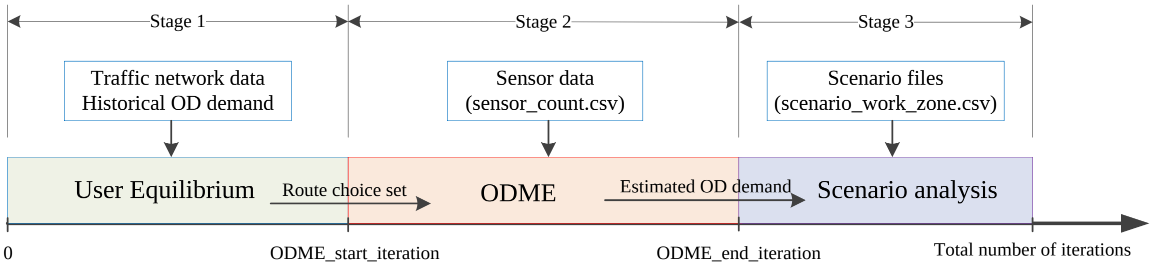


Figure 88: The process of one simulation for scenario analysis using estimated OD demand

At Stage 1, the user equilibrium is reached on the basis of traffic network data and historical OD demand and the route choice set is generated for the path flow adjustment at Stage 2, where observed sensor data is input and used for OD demand estimation. Based on the estimated OD demand, the scenario analysis is performed for traffic state prediction or new user equilibrium condition searching.

The case study was performed on the Virginia St. bridge construction (Figure 89). Its data was collected by NDOT (Figure 90). The work zone occurs on links 11083→1645 and 1645→11083 (Figure 91). The detailed data are input in file Scenario_Work_Zone.csv shown in Figure 92.

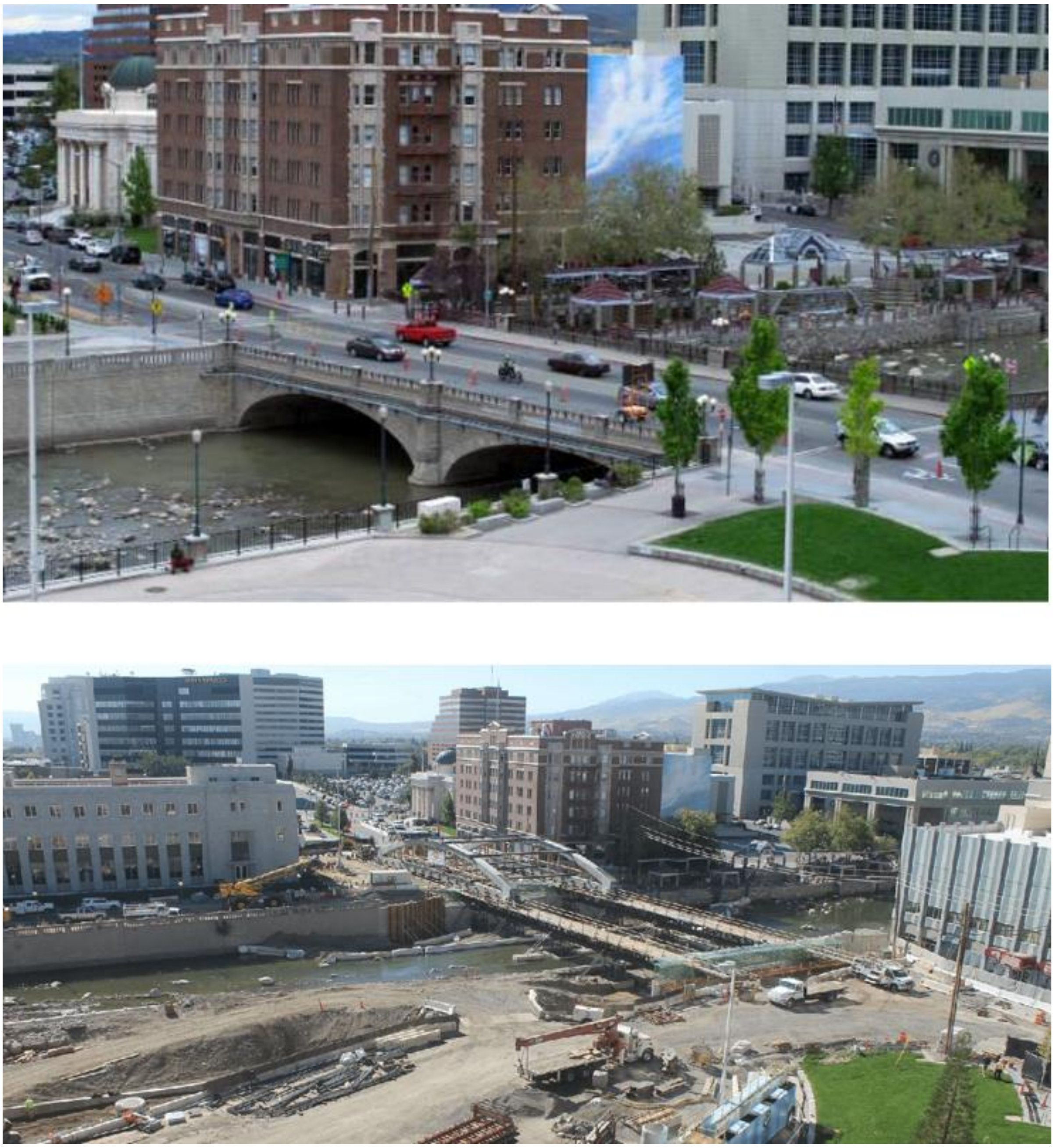
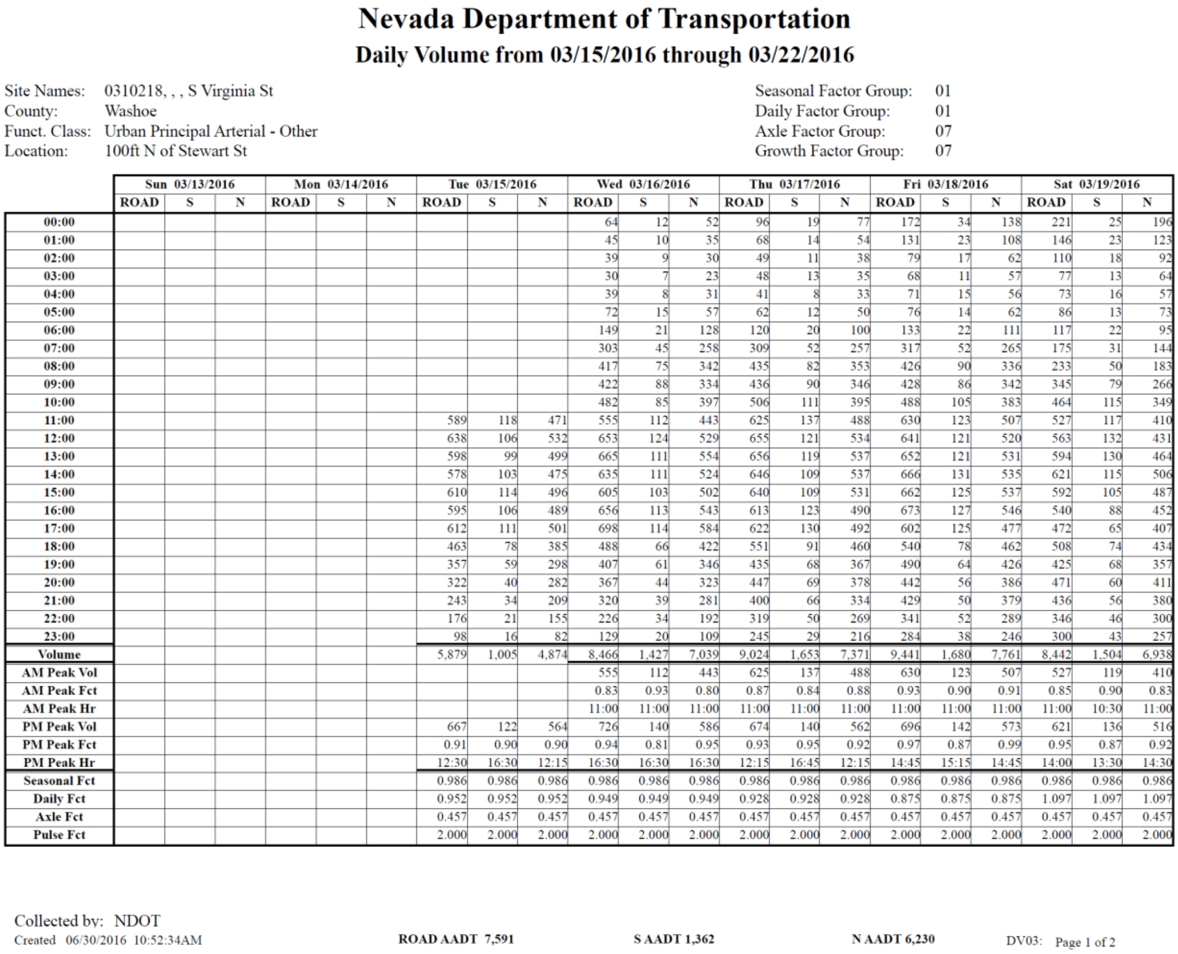


Figure 89: Case study region. Above: before bridge construction. Below: During bridge construction



Nevada Department of Transportation

Daily Volume from 03/15/2016 through 03/22/2016

Site Names: 0310218, , , S Virginia St
County: Washoe
Funct. Class: Urban Principal Arterial - Other
Location: 100ft N of Stewart St

Seasonal Factor Group: 01
Daily Factor Group: 01
Axle Factor Group: 07
Growth Factor Group: 07

	Sun 03/13/2016			Mon 03/14/2016			Tue 03/15/2016			Wed 03/16/2016			Thu 03/17/2016			Fri 03/18/2016			Sat 03/19/2016		
	ROAD	S	N	ROAD	S	N	ROAD	S	N	ROAD	S	N	ROAD	S	N	ROAD	S	N	ROAD	S	N
00:00										64	12	52	96	19	77	172	34	138	221	25	196
01:00										45	10	35	68	14	54	131	23	108	146	23	123
02:00										39	9	30	49	11	38	79	17	62	110	18	92
03:00										30	7	23	48	13	35	68	11	57	77	13	64
04:00										39	8	31	41	8	33	71	15	56	73	16	57
05:00										72	15	57	62	12	50	76	14	62	86	13	73
06:00										149	21	128	120	20	100	133	22	111	117	22	95
07:00										303	45	258	309	52	257	317	52	265	175	31	144
08:00										417	75	342	435	82	353	426	90	336	233	50	183
09:00										422	88	334	436	90	346	428	86	342	345	79	266
10:00										482	85	397	506	111	395	488	105	383	464	115	349
11:00							589	118	471	555	112	443	625	137	488	630	123	507	527	117	410
12:00							638	106	532	653	124	529	655	121	534	641	121	520	563	132	431
13:00							598	99	499	665	111	554	656	119	537	652	121	531	594	130	464
14:00							578	103	475	635	111	524	646	109	537	666	131	535	621	115	506
15:00							610	114	496	605	103	502	640	109	531	662	125	537	592	105	487
16:00							595	106	489	656	113	543	613	123	490	673	127	546	540	88	452
17:00							612	111	501	698	114	584	622	130	492	602	125	477	472	65	407
18:00							463	78	385	488	66	422	551	91	460	540	78	462	508	74	434
19:00							357	59	298	407	61	346	435	68	367	490	64	426	425	68	357
20:00							322	40	282	367	44	323	447	69	378	442	56	386	471	60	411
21:00							243	34	209	320	39	281	400	66	334	429	50	379	436	56	380
22:00							176	21	155	226	34	192	319	50	269	341	52	289	346	46	300
23:00							98	16	82	129	20	109	245	29	216	284	38	246	300	43	257
Volume							5,879	1,005	4,874	8,466	1,427	7,039	9,024	1,653	7,371	9,441	1,680	7,761	8,442	1,504	6,938
AM Peak Vol										555	112	443	625	137	488	630	123	507	527	119	410
AM Peak Fct										0.83	0.93	0.80	0.87	0.84	0.88	0.93	0.90	0.91	0.85	0.90	0.83
AM Peak Hr										11:00	11:00	11:00	11:00	11:00	11:00	11:00	11:00	11:00	11:00	10:30	11:00
PM Peak Vol							667	122	564	726	140	586	674	140	562	696	142	573	621	136	516
PM Peak Fct							0.91	0.90	0.90	0.94	0.81	0.95	0.93	0.95	0.92	0.97	0.87	0.99	0.95	0.87	0.92
PM Peak Hr							12:30	16:30	12:15	16:30	16:30	16:30	12:15	16:45	12:15	14:45	15:15	14:45	14:00	13:30	14:30
Seasonal Fct							0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986	0.986
Daily Fct							0.952	0.952	0.952	0.949	0.949	0.949	0.928	0.928	0.928	0.875	0.875	0.875	1.097	1.097	1.097
Axle Fct							0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457	0.457
Pulse Fct							2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000

