Dynamic, Integrated Model System: Sacramento-Area Application, Volume 2: Network Report

DETAILS

61 pages | 8.5 x 11 | PAPERBACK
ISBN 978-0-309-43351-8 | DOI 10.17226/22369

AUTHORS

Cambridge Systematics, Inc.

The Second
STRATEGIC HIGHWAY RESEARCH PROGRAM

SHRP 2 REPORT S2-C10B-RW-2

Dynamic, Integrated Model System: Sacramento-Area Application

Volume 2: Network Report

CAMBRIDGE SYSTEMATICS, INC.
in association with
SACRAMENTO AREA COUNCIL OF GOVERNMENTS
UNIVERSITY OF ARIZONA
UNIVERSITY OF ILLINOIS, CHICAGO
SONOMA TECHNOLOGY, INC.
FEHR AND PEERS

TRANSPORTATION RESEARCH BOARD
WASHINGTON, D.C.
2014
www.TRB.org

Subject Areas

Environment
Highways
Planning and Forecasting

The Second Strategic Highway Research Program

America's highway system is critical to meeting the mobility and economic needs of local communities, regions, and the nation. Developments in research and technology—such as advanced materials, communications technology, new data collection technologies, and human factors science—offer a new opportunity to improve the safety and reliability of this important national resource. Breakthrough resolution of significant transportation problems, however, requires concentrated resources over a short time frame. Reflecting this need, the second Strategic Highway Research Program (SHRP 2) has an intense, large-scale focus, integrates multiple fields of research and technology, and is fundamentally different from the broad, mission-oriented, discipline-based research programs that have been the mainstay of the highway research industry for half a century.

The need for SHRP 2 was identified in *TRB Special Report 260: Strategic Highway Research: Saving Lives, Reducing Congestion, Improving Quality of Life,* published in 2001 and based on a study sponsored by Congress through the Transportation Equity Act for the 21st Century (TEA-21). SHRP 2, modeled after the first Strategic Highway Research Program, is a focused, time-constrained, management-driven program designed to complement existing highway research programs. SHRP 2 focuses on applied research in four areas: Safety, to prevent or reduce the severity of highway crashes by understanding driver behavior; Renewal, to address the aging infrastructure through rapid design and construction methods that cause minimal disruptions and produce lasting facilities; Reliability, to reduce congestion through incident reduction, management, response, and mitigation; and Capacity, to integrate mobility, economic, environmental, and community needs in the planning and designing of new transportation capacity.

SHRP 2 was authorized in August 2005 as part of the Safe, Accountable, Flexible, Efficient Transportation Equity Act: A Legacy for Users (SAFETEA-LU). The program is managed by the Transportation Research Board (TRB) on behalf of the National Research Council (NRC). SHRP 2 is conducted under a memorandum of understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Federal Highway Administration (FHWA