Collected by: NDOT
Created 06/30/2016 10:52:34AM

ROAD AADT 7,591 **S AADT 1,362** **N AADT 6,230** DV03: Page 1 of 2

Figure 90: A sample of data collected for Virginia St. case study

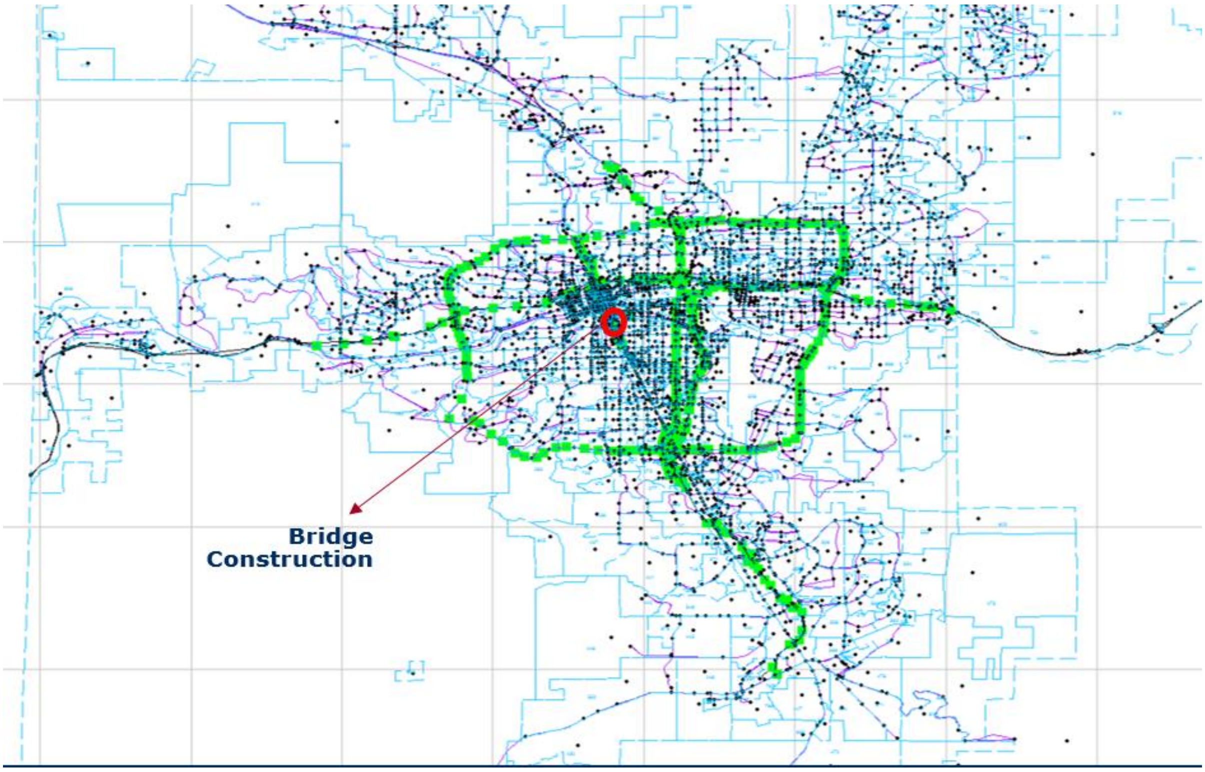
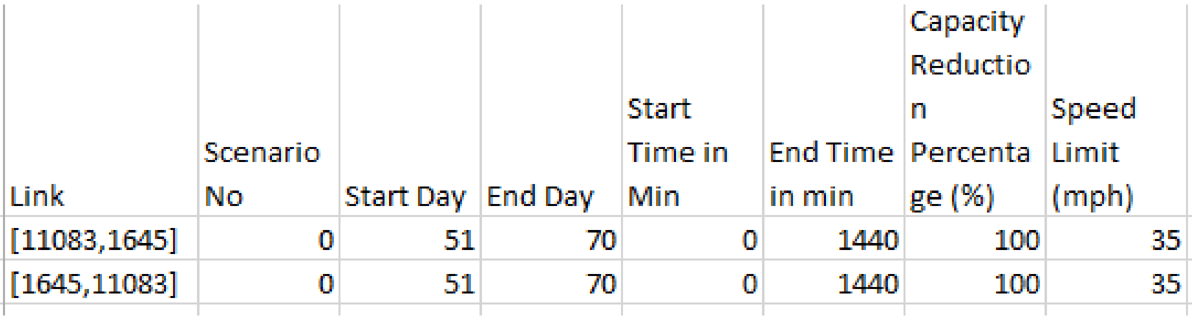


Figure 91: Bridge construction region on NeXTA environment

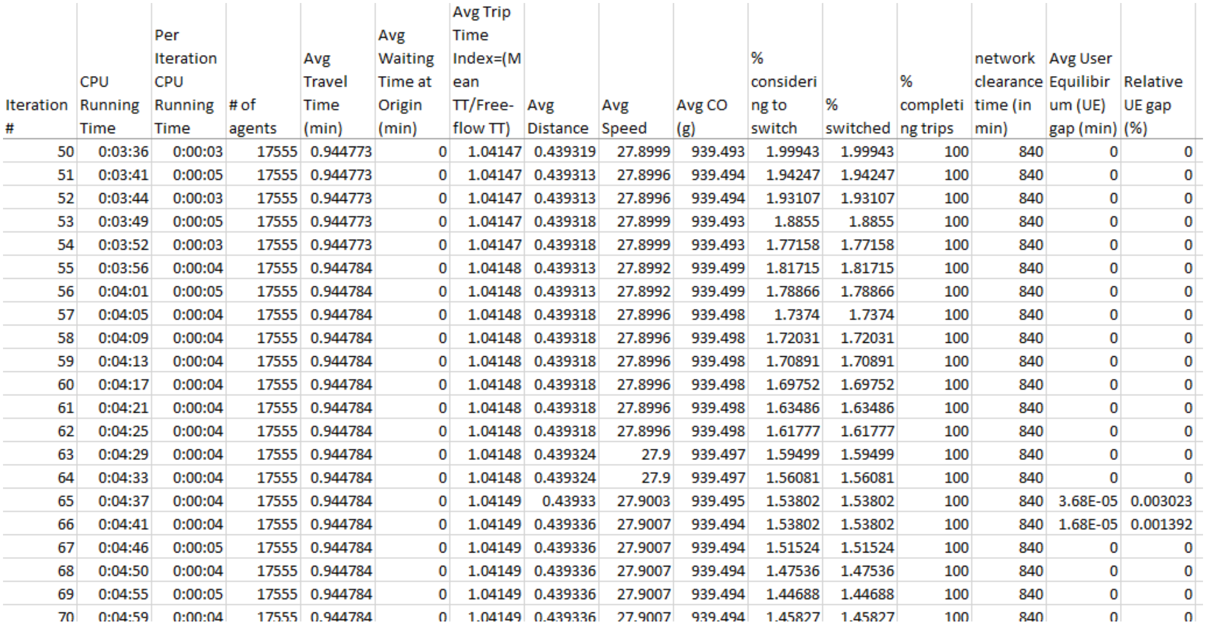


Link	Scenario No	Start Day	End Day	Start Time in Min	End Time in min	Capacity Reduction Percentage (%)	Speed Limit (mph)
[11083,1645]	0	51	70	0	1440	100	35
[1645,11083]	0	51	70	0	1440	100	35

Figure 92: Input data of Scenario_Work_Zone.csv

As shown in Figure 92, “Start Day” is 51, which indicates that the work zone will happen at the 51th iteration and the first 50 iterations is the process of ODME. As a result, the value of “number_of_assignment_days” should be set to 70 to correspond with “End Day” in file input_scenario_settings.csv.

The simulation result of the 50th iteration and scenario iterations (51 to 70) in output_summary.csv is shown in Figure 93. It is observed that this work zone does not have a significant effect on the average speed and the average travel.



Iteration #	CPU Running Time	Per Iteration CPU Running Time	# of agents	Avg Travel Time (min)	Avg Waiting Time at Origin (min)	Avg Trip Time Index=(Mean TT/Free-flow TT)	Avg Distance	Avg Speed	Avg CO (g)	% considering to switch	% switched	% completing trips	network clearance time (in min)	Avg User Equilibrium (UE) gap (min)	Relative UE gap (%)
50	0:03:36	0:00:03	17555	0.944773	0	1.04147	0.439319	27.8999	939.493	1.99943	1.99943	100	840	0	0
51	0:03:41	0:00:05	17555	0.944773	0	1.04147	0.439313	27.8996	939.494	1.94247	1.94247	100	840	0	0
52	0:03:44	0:00:03	17555	0.944773	0	1.04147	0.439313	27.8996	939.494	1.93107	1.93107	100	840	0	0
53	0:03:49	0:00:05	17555	0.944773	0	1.04147	0.439318	27.8999	939.493	1.8855	1.8855	100	840	0	0
54	0:03:52	0:00:03	17555	0.944773	0	1.04147	0.439318	27.8999	939.493	1.77158	1.77158	100	840	0	0
55	0:03:56	0:00:04	17555	0.944784	0	1.04148	0.439313	27.8992	939.499	1.81715	1.81715	100	840	0	0
56	0:04:01	0:00:05	17555	0.944784	0	1.04148	0.439313	27.8992	939.499	1.78866	1.78866	100	840	0	0
57	0:04:05	0:00:04	17555	0.944784	0	1.04148	0.439318	27.8996	939.498	1.7374	1.7374	100	840	0	0
58	0:04:09	0:00:04	17555	0.944784	0	1.04148	0.439318	27.8996	939.498	1.72031	1.72031	100	840	0	0
59	0:04:13	0:00:04	17555	0.944784	0	1.04148	0.439318	27.8996	939.498	1.70891	1.70891	100	840	0	0
60	0:04:17	0:00:04	17555	0.944784	0	1.04148	0.439318	27.8996	939.498	1.69752	1.69752	100	840	0	0
61	0:04:21	0:00:04	17555	0.944784	0	1.04148	0.439318	27.8996	939.498	1.63486	1.63486	100	840	0	0
62	0:04:25	0:00:04	17555	0.944784	0	1.04148	0.439318	27.8996	939.498	1.61777	1.61777	100	840	0	0
63	0:04:29	0:00:04	17555	0.944784	0	1.04148	0.439324	27.9	939.497	1.59499	1.59499	100	840	0	0
64	0:04:33	0:00:04	17555	0.944784	0	1.04148	0.439324	27.9	939.497	1.56081	1.56081	100	840	0	0
65	0:04:37	0:00:04	17555	0.944784	0	1.04149	0.43933	27.9003	939.495	1.53802	1.53802	100	840	3.68E-05	0.003023
66	0:04:41	0:00:04	17555	0.944784	0	1.04149	0.439336	27.9007	939.494	1.53802	1.53802	100	840	1.68E-05	0.001392
67	0:04:46	0:00:05	17555	0.944784	0	1.04149	0.439336	27.9007	939.494	1.51524	1.51524	100	840	0	0
68	0:04:50	0:00:04	17555	0.944784	0	1.04149	0.439336	27.9007	939.494	1.47536	1.47536	100	840	0	0
69	0:04:55	0:00:05	17555	0.944784	0	1.04149	0.439336	27.9007	939.494	1.44688	1.44688	100	840	0	0
70	0:04:59	0:00:04	17555	0.944784	0	1.04149	0.439336	27.9007	939.494	1.45827	1.45827	100	840	0	0

Figure 93: Summary of the 50th iteration and scenario iterations (51 to 70) for work zone scenario

Figure 94 shows observed versus simulation counts for the entire network before the bridge construction. Figure 96 shows the observed versus simulated counts after adding the Virginia St. work zone and cutting the subarea shown in Figure 95. As it can be seen in Figure 96, the model is not suitable for this application. The R-squared for predication in the subarea is 0.084.

The model was then calibrated again for the subarea only and the observed versus simulated counts were calculated. The result is shown in Figure 97. Though the R-squared is better than the previous case, this is still not an acceptable range.

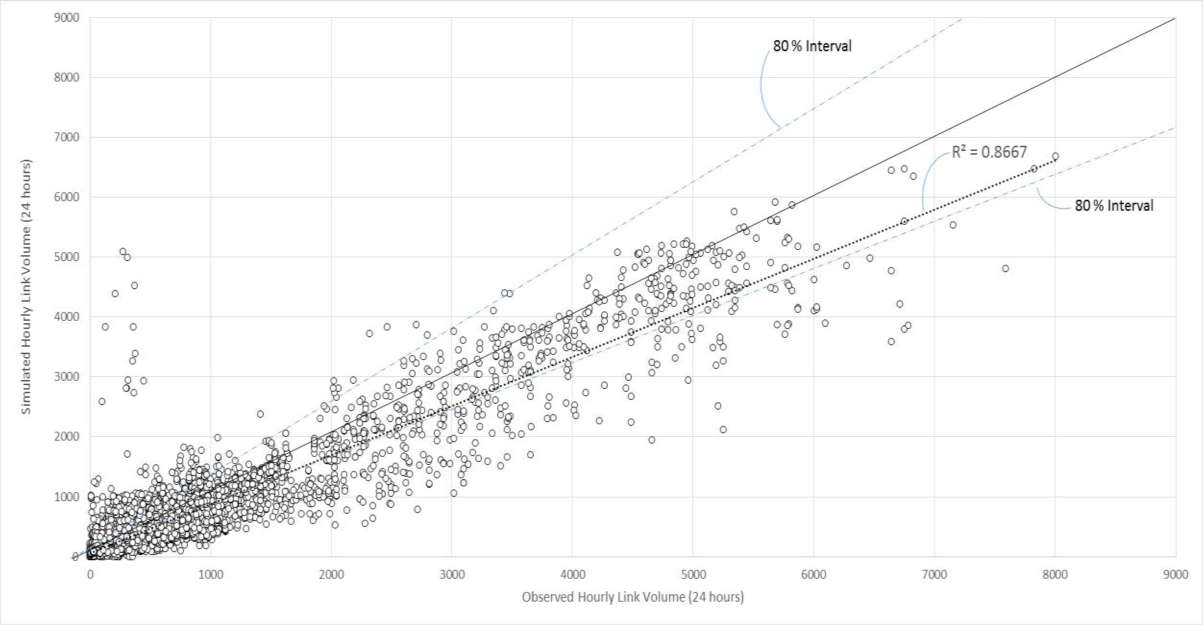


Figure 94: Observed vs simulated counts for whole network before work zone

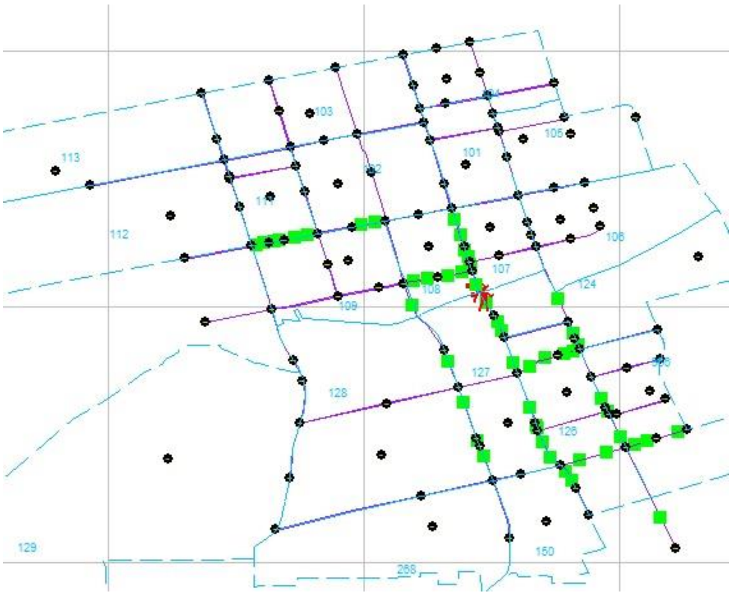


Figure 95: Subarea around bridge construction

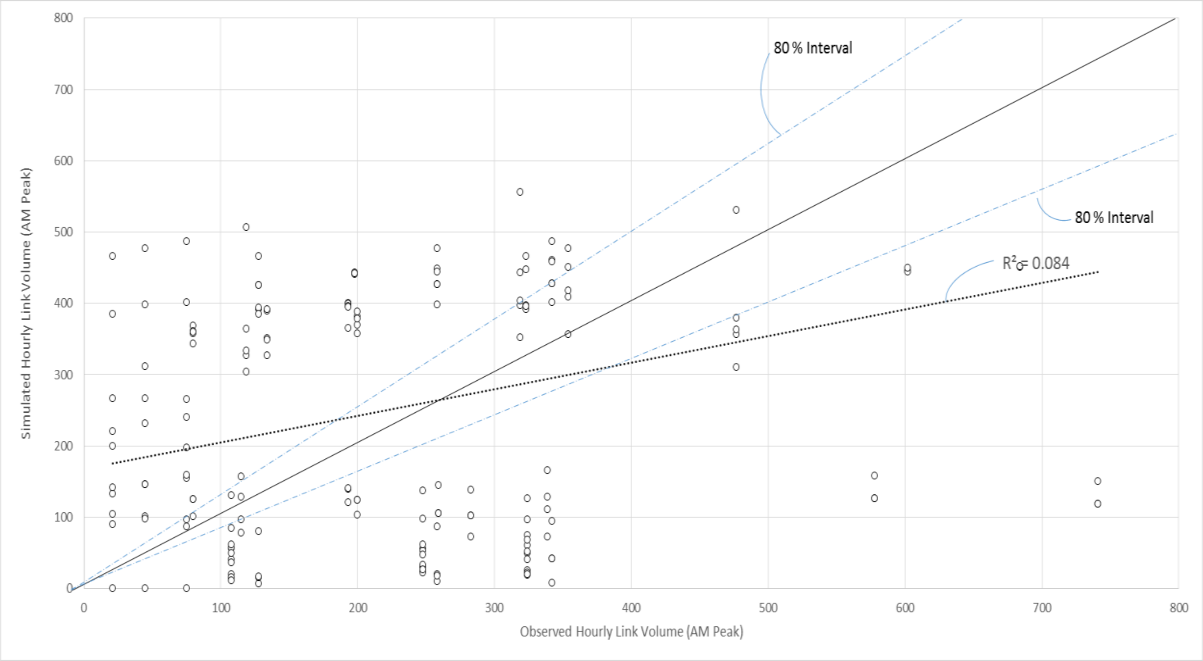


Figure 96: Subarea observed vs simulated counts, calibration on whole network

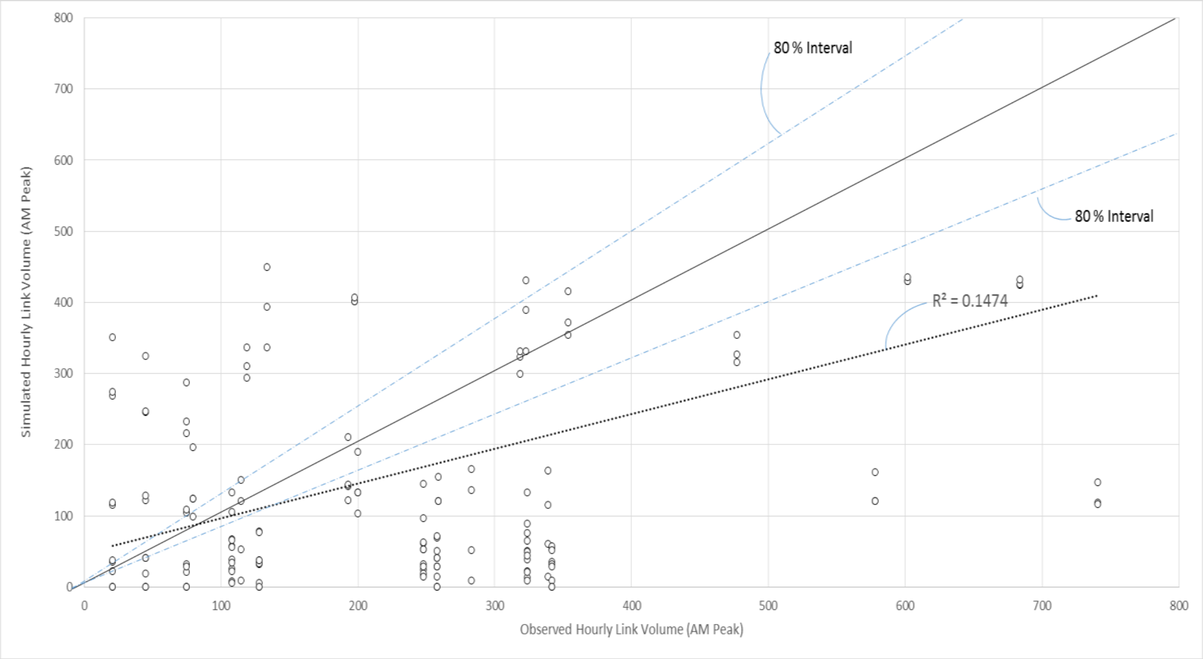


Figure 97: Subarea observed vs simulated counts, calibration on subarea

7. NDOT FUTURE USAGE OF DTA MODEL

Though the Reno/Sparks DTA model was calibrated, similar to STA models, the DTA model also needs maintenance and some consideration for each usage. This section tries to highlight some important topics in this regard.

6.1. Continual System Monitoring and Recalibration

A transportation network is constantly changing in various ways. In the short term, people are combining activities that help them achieve their daily goals. These goals and activities can change from day to day, causing variations in demand patterns. For example, the typical weekday commute to and from a work place can be perturbed by the need to visit a grocery store or drop off or pick up children from after-school programs. In the medium term, people may change their lifestyles, resulting in activity pattern modifications. In the long term, residential location choices and job changes can cause traffic patterns to shift. Land use changes may also cause activity patterns to reorganize in space and time as people try to work their usual activities into a changing landscape of shopping areas, recreational opportunities and employment centers. The above demand fluctuations must be juxtaposed to changes in the physical network itself. Repair and maintenance schedules might warrant that some links be fully or partially closed temporarily. New transportation links might be added, while older ones may be decommissioned. There will undoubtedly be a learning process as travelers, both seasoned and otherwise, adjust to small and large changes. Traffic conditions, therefore, may always be in a state of flux. Before reusing the existing DTA model on a new project, the model user must remember to validate his or her model with new (current) data to ensure that its predictions are sufficiently accurate. When significant deviations are identified, recalibration and revalidation may be necessary to restore the desired accuracy level. This step will require a regular data collection effort supported by expertise in the area of rigorous model calibration and validation. Ideally, if a planning agency realizes and appreciates the value and benefit of a regional DTA model, continual efforts and resources may be planned and committed to regularly keep the DTA model up to date. This update can be put in place in conjunction with regular update of the travel demand model [3].

6.2. Model Consistency Checking

NDOT should continue effort on the calibration of the existing models. Before a basic model can be calibrated or a future model can be used in an analysis, it is an essential component of the modeling to ensure that the model is consistent with the design year network conditions and is free from possible coding mistakes. Errors of omission and transposition of information are common occurrences.

The model should be examined for the properties and attributes of the network to ensure the network is coded correctly, including links, traffic control types of nodes, zone and demand data, and any other fields of information regarding the created network. The travel demand inputs should be validated before model calibration. For example, if time-dependent demand matrices at a smaller interval (e.g., 15-minutes) are disaggregated from existing peak-period O-D matrices, the distribution factor should be reviewed to make sure it closely represents realism.

6.3. Data Collection

As the created DTA models have already incorporated road network characteristics, demand data, and node control types, data collection will mainly put effort on collecting calibration-related data. Required sets of data include: (i) link volume counts, (ii) existing traffic control information, (iii) travel speed and travel times, and (iv) data related to case studies.

The first set of data includes AADT counts and hourly volumes for freeway and major roads which can be obtained from NDOT’s Traffic Record Information Access (TRINA) site, and time-dependent link volume counts of critical locations for a finer granularity (e.g., 15 minutes), which can be obtained from NDOT’s Automatic Traffic Recorder (ATR) stations or by field observation. The second set of data includes intersection signal plans, ramp metering and traveler information plans (e.g., Dynamic Message Signs (DMS)), which can typically be obtained from local transportation agencies. The third set of data includes travel speed, which can be easily accessible from NDOT maintained ITS devices, and travel times along paths and corridors, which can be acquired by using new data sources of Bluetooth. NPMRDS data is also another option for obtaining travel time and speed information. The fourth set of data set depends on targets of the case studies, which could be a work zone analysis, and incident analysis. Location and time of work zone or incident events should be collected to test model reliability.

6.4. DTA Model Calibration

Depending on the calibration objectives, the calibration of a DTA model could be a very complex and time-consuming task. Besides some of the typical measures used in calibration of any transportation model, such as volumes, speeds, travel times, and congestion (length and duration of queuing), other measures including route choice and trip by time of day can be used in the calibration of a DTA model as well. Any changes to any supply or demand parameters could have different effects in these model parameters.

On the demand side, the model should be calibrated for the route choice parameters, such as highway bias and operational costs of vehicles, as those factors could significantly affect the vehicular flow distribution and the resulting AADT pattern. On the supply side, the model should be examined for the capacity representation in the existing DTA models for key bottleneck locations (including lane drop, merge, diverge and weaving and major signalized intersections).

DTALite has a built-in ODME engine that can adjust demand to better match simulated link volumes to observed traffic counts. After an initial ODME run, the model should be checked and validated for several factors, especially demand. For reasons mentioned in previous sections, demand may need to be modified by changing the loading multiplier or time dependent profiles.

6.5. DTA Model Usage

When the model is readily built and calibrated against field data, it can be used for different types of predictive analysis. These analysis cases can include work zone analysis, bottleneck identification study, incident analysis, etc. However, there are some restrictions and considerations that should be carefully understood to obtain the best results from DTA models. Dynamic Traffic Assignment: A Primer [3] has summarized some issues that should be kept in mind when using the DTA model, as follows:

First, DTA models are not the universal cure that can cost-effectively address all types of problems at hand. DTA models take more time and resources to construct and calibrate (as compared with static traffic assignment models) and represent traffic dynamics in a coarser granularity (as compared with microscopic traffic simulation models.) Practitioners are advised to match the choice of modeling approaches to the problem at hand. For long-term planning, for example, the available level of input data required for DTA may not be available, so practitioners may have to make many assumptions in order to construct such models. If the additional detail and precision in their output data compared to conventional network forecasting is not beneficial for the modeling question in mind, then it may not be worth the additional modeling effort. For a smaller bounded area in which a detailed representation of multiple modes (e.g., auto, transit, pedestrians) and facilities (roadways, parking, crosswalks, etc.) are required, microscopic models may be more appropriate and useful.

The DTA model can, to a certain degree, represent the effects (capacity, delay, etc.) of most existing traffic signal control logics (pre-timed, actuated, stop signs, etc.) in their mesoscopic simulation logic. However, this representation is relatively simplistic. DTALite models represent the major features of the different control types without including the exact logic and settings of the existing commercial controllers. If the application at hand requires detailed representation of signal timing and coordination or of intersection or ramp vehicle interactions, DTA models may not be as effective as microscopic simulation models. Furthermore, if the application requires signal-timing optimization, this may imply the simultaneous optimization of routes and controls—generally referred to as the network design problem. DTALite does not directly perform this analysis; additional customization is usually needed, as explained in Section 4.9. The NeXTA GUI is helpful for displaying the results, comparing projects and scenarios, and analyzing the network states. However, clearly illustrating why one project or scenario is better or worse than another is a question of results interpretation that cannot be automated or simplified. This difficulty is of course equally true for static assignment models, though the higher sensitivity of DTA models can bring this issue to a higher level. For example, adding capacity to the network may negatively affect some regions of the network or even the entire system. This occurrence is known as the Braess paradox and can easily be explained in the context of static traffic assignment but, because the effects may be spread out over the entire system and over several time periods, there may not be a straightforward way to show where and why the network is negatively affected. At times the analyst may wish to convert quickly from an existing travel forecasting model to a DTA model.

DTALite provides ODME for initial demand calibration. However, manual refinement of the model is always necessary to ensure data set quality. The learning curve, required knowledge, time, and software development also need to be considered when planning DTA model development.

The DTA model deals with large-scale dynamic networks where the network states are the result of many network and demand factors interacting over time and space. Calibration of the model is a critical step that requires knowledge of the DTA model and actual traffic conditions at the site of interest.

8. PROPOSING A DYNAMIC PUBLIC TRANSIT DEMAND ASSIGNMENT FOR RENO/SPARKS NETWORK

The number of bus transit passengers that are attracted to a route is related to different levels of transit design including line configuration and frequency. One of the important factors of lines are terminals. Location of terminals is also related to line routing that is another level of transit system design. The output terminal locations, line routes, and frequencies affect different layers of related parties, such as users, operators and society. Therefore, bus transit demand assignment (BTDA) can be considered as a multilevel and multilayer problem. In this chapter, the link pheromone-updating concept of Ant Colony Optimization (ACO) was used also for nodes to weight appropriate nodes during the ACO process. While traditional ACO links have pheromone, nodes also receive pheromone based on the total system cost in the proposed method. The total system cost includes user, operator and society costs. Node pheromones gradually increase at good node candidates and decrease at poor locations. At each iteration of ACO, routing and frequency calculation is performed in addition to terminal selection so the effect of these two levels of transit system design is considered at the same time on the assignment process. Thus, with a small modification in the ACO process, the model performs the BTDA as a multilevel and multilayer problem without increasing the complexity of the design. The proposed method here needs to be customized for the Reno/Sparks network. All unit costs are now assumed. A sample network has been solved for this proposed method.

8.1. Introduction

The public transport design has been categorized into four levels including network, frequency, timetable, and crew design. The goal of network design is to plan or select various lines to cover the demand of public transportation.

Though literature is full of studies related to facility location problems, few have studied the bus terminal problem as a public transportation problem and these few research have mostly been a facility location problem. Since it is also an NP-hard problem, it has been studied as a classical computer science problem without thoroughly addressing the issues of a transit network design. Therefore, the research about bus terminal problems also can be considered as a facility design problem since transit related parameters such as different level of public transportation design and different layers of affected parties such as users, operator and society (non-users) are not studied in them. On the other hand, the vast number of network design research usually use simplified methods to determine the beginning and end of lines. For example, one common method is to use terminals based on demand (6).

Assigning transit demand to links also has been widely studied. To the best of our knowledge, these assignment methods are not based on different transit design levels and usually consider only user perspective.

The objective of this chapter is to propose a method to assign bus transit demand by designing bus network terminals simultaneously with routing and considering costs of users, operators and society. After designing terminals and routes, frequency of lines are calculated and then the passengers attracted to each line will be added to links. By adding up all passengers passing from each link, the total demand of each link can be calculated.

The solution method used in this chapter is inspired by the ACO process. Though only the links obtain pheromone at each iteration based on the objective function of the problem, nodes also receive pheromone based on their effectiveness to be as a terminal. In addition to combining routing and terminal selection, different layers (users, operators and society) are considered for assignment process.

8.2. Methodology

The bus network designed here has three levels including terminal, route and frequency design. All levels have a mutual effect on each other. For example, terminals affect routes, routes affect frequency and frequency affects terminal selection process. The methodology proposed here used ACO for terminal and routing design, and frequency was determined analytically. The methodology can be summarized as follows: in ACO, two nodes were selected based on the amount of nodes' pheromones at first and last stops of one line, then routing was performed based on the ACO algorithm. Using the demand covered by each line, frequency was then calculated. This process was repeated until lines covered all demand. This steps forms one iteration in the ACO algorithm. At the end of each iteration, the objective function was calculated and pheromones of nodes and links were updated based on the objective function value. Iterations continued until the termination criteria were satisfied. Figure 98 shows this process.

Ant colony optimization is a metaheuristic that finds the close-to-optimum solution by imitating the behavior of ants in finding the shortest path toward food. Each ant selects the best path by the amount of pheromones that is laid by other ants. The shorter path has more probability of higher pheromone levels because of the shorter time that each ant traverses it. Gradually, the shortest path toward food piles up more pheromone and after a while, all ants use only this path.

In network design, ants represent an agent for making a route, line, or a set of lines. An ant starts its journey from one terminal and at each node, based on the pheromone of each link, picks one link and continues toward the next node. Though the link with most pheromone has the higher chance to be chosen, similar to reality, an ant does not always choose the link with the most pheromone. Each link has a probability of being chosen based on its current amount of pheromone. This probability can be calculated as follows:

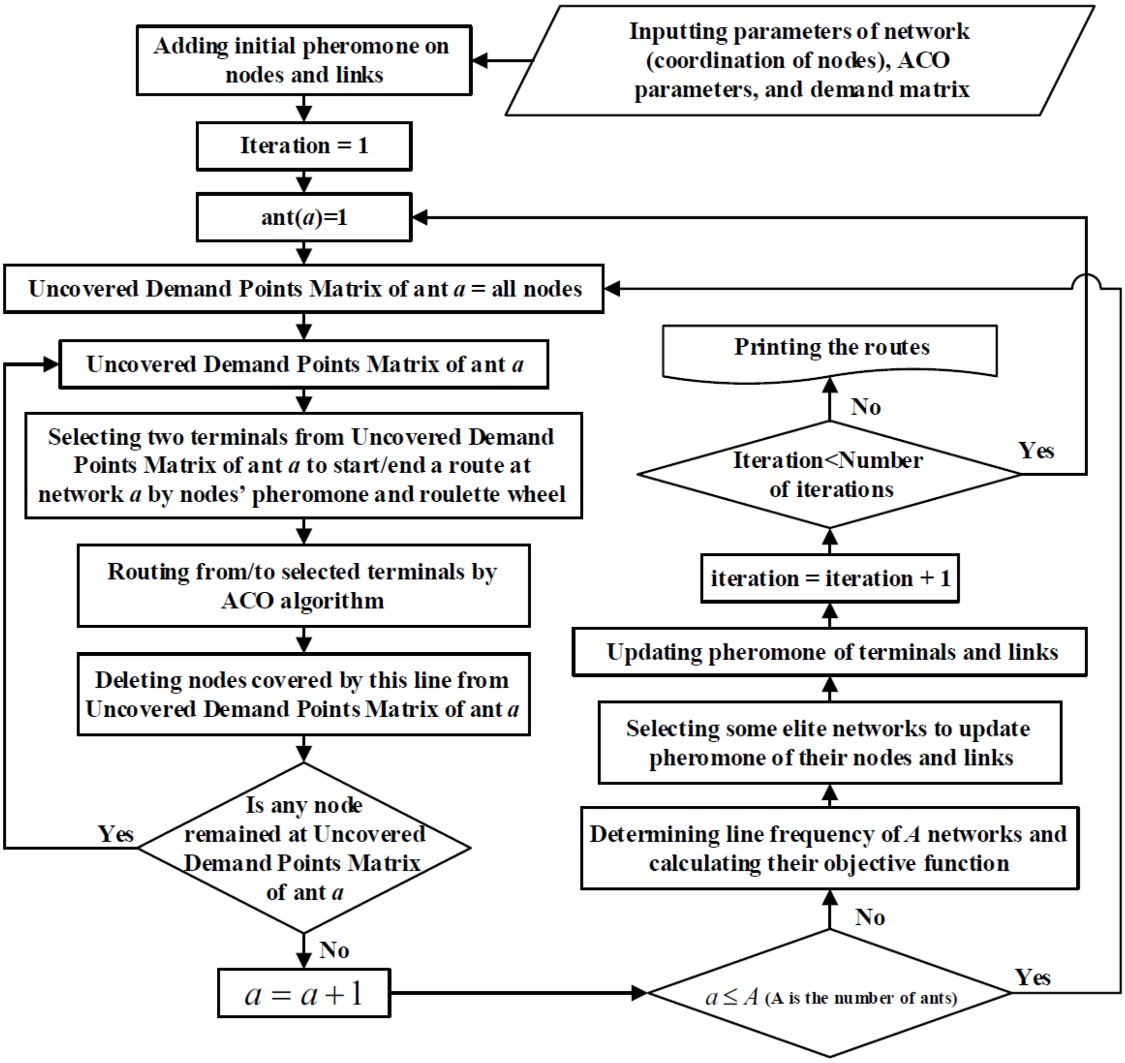

Adding initial pheromone on nodes and links
Inputting parameters of network (coordination of nodes), ACO parameters, and demand matrix
Iteration = 1
ant(a)=1
Uncovered Demand Points Matrix of ant a = all nodes
Uncovered Demand Points Matrix of ant a
Selecting two terminals from Uncovered Demand Points Matrix of ant a to start/end a route at network a by nodes' pheromone and roulette wheel
Routing from/to selected terminals by ACO algorithm
Deleting nodes covered by this line from Uncovered Demand Points Matrix of ant a
Is any node remained at Uncovered Demand Points Matrix of ant a
Yes
No
a = a + 1
a ≤ A (A is the number of ants)
Yes
No
Determining line frequency of A networks and calculating their objective function
Selecting some elite networks to update pheromone of their nodes and links
Updating pheromone of terminals and links
iteration = iteration + 1
Iteration<Number of iterations
Yes
No
Printing the routes

Figure 98: Methodology flowchart of PTDA

$$P_{ij}^{a} = \frac{\tau_{ij}^{\beta}}{\sum_{s \in \omega} \tau_{is}^{\beta}}$$

Equation 1

where:

P_{ij}^{k} : Probability of choosing node *j* from node *i* by ant *a*

ω : The nodes which their demand is not covered by ant *a*

β : Parameter to regulate the influence of the pheromone of link *ij*. This value should be defined based on designer’s experience

τ_{ij} : Amount of pheromone on link *ij*

From equation 1, the probability of each link can be calculated. The next link will be selected based on the roulette wheel method. In this method, first the cumulative probability of each link will be calculated and then one random number will be generated. The point that random number shows determines which link should be selected. For instance, if the probabilities of three possible links are 0.1, 0.7, and 0.2, then the cumulative probabilities of these three links are 0.1, 0.8, and 1. Now, if the random number is 0.83, then the third link will be selected for the route. Note that here the second link has the most chance to be selected, still based on the random number, other links might be chosen. The process of selecting the links continues until the ant reaches the second terminal. One issue that happens during routing is unnecessary loops. In this proposed method, loops were removed from line nodes after completing the process of routing of the line. For instance, the node matrix of the line shown in Figure 99 is {1, 4, 7, 8, 5, 6, 9, 8, 11, 12}. Since node 8 is added to this matrix twice, it means there is a loop. Removing the nodes between repeated numbers eliminates the loop. At this example, after removing the loop, the node matrix becomes {1, 4, 7, 8, 11, 12}. However, some bus transit agencies design their lines with loops to cover more nodes. Though this is not favorable for most of passengers, it increases the line coverage. Even when the loop deleting function is not used, due to unnecessary riding costs that are imposed to some passengers, most of the lines with loops do not have a chance to be selected as elite networks for pheromone updating.

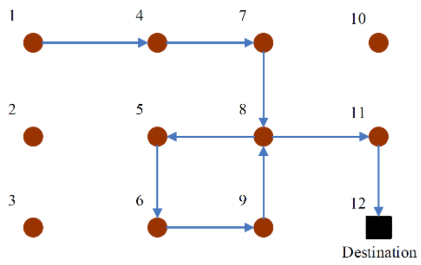


Figure 99: A line with loops

After finishing the first line, all nodes for which the demand was covered by the line will be removed from ω. Before removing the covered demand of each stop along this line, the following equation should be checked:

$$\frac{P_s}{2F_l}\lambda_a + \frac{d_s P_s}{V_{o,l}}\lambda_r \leq \frac{d_w P_s}{V_w}\lambda_w$$

Equation 2

where:

F_l: Required frequency on line *l* (veh/hr)

P_s : Demand of stop *s* on line l (passenger/hr)

d_s : Passengers' travel distance of stop *s* on line l (mile)

$V_{o,l}$: Average operating speed on line l (mile/hr)

d_w : Passengers' walking travel distance from stop *s* to destination (mile)

V_w : Average walking speed (mile/hr)

λ_a : Unit waiting cost of passengers (dollar/passenger-hr)

λ_r : Unit riding cost of passengers (dollar/ passenger-hr)

λ_w: Unit walking cost (dollar/hr)

This equation shows which passengers should be assigned to this line. It means if walking cost is lower for a passenger to reach the destination, it should not be assigned to a line. For example in Figure 99, if the line is as shown (meaning operating agency prefers to have the loop), it is probable that passengers with origin stop 8 prefer to walk if their destination is stop 11. Note that the walking cost is not only based on walking time, but also based on several other factors that should be considered in calculation of λ_w. The first term of Equation 2 refers to waiting time of passengers at stop *s* to receive service. The second term refers to travel time for passengers to travel from stop *s* to their destination. The third term is similar to the second one with the difference that speed and distance is related to walking.

The required frequency of lines in Equation 2 can be calculated by the following equation:

$$F_l = \frac{P_l}{c}$$

Equation 3

P_l : Maximum demand along line *l* (passenger/hr)

c: Capacity of one bus (passenger/veh)

However, the actual frequency of each line is more than this frequency because it is meaningless if the number of buses is fractional. Based on Equation 3, the required fleet size can be calculated as follows:

$$n_l = \left\lceil \frac{2F_l \, l_l}{V_{c,l}} \right\rceil$$

Equation 4

n_l: Fleet size of line *l*

l_l : Length of line l (mile)

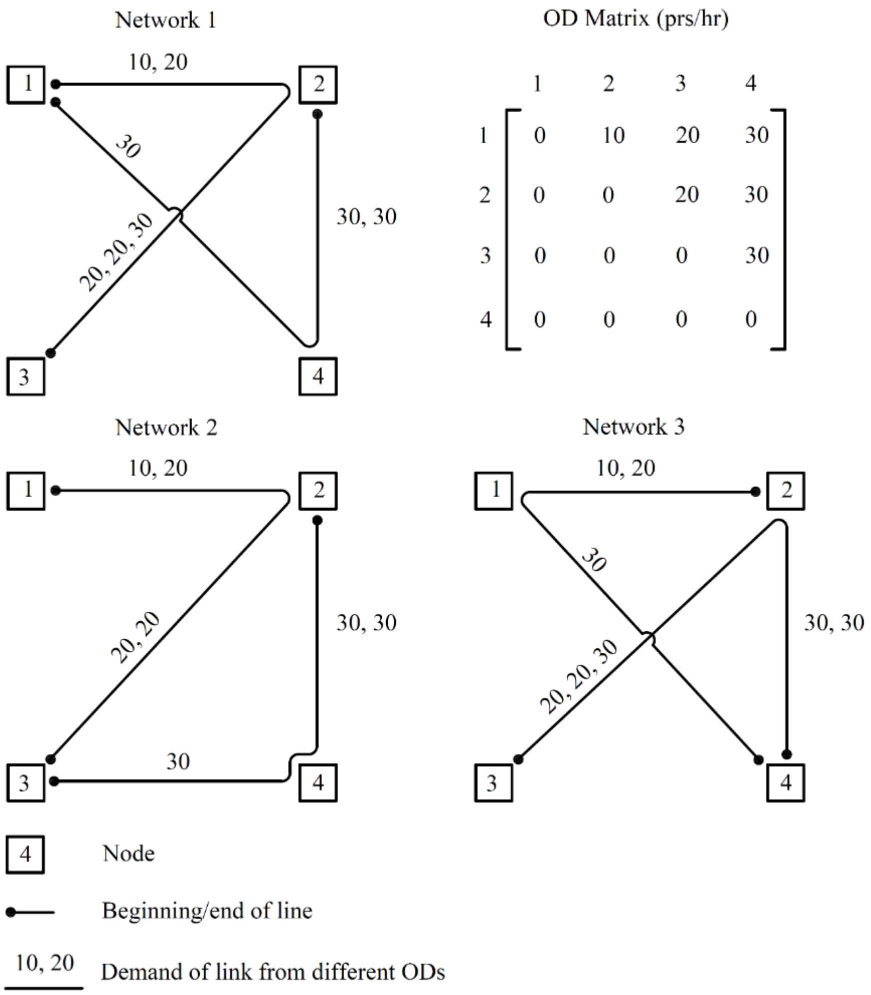
$V_{c,l}$: Average cycle speed on line l (mile/hr)

The half bracket sign of this term means that the value inside of the bracket should be rounded up to an integer number. Since the number of fleet size changes to an integer number, the frequency can be recalculated by the following equation:

$$F_l = \frac{n_l V_{c,l}}{2l_l}$$

Equation 5

After finishing this line, the ant that made it starts routing another line by selecting two terminals. As stated before, terminals are selected based on pheromone of nodes. To show the importance of node pheromone, consider the three networks of Figure 100. Networks 1 and 2 have the same terminal locations (Nodes 1, 2 and 3) and Networks 1 and 3 have the same links covered by lines. If only link pheromones are used for routing, then Networks 1 and 3 leave no distinguishable characteristics for the next iteration, meaning the same links receive pheromone and an ant at the next iteration can make either Network 1 or 3 without considering which network is better in regards of transfers and total system cost. However, if nodes that have been used as a terminal receive pheromone, the network with lower cost receives more pheromone for its terminals and as a result, the next iteration has a higher chance to be used in the process of terminal and routing design. On the other hand, Network 1 and 2 have the same terminals. However, because they receive different pheromone levels for their links, an ant is more probable to follow links of the better network at the next iteration.



OD Matrix (prs/hr)

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 10 & 20 & 30 \\ 2 & 0 & 0 & 20 & 30 \\ 3 & 0 & 0 & 0 & 30 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

Figure 100 Sample networks.

At the beginning, the same amount of pheromone is given to all nodes. For selecting one terminal, the following probability should be calculated:

$$P_n^a = \frac{\tau_n^\mu}{\sum_{i \in \Pi} \tau_i^\mu}$$

Equation 6

where:

P_n^a : Probability of node *n* being chosen by ant *a* as a terminal

μ : Parameter to regulate the influence of the pheromone of node *n*. This value should be defined based on designer’s experience

τ_n : Amount of pheromone on node *n*

Π: Set of eligible nodes to be a terminal

Similar to selecting process of links by one ant, the terminal is also selected based on the cumulative probability and roulette wheel. For any line, two terminals will be chosen using Equation 6.

Equation 2 determines the demand of each line if walking has costs less than riding the line. After routing all required lines, this uncovered demand should be checked to see if any of it can be covered by two or more lines by the following equation:

$$\frac{P_s}{2F_l}\lambda_a + \frac{d_s P_s}{V_{o,l}}\lambda_r + P_s t_c \lambda_c \le \frac{d_w P_s}{V_w}\lambda_w$$

Equation 7

where:

t_c : Number of transfers

λ_c : Unit cost of one transfer (dollar)

The objective function is minimizing the total system cost. The following equation was developed to calculate the total system cost:

$$C = \sum_{l=1}^{L}\left(\frac{P_l}{2F_l}\lambda_a + \frac{d_l P_l}{V_{o,l}}\lambda_r + 2F_l\, l_l\,(\lambda_o + \lambda_s) + \left\lceil \frac{2F_l\, l_l}{V_{c,l}} \right\rceil(\lambda_f + \lambda_m)\right) + \sum_{t=1}^{T}\lambda_t + P_c\lambda_c + \sum_{u=1}^{U} P_u l_u \lambda_w$$

Equation 8

C : Total system cost (dollar/hr)

L : Number of lines

d_l : Average passengers’ travel distance on line l (mile)

$V_{o,l}$: Average operating speed on line l (mile/hr)

T : Number of terminals

P_c : Total number of passengers transferred from one line to another

P_u : Number of uncovered passengers with length of l_u (passengers/hr)

U : Number of uncovered group of passengers

l_u : Trip length of uncovered passengers P_u (mile)

λ_a : Unit waiting cost of passengers (dollar/passenger-hr)

λ_r : Unit riding cost of passengers (dollar/ passenger-hr)

λ_o : Unit operating cost of bus lines (dollar/veh-mile)

λ_s : Unit social cost of system (dollar/veh-mile)

λ_f : Unit fixed cost of each bus (dollar/veh)

λ_m : Unit maintenance cost of bus lines (dollar/veh)

λ_t : Unit cost of terminal t (dollar/hr)

λ_c : Unit cost of one transfer (dollar/hr)

λ_w : Unit walking cost (dollar/hr)

The first summation of Equation 8 calculates different costs associated with lines including the waiting cost of passengers, riding cost of passengers, operating and social costs, and fixed and maintenance costs. Waiting time of passengers is assumed to be half of headway. Riding cost is related to average travel time of passengers. Operating and social costs are calculated based on vehicle-mile. Fixed and maintenance costs are related to number of buses. Again, the half bracket sign of this term means that the value inside of the bracket should be rounded up to an integer number. The parts of maintenance costs that are related to vehicle-mile should be considered in calculation of operating cost.

The second summation of Equation 8 calculates the terminal costs. The terminal unit cost considers all costs related to building and operating of a terminal. At the beginning of network design, a cost estimate should be made for all nodes of network and those that are not suitable to be terminals should be determined.

The last term of equation 8 calculates the cost related to passengers transferring from one line to another. This term brings into account the tradeoff between long lines that cause unnecessary riding costs for some passengers, and shorter lines that cause transfers.

At each iteration, A networks (A is the number of ants) will be made and for each network, Equation 8 calculates the total system cost. This value determines which networks are better in terms of system costs. The elite networks will be selected for pheromone update. Based on each elite network cost, the links and terminals of that network receive pheromone. The amount of added pheromone is calculated by:

$$\Delta \tau_e = \frac{\sigma}{C_e}$$

Equation 9

where:

$\Delta\tau_e$: Value added at each iteration to links and terminals of elite network *e*

σ : A predefined coefficient for adjusting the amount of adding pheromones at each iteration

C_e: Total cost of elite network *e*.

After updating the links and terminals of elite networks, the pheromone of all links and terminals will be multiplied by coefficient θ. This action is called evaporation. The value of θ will be determined based on designer's experience. A number between 0.999 and 0.9999 can be selected for this coefficient. To manage the speed and accuracy of ACO, another coefficient (ϑ) can be used to decrease θ at each iteration. In other words, ϑ is a factor to increase the amount of evaporation at each iteration. All links and eligible nodes will receive an initial pheromone before beginning the first iteration. This initial pheromone makes the probability of selecting the links and nodes equal at beginning iterations. When increasing the iterations, it is better to remove this initial pheromone. The two coefficients let the program have a global search at the beginning iterations and more local search at the final iterations. The coefficient ϑ causes a faster convergence at the final iterations. A number between 0.999 and 0.9999 can be selected for this coefficient. For instance, Figure 101 shows the convergence of result for the same network. At the left diagram, the ϑ is 0.999 while the right diagram has a ϑ equal to 0.9999. The rest of the parameters are the same for both diagrams. As can be seen, the left diagram has reached a solution after around 250 iterations with a cost of 1738 dollar, while the right diagram has reached a solution after 700 iterations with the cost of 1743 dollar.

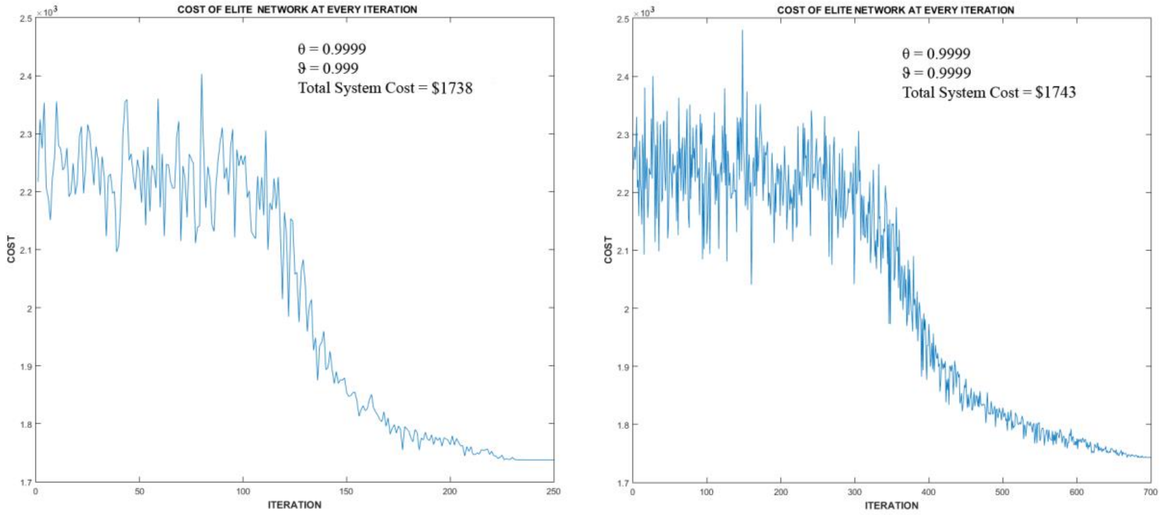


Figure 101: The effect of $\boldsymbol{\theta}$ *and* $\boldsymbol{\vartheta}$ *on the results.*

This process continues until it reaches the termination criteria. The termination criteria can be either a certain number of iterations or a threshold for difference of several consecutive solutions. For example, if solutions do not improve after 10 consecutive iterations, the program ends.

After a bus network is made, two link networks will be made to show the assignment results. One network shows passenger assignment and another one shows bus assignment. For the sample networks of Figure 100, the assignment results are shown in Figure 102. It is assumed that bus capacity is 30 passengers for this example. Frequencies are calculated based on Equation 3. Note that bus assignment is not calculated directly from passenger assignment links. Each line has a maximum load section (MLS) from which frequency is calculated based on this volume and the rest of line has the same frequency of this section.

Though this network and the origin-destination (OD) matrix are very simple, numerous line configurations can be designed to cover the same demand and as a result, the assignments differ significantly. For example, Network 1 and 3 have covered the same links but the volume of buses are different on three links. The methodology proposed in this section tries to provide an assignment that minimizes the costs of users, operators and society.

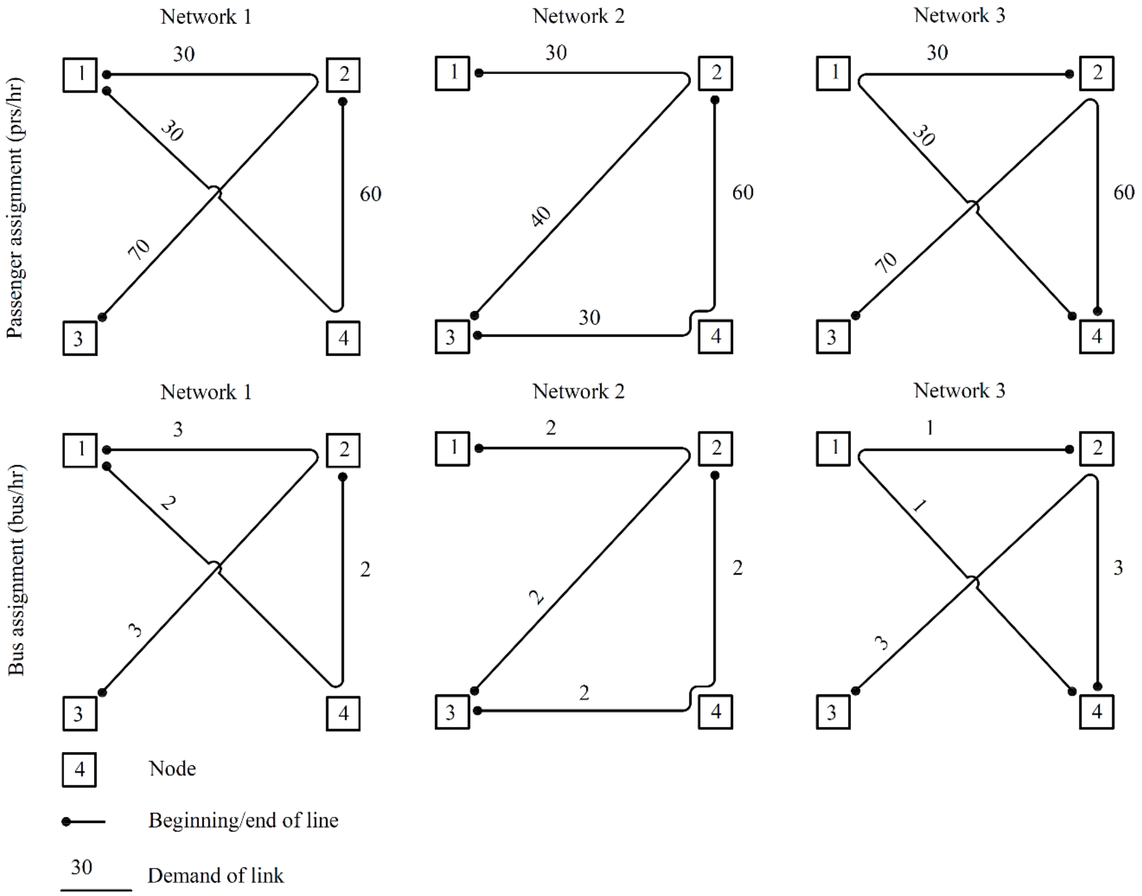


Figure 102: Assignment results for sample networks of Figure 3.

8.3. Numerical Example

In this section, the network of Figure 6 was used for a numerical example. This network has 59 nodes. The demand was selected as a random number from 5 to 100 persons per hour as shown in Table 28. Table 29 shows the unit costs, ACO and bus parameters. Nineteen lines were designed (Figure 103 (a)) by using these parameters. Table 30 shows the summary of these lines. The length of lines range from 1.7 to 6.3 miles and headway is from 1.8 to 12.6 minutes. The average headway of system is 3.2 minutes. This number was weighted based on the fleet size of each line as follows:

$$h_a = \frac{60\ N}{\sum_{l=1}^{L} F_l n_l}$$

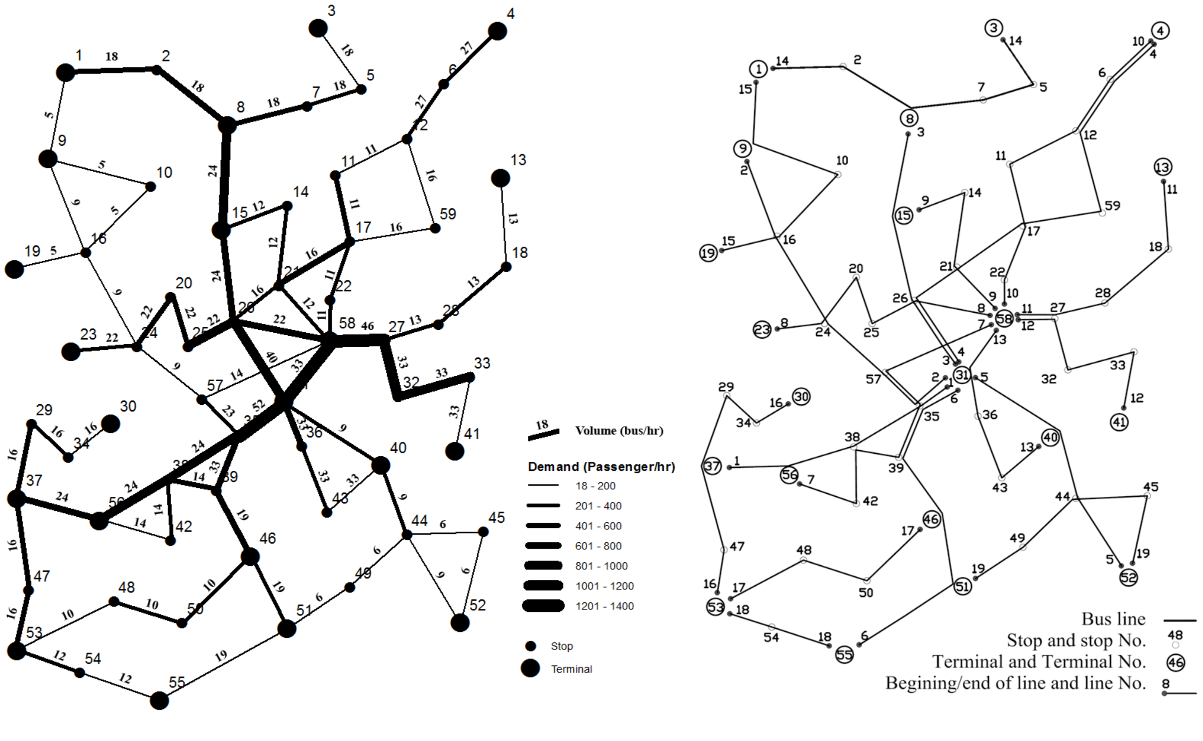
Equation 10

F_l: Actual frequency on line *l* (veh/hr)

h_a: Average system headway (min)

L: Number of lines

N: Total system fleet size



(b) (a)

Figure 103: (a) designed lines; (b) demand assignment

Table 28: Demand Matrix (passenger/hr)

OD	1	4	15	31	33	38	52	53	58
1	0	0	86	86	0	0	0	0	86
2	92	0	92	92	0	0	0	0	0
3	88	0	88	88	0	0	0	0	0
4	0	0	0	68	0	0	0	0	68
5	46	0	46	46	0	0	0	0	0
6	0	16	0	16	0	0	0	0	16
7	56	0	56	56	0	0	0	0	0
8	12	0	12	12	0	0	0	0	0
9	64	0	0	64	0	64	0	0	0
10	14	0	14	14	0	0	0	0	0
11	0	0	94	94	0	0	0	0	94
12	0	63	0	63	0	0	0	0	63
13	0	0	0	32	32	0	0	0	32
14	0	0	84	84	0	0	0	0	84
15	0	0	0	57	0	57	0	0	57
16	11	0	11	11	0	0	0	0	0
17	0	0	0	34	34	0	0	0	34
18	0	0	0	62	62	0	0	0	62
19	24	0	24	24	0	0	0	0	0
20	0	0	60	60	0	0	0	0	60
21	0	0	49	49	0	0	0	0	49
22	0	0	0	10	10	0	0	0	10
23	0	0	78	78	0	0	0	0	78
24	0	0	35	35	0	0	0	0	35
25	0	0	81	81	0	0	0	0	81
26	0	0	33	33	0	0	0	0	33
27	0	0	0	80	80	0	0	0	80
28	0	0	0	31	31	0	0	0	31
29	0	0	0	91	0	91	0	91	0

30	0	0	0	6	0	6	0	6	0
31	0	0	0	0	100	0	0	0	100
32	0	0	0	82	82	0	0	0	82
33	0	0	0	38	0	38	0	0	38
34	0	0	0	87	0	87	0	87	0
35	0	0	0	18	0	18	0	0	18
36	0	0	0	55	0	55	0	0	55
37	0	0	0	70	0	70	0	70	0
38	0	0	0	85	0	0	0	0	0
39	0	0	0	30	0	30	0	0	30
40	0	0	0	95	95	0	0	0	95
41	0	0	0	61	61	0	0	0	61
42	0	0	0	66	0	66	0	0	66
43	0	0	0	89	0	89	0	0	89
44	0	0	0	58	58	0	58	0	0
45	0	0	0	50	50	0	50	0	0
46	0	0	0	97	0	97	0	97	0
47	0	0	0	25	0	25	0	25	0
48	0	0	0	43	0	43	0	43	0
49	0	0	0	9	0	9	9	9	0
50	0	0	0	59	0	59	0	59	0
51	0	0	0	79	0	79	79	0	0
52	0	0	0	8	8	0	0	0	8
53	0	0	0	7	0	7	0	0	0
54	0	0	0	93	0	93	0	93	0
55	0	0	0	17	0	17	0	17	0
56	0	0	0	30	0	30	0	0	30
57	0	0	0	45	0	45	0	0	45
58	0	0	0	70	70	0	0	0	0
59	0	0	0	20	0	20	0	0	20

Table 29: Design Parameters and Summary of Designed Lines.

λ_a	λ_r	$\lambda_o + \lambda_s$	$\lambda_f + \lambda_m$	λ_t	λ_w	λ_c	Π
2	1	3.8	2.5	0*	2.5	0.5	All nodes
β	μ	σ	θ	ϑ	c	V_c	V_o
4	4	100	0.9999	0.999	40	20	20

* It is assumed that beginning/end stops are on street

Table 30: Summary of Designed Lines

Line No.	Line Nodes	Max Demand (prs/hr)	Length (mile)	Frequency (veh/hr)	Actual Frequency (veh/hr)	Headway (min)	Fleet Size (veh)
1	37 56 38 35 31	874	3.4	21.85	23.70	2.5	8
2	31 35 57 24 16 9	309	4.5	7.725	8.80	6.8	4
3	31 26 15 8	858	3.4	21.45	23.58	2.5	8
4	31 26 21 17 59 12 6 4	609	6.3	15.225	15.94	3.8	10
5	31 40 44 52	335	3.4	8.375	8.93	6.7	3
6	31 35 39 46 51 55	684	4.8	17.1	18.94	3.2	9
7	58 57 35 39 38 42 56	479	5.1	11.975	13.80	4.3	7
8	58 26 25 20 24 23	864	3.7	21.6	21.74	2.8	8
9	58 21 14 15	368	2.6	9.2	11.50	5.2	3
10	58 22 17 11 12 6 4	359	4.5	8.975	11.00	5.5	5
11	58 27 28 18 13	375	3.3	9.375	12.12	5.0	4
12	58 27 32 33 41	1235	2.8	30.875	32.14	1.9	9
13	58 31 36 43 40	1246	3.0	31.15	32.89	1.8	10
14	1 2 8 7 5 3	644	4.4	16.1	17.99	3.3	8
15	19 16 10 9 1	113	4.2	2.825	4.77	12.6	2
16	30 34 29 37 47 53	552	3.8	13.8	15.96	3.8	6
17	53 48 50 46	301	3.1	7.525	9.57	6.3	3
18	53 54 55	313	1.7	7.825	11.79	5.1	2
19	51 49 44 45 52	188	3.7	4.7	5.46	11.0	2
	Total	-	71.6	-	-	-	111
	Average		3.8	-	-	3.2	5.84

Figure 103 (b) shows both the passenger and bus demand along lines. This is the demand assigned to links based on the total system cost. Since the total system cost is related to line configurations, terminal locations, and frequency design, the demand shown in this figure was determined after all mentioned steps were done.

It is important to note that the line configurations and their corresponding demand assignment are based on the unit costs mentioned in Table 29. Numerous networks can be designed for the same bus transit demand if any or some of the parameters be changed and as a result, the demand assignment shown in Figure 103 (b) will change. Gholami et al. [34] show how to calculate some of the parameters. However, calculation of these parameters and entering them into the design process is more policy related than pure scientific process. For instance, waiting time was assumed 2 dollar per hour. If the city policy is to encourage more passengers to public transportation, this number should be increased in the design process. Also, this value can be changed city to city because of other factors such as weather conditions. In very cold or very hot cities, the value of waiting time should be higher than cities with mild weather.

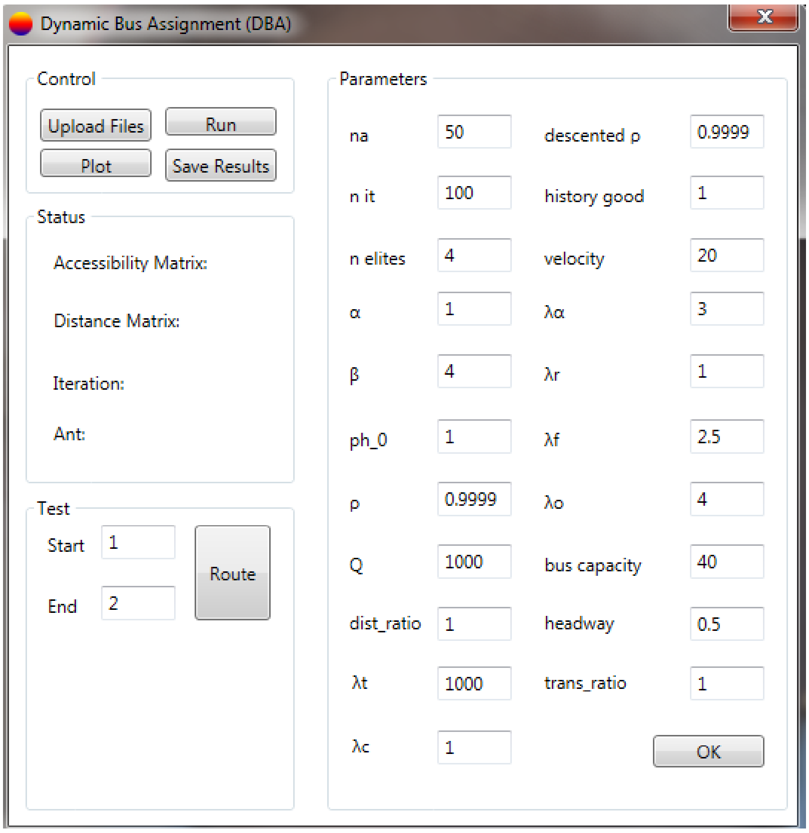
The methodology proposed here was coded into MATLAB and was run several times. Though the nature of this method is stochastic-evolutionary, with choosing appropriate parameters for ACO, the solutions are consistent at different runs.

Since the program considers cost of different layers (users, operators and society), improving one parameter does not necessarily improve the overall system. For example, with decreasing fleet size, the cost of operator reduces but the system become less attractive for passengers, which can eventually lead to less profit for the operating agency. All other parameters have the similar impact on the efficiency of the system. The reason for adding different parameters to an objective function was to consider the tradeoff of all these parameters.

8.4. Summary and Conclusion

Considering different aspects of the bus transit system for BTDA was the incentive of this study. From one perspective, different levels of design, including routing and frequency determination affect, the BTDA. From another perspective, costs and benefits of different affected layers including users, operator, and society can significantly affect the process of BTDA. The objective of this chapter was to take into account these factors. Though the integration of different layers and levels into the design process can usually make it very complex, using ACO and its ability to design different aspects of line configuration simultaneously keeps the model simple. The process of BTDA was integrated into ACO using an innovative method by adding the concept of pheromone to nodes in addition to links. Based on the proposed method, demand assignment was performed. In this method, line configurations and their corresponding demand assignments were based on the designed levels (terminal selection, routing and frequency determination) and affected layers (users, operators, society). Using this model can benefit the overall system without compromising efficiency in favor of simplicity.

If all unit costs are to be calculated for the Reno/Sparks area, the software package developed based on this method (Figure 104) can be used to design an actual Reno/Sparks network.



Start	1		
End	2		
na	50	descented ρ	0.9999
n it	100	history good	1
n elites	4	velocity	20
α	1	λα	3
β	4	λr	1
ph_0	1	λf	2.5
ρ	0.9999	λo	4
Q	1000	bus capacity	40
dist_ratio	1	headway	0.5
λt	1000	trans_ratio	1
λc	1		

Figure 104: Dynamic Bus Assignment (DBA) software package

9. SUMMARY AND CONCLUSIONS

This chapter summarizes the procedure of building the DTA model and the project findings. It also highlights some important considerations regarding usage and application of the DTA model.

9.1. Procedure of Building the DTA Model

To assure an efficient DTA model, the following steps should be checked:

1. Network connectivity.
    a. Activity Location (node, zone)

    The file input_activity_location.csv can check each node ID associated with the zone ID. These nodes represent the activity location.

        i. Node (input_node.csv)
        ii. Zone (input_zone.csv)
            - Each zone has at least one activity location.
            - Zone number used in OD demand table files must be defined in input_zone.csv first.

    b. Link (input_link.csv)

    The fields link type, length, speed, capacity and number of lanes for each link must define.

        i. Checking the link connectivity:
            - How the different link types are connected/related. For example, Freeways connect to arterial using ramp. Problematic links will check in this part.
            - How each link type set up in the system based on capacity, speed and number of lanes

    c. Movement (input_movement.csv)

    To model traffic network in the different scenarios such as work zone, incident, toll, user can prohibit movement from specified nodes. In this case, the user needs to set prohibited_flag to 1 to forbid vehicles to pass from the node.

        i. Prohibited movements
            - Shows whether there is any movement from selected node or not

- Can accept value 0 or 1 which 1 means to forbid vehicles from travelling on

d. Output_summary (Validate travel model

After running the simulation, all the results summarize in the file output_summary. This file includes information on network MOE, link type, number of agents, traffic assignment, ODME, Link MOE.

- Validation of traffic assignment
- Validation of link performance
- Link type statistics

Checking average lane capacity, speed limit and length for each link type.

2. Demand

a. Check the number of vehicles for entire horizon

Are the number of vehicles simulated reasonable?

b. Static traffic assignment: total VMT (Vehicle Miles Traveled), total VHT (Vehicle Hours of Travel)

c. Dynamic traffic assignment

Average travel time and average distance

3. Run simulation for one iteration to check connectivity

Use an extremely low demand. The model is expected to have a free-flow travel time with travel time index = 1.0

4. MSA (method of successive average) to check UE conditions

In principle of User Equilibrium, each user goes on the shortest path. The equilibrium assignment is an iterative technique.

- Setup the field traffic_assignment_method = 0 for MSA in input_scenario_setting
- Reach UE or not?

Check fields Avg User Equilibrium Gap and Relative User Equilibrium Gap in the file output_summary.csv

5. Given sensor data:

a. 40 iterations of MSA plus 1 ODME iteration

b. Have ODME_log check trip generation and trip distribution for each zone.

c. Set up input_scenario_settings.csv

d. Run ODME for x iterations: x = 40 iteration for one hour and at least 3 to 10 global iterations.

e. Expect ODME's adjustment process: Increase or decrease the total demand for major OD pair to match observed link volume

6. ODME_log

The user can check the process of ODME log in files ODME_zone_based_log.csv and file ODME_link_based_log.csv.

7. Spatial dimension

a. Link MOE in GIS layer to check the performance of each link (lane volume, density, speed)

b. Path: Path analysis tool in NeXTA

c. OD: Vehicle analysis tool in NeXTA

8. Time dimension

a. Different departure time interval setting in file input_demand_file_list.csv to distribute demand over time

b. Controlling Time with the Clock Bar feature in NeXTA: The Clock Bar is a toolbar feature located at the top of the screen which allows the user to view time-dependent MOEs by controlling the position of the slider on the toolbar. The toolbar is divided into hours so that the position of the slider refers to the time within a 24-hour modeling time horizon.

9. Aggregate checks

Comparison of simulated to observed traffic network

i. Report matching error deviation per link type, for different demand periods.

ii. Assignment results should be close to the observation for early morning period as no congestion appears with very simple path selection, if we have unexpected bottleneck at early morning, we have an issue.

iii. Check links with major observed volume

iv. Check major OD pairs, vehicle path dialog.

Define volume threshold, >=20 vehicles.

v. Different link type, vehicle type, and departure time.

vi. Check time-dependent path selection

Different departure time periods correspond to different path set selection.

vii. Check different departure time profiles for major OD pairs

viii. Know how to edit the departure time profile in input_demand_file_list.csv. We should not just use the flat demand pattern.

ix. Set distance threshold for each OD pair.

x. Selected link analysis check in all passing OD pairs I vehicle dialogue.

xi. Understand the mapping between OD pair, path flow and link volume

9.2. Capabilities and Benefits of DTA

Given the network characteristics and time-dependent travel demand data–which are typically produced in a TransCAD travel forecasting model—DTA models can be used to estimate dynamic traffic flow patterns over the vehicular network. That is, DTA models load individual vehicles onto the network and assign them on their routes to achieve system-wide objectives.

DTA models provide an easy-accessed graphical user interface (GUI) to display simulation process and statistical results. The assigned vehicle routes can be viewed in the form of animation (either minute-by-minute or second-by-second). In addition, DTA models provide detailed system-level and link-level outputs that describe time-dependent network performance, and the GUI can assist to show these network characteristics and statistics graphically.

The capabilities and benefits of using DTA to analyze traffic network can be concluded as follows:

- DTA provides a mesoscopic traffic analysis tool, providing a connection between regional travel demand forecasting and micro-simulation models.

Travel demand forecasting models such as TransCAD typically provide long-term travel demand forecasting for large-scale networks and micro-simulation tools such as VISSIM and SimTraffic typically focus on animation of individual vehicle movements on small networks. DTA is a connection between these two types of tools, which can both handle the network scale of TransCAD and display animations of individual vehicles running in the network similar to VISSIM.

It is one step further from the planning level of travel forecasting towards the operating detail of micro-simulation, i.e., DTA analyzes large networks as a travel planning tool and provides time-varying traffic network performance (e.g., queue formation, bottleneck identification) but not as much detailed as micro-simulation models.

- Comparing with micro-simulation models, which normally represent known traffic flow patterns, DTA can both represent current traffic performance and evaluate near-term traffic flow impacts from network changes.

Micro-simulation tools are typically designed for traffic performance analysis of the current network of road facilities from an operational perspective. Their primary advantage is recreating real-life scenarios and providing visualized representation of traffic performance.

DTA models can be used for both operational and planning perspectives. It is particularly useful to model a regional level network to forecast traffic flow pattern changes and operational impacts due to incidents such as work zone, special events, and accidents in freeways. However, the case study in this project showed that the DTA model is not efficient for these applications in arterials such as Virginia St.

- NeXTA/DTALite features built-in demand adjustment tools in ODME for initial model calibration.

As demonstrated in Chapters 4 and 5, NeXTA/DTALite features ODME as one of the calibration methods to better match simulated link volumes to observed field data. The demand adjustment tools in ODME are easy to use and the capabilities of changing start times and using alternative route choices for traffic assignment are particularly beneficial in model calibration. This represents the potential for time savings, especially for large networks. However, it should be noted that after running ODME, the model needs further demand adjustments.

9.3. When to Apply DTA models

DTA is a relatively new tool to analyze large-scale route choice alternatives, and can also be a useful tool in the evaluation of a wide variety of transportation conditions [35]. Appropriate applications of DTA models include, but are not limited to:

- Visual animation of vehicles on a large-scale network.

DTA tools can provide a coarse animation of individual vehicles running on the links of a large-scale network over the simulation period. However, the level of simulation is not as detailed as micro-simulation tools. For example, DTA cannot model lane change or intersection operations.

- Visual display of system performance details of a large-scale network.

DTA provides a GUI to display system-level and link-level statistical outputs that describe time-dependent network performance, such as link volume, density, link speed, queue formation, and bottleneck identification. It is an efficient tool to present traffic planners and operators the visualization of these network performance measurements.

- Work zone analysis.

DTA models can be used for analyzing work zone impact on freeway roads. By simulating vehicle routing choices, it can determine where cars would reroute during short-term construction area or long-term road closures for construction.

- Near-term planning project analysis.

DTA models can be used to determine what impact it will have on an entire network or a specified area if a near-term planning project is likely to induce a travel pattern change in time or space among different facilities. Such projects may include:

(a) significant roadway configuration changes (e.g. change streets from one-way to two-way or vice versa),

(b) freeway expansions or road diet,

(c) lane use changes such as adding or converting HOV–HOT lanes,

(d) travel demand management strategies such as peak spreading or congestion pricing,

(e) special event,

(f) incident management response scenarios (e.g., evacuations), and

(g) bottleneck removal studies.

9.4. Requirements for DTA Model Development and Applications

To build and calibrate a fine DTA model to closely present real traffic conditions, there are several requirements and considerations that must be met, including data requirement and modeling efforts.

9.4.1. Data requirements for network development

For network development, basic data requirements include: geometric data, traffic control data, traffic demand, OD demand data and transit demand.

For DTA modeling, estimating the demand is of critical importance. This includes estimating not only the origin and destination patterns, but also the temporal distribution of travel demand. Building a DTA model without this type of information is very difficult. Preferably, the travel demand data is desired to be divided into very short time intervals such as 5 or 15-minute intervals. Typically, a regional travel demand model (e.g. TransCAD model) can provide network characteristics as well as OD travel demand data. Ideally, the travel demand model should distribute daily demand (24-hour assignment) into peak periods or individual hours.

9.4.2. Data requirements for model calibration

For model calibration, quantified measures of effectiveness that can be observed in the field or produced from other model outputs are required. In this project, the initial calibration was conducted by using ODME, which required link volumes data. Hourly volumes were obtained for freeway and major links; therefore, model calibration was limited to a certain extent. The fidelity of a DTA model depends on more than link volumes.

Depending on availability and coverage of observed data sets, model calibration methods vary. In general, the calibration process compares field data to the model outputs, and if the calibration acceptance criteria are satisfied, the model is considered calibrated. Typical types of data for calibration strategies can include:

- Travel times;
- Travel speeds;
- Traffic counts (including temporal peaking, preferably hour-by-hour link volumes);
- Lane utilization;
- Queue information; and

- Transit operations.

However, these types of data are not all required for a single DTA application. When developing a DTA model, modelers should consider the primary area within the network for calibration and what performance measures are most importance, and prepare data collection plans accordingly.

9.4.3. Transportation modeling skills

The application of DTA requires skills of existing transportation modeling techniques. Having fundamental knowledge of travel demand modeling and micro/mesoscopic simulation modeling techniques and working knowledge of model calibration and statistical analysis is needed to apply DTA successfully.

9.4.4. Level of efforts

Based on the choice of DTA software package, network size, data collection needs, and model limits, the level of effort needed in a DTA application can vary significantly. More detailed network profile, travel demand data with shorter time intervals, and more sufficient observed data sets of multiple types and wider coverage are a prerequisite for potential time savings. Otherwise, a large amount of time will be spent on data collection rather than model building and calibration.

9.5. Limitations of DTA Model Applications

Applying DTA methods may require more effort than other static transportation modeling techniques; therefore, the need for DTA methods should be considered carefully, taking into account data needs, model building time, and calibration. DTA models are not the universal solution for all types of traffic problems.

9.5.1. Regarding long-term planning

DTA can be a good tool for present and near-term network performance analysis. For example, it can easily identify active bottlenecks and queue formation of the study network when analyzing the impact of a short-term planning project. However for long-term planning, DTA tools will not be able to adjust model parameters to produce a well-calibrated model because there are no observed field data to calibrate against. The lack of future travel demand data and corresponding field data makes DTA not a suitable tool for long-term planning.

Instead, travel-forecasting models, such as TransCAD, are a better fit for bottlenecks (including upstream and downstream) estimation studies. While DTA can only identify active bottlenecks, travel demand forecasting models can predict all potential bottleneck locations, which is necessary for long-term planning purposes.

9.5.2. Regarding resolution of travel demand data and data collection effort

When detailed input data required for DTA is unavailable, one may have to make many assumptions to build such models. For example, if the travel demand data are provided as an OD matrix describing trips for a day rather than trips in 15-minute intervals, the temporal distribution of travel demand is unclear so that the assigned trips is difficult to replicate real traffic conditions.

The level of precision from DTA models largely depends on data availability. DTA requires a significantly larger amount of data, which may not be readily available in most cases. If the additional effort to collect detailed input data offsets the precision in the DTA output data compared to travel forecasting models, it may not be beneficial to apply DTA and not worth the additional modeling effort. Decision makers need to assess the desired level of precision and the available resources and choose if DTA or conventional travel demand models should be used.

9.5.3. Regarding smaller area analysis

Although DTA tools can model vehicle trajectories and display their movements on links, they are limited in the degree of simulation resolution to model more detailed network performance.

For a smaller network or area, if a careful examination of traffic behavior is needed, micro-simulation is a better tool to continuously model car movements and include a more detailed description of roadway facilities and multiple traffic modes.

9.5.4. Regarding pedestrian simulation

As described in previous sections, DTA is a useful tool to model vehicular network, but it cannot simulate pedestrian behaviors. If pedestrian mode is required for the study, microscopic models may be more appropriate.

9.6. Future Research Focuses

The DTA calibration conducted in this project is considered limited due to data availability and time constraints. Future efforts should be directed to the following aspects.

9.6.1. Model testing with detailed signal timing and turn pockets

Due to some practical problems, in the current DTA model, detailed signal timing and turn pockets are not coded. However, it may be possible in the future to add signal timing or intersection links to increase network fidelity.

9.6.2. Further model calibration

Current model calibration was based on hourly link volumes. Currently, DTALite is not suitable to use link speed data for calibration. In the future, if this feature is modified in DTALite, further calibration could utilize link speed data, which is easily accessible from NDOT maintained ITS devices. Accurate link speed data can be input into the sensor_count.csv file.

9.6.3. Model Applications through Case Studies

During the completion of this project, the only case study that was available was the Virginia St. Bridge construction. In the future, some other case studies, especially those involving freeways, should be conducted to demonstrate the applicability of DTA models. Very limited literature was found to document such case studies.

9.6.4. Bus transit demand assignment

A method was proposed in this study for bus transit demand assignment. However, the model was tested only on a sample network with assumed unit costs. All unit costs should be calculated for the Reno/Sparks area.

9.7. Project Achievements

The DTA model development was completed for the study network of the Reno/Sparks area. It successfully demonstrated the conversion from an established TransCAD regional travel demand model into NeXTA/DTALite for mesoscopic analysis.

Basic model calibration and validation against hourly link volume data was successfully conducted and a reasonably improved result has been achieved. It demonstrated the capability of NeXTA/DTALite software package to better match link volume to observed data by adjusting OD matrices.

Although the calibration process improved the initial DTA model to a certain extent, it was considered a limited one because the field data obtained for case study was not sufficient enough to perform more studies on different applications of DTA. It is highly recommended to conduct more case studies if other projects arise in the Reno/Sparks area.

REFERENCES

1. Tian, Z., Lin, D., Yin, K., Zhou, X. (2014). Development of a Dynamic Traffic Assignment Model for Northern Nevada. UTC-NDOT Research Program.

2. Lin, D., Tian, Z. (2015). Linking Travel Demand Modeling with Micro-Simulation. Prepared for: Regional Transportation Commission of Southern Nevada.

3. Dynamic Traffic Assignment: A Premier (2011). TRB Transportation Network Modeling Committee. Link: http://onlinepubs.trb.org/onlinepubs/circulars/ec153.pdf.

4. Beckmann, M. J., C. B. McGuire, and C. B. Winsten. (1956). Studies in the Economics of Transportation. Yale University Press.

5. Sbayti, H. and Roden, D. (2010).Best Practices in the Use of Micro Simulation Models. NCHRP 8-36 Task 90.

6. Peeta, S. and Ziliaskopoulos, A. (2001). Fundamentals of Dynamic Traffic Assignment: the Past, the Present and the Future. Networks and Spatial Economics 1(2), 201–230.

7. Merchant, D. K., & Nemhauser, G. L. (1978). A model and an algorithm for the dynamic traffic assignment problems. Transportation science, 12(3), 183-199.

8. Merchant, D. K., & Nemhauser, G. L. (1978). Optimality conditions for a dynamic traffic assignment model. Transportation Science, 12(3), 200-207.

9. Carey, M. (1992). Nonconvexity of the dynamic traffic assignment problem. Transportation Research Part B: Methodological, 26(2), 127-133.

10. Janson, B. N. (1991). Dynamic traffic assignment for urban road networks. Transportation Research Part B: Methodological, 25(2), 143-161.

11. Ziliaskopoulos, A. K. (2000). A linear programming model for the single destination system optimum dynamic traffic assignment problem. Transportation science, 34(1), 37-49.

12. Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological, 28(4): 269-287.

13. Daganzo, C. F. (1995). The cell transmission model, part II: Network traffic. Transportation Research Part B: Methodological 29(2): 79-93.

14. Jayakrishnan, R., Tsai, W. K., & Chen, A. (1995). A dynamic traffic assignment model with traffic-flow relationships. Transportation Research Part C: Emerging Technologies, 3(1), 51-72.

15. Carey, M. (1986). A constraint qualification for a dynamic traffic assignment model. Transportation Science, 20(1), 55-58.

16. Smith, M. J. (1993). A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. Transportation Research Part B: Methodological, 27(1), 49-63.

17. Lo, H. K. (1999). A dynamic traffic assignment formulation that encapsulates the cell-transmission model. In 14th International Symposium on Transportation and Traffic Theory.

18. Lam, W. H., & Huang, H. J. (1995). Dynamic user optimal traffic assignment model for many to one travel demand. Transportation Research Part B: Methodological, 29(4), 243-259.

19. Kerner, B. S. (2017). Breakdown minimization principle versus Wardrop’s equilibria for dynamic traffic assignment and control in traffic and transportation networks: A critical mini-review. Physica A: Statistical Mechanics and its Applications, 466, 626-662.

20. DynusT. (2012). DynusT© Online User's Manual. Website link: http://dynust.net/wikibin/doku.php (Accessed on 07/09/2013).

21. Chiu, Y.-C., Zhou, L., and Song, H. (2010). Development and calibration of the Anisotropic Mesoscopic Simulation model for uninterrupted flow facilities. Transportation Research Part B: Methodological, 44(1), 152-174.

22. DTALite, Project Descriptions. Website link: https://sites.google.com/site/dtalite/

23. Newell, G. F. (1993). A simplified theory of kinematic waves in highway traffic, I. General theory, II. Queuingat freeway bottlenecks, III. Multidestination flows. Transportation Research Part B: Methodological 27, 281-287, 289-303, 305-313.

24. NeXTA: Network eXplorer for Traffic Analysis. Website link: https://code.google.com/p/nexta/.

25. TransModeler website: http://www.caliper.com/transmodeler/

26. Horowitz, A. Statewide Travel Forecasting Models: A Synthesis of Highway Practice, HCHRP Synthesis 358, Washington D.C.2006

27. Balakrishna, R., Morgan, D., Yang, Q., and Slavin, H. (2012). Comparison of Simulation-Based Dynamic Traffic Assignment Approaches for Planning and Operations Management. 4th Transportation Research Board Conference on Innovations in Travel Modeling

28. Zlatkovic, M., & Zhou, X. (2015). Integration of signal timing estimation model and dynamic traffic assignment in feedback loops: system design and case study. Journal of Advanced Transportation, 49(6), 683-699.

29. Li, P., Mirchandani, P., & Zhou, X. (2015). Hierarchical Multiresolution Traffic Simulator for Metropolitan Areas: Architecture, Challenges, and Solutions. Transportation Research Record: Journal of the Transportation Research Board, (2497), 63-72.

30. Lu, C. C., Zhou, X., & Zhang, K. (2013). Dynamic origin–destination demand flow estimation under congested traffic conditions. Transportation Research Part C: Emerging Technologies, 34, 16-37.

31. Ziliaskopoulos, A. K., & Mahmassani, H. S. (1993). Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. Transportation research record, 94-94.

32. Lu, C. C., Mahmassani, H. S., & Zhou, X. (2009). Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. Transportation Research Part B: Methodological, 43(3), 345-364.

33. Box, G. E. P., and N. R. Draper. Empirical Model-Building and Response Surfaces. Wiley and Sons, New York, 1987.

34. Gholami, A., Y. Taghizadeh, Z. Tian (2014). Classification of taxi khattee (jitney) lines based on topography and line cost indices. Transportation Research Part A: Policy and Practice Volume 59, 2014, pp. 239–249.

35. Melson, C. L., Boyles, S. D., and Machemehl, R. B. (2013). Modeling the Traffic Impacts of Transit Facilities Using Dynamic Traffic Assignment. In Transportation Research Board 92nd Annual Meeting (No. 13-2267).

Nevada Department of Transportation
Rudy Malfabon, P.E. Director
Ken Chambers, Research Division Chief
(775) 888-7220
kchambers@dot.nv.gov
1263 South Stewart Street
Carson City, Nevada 89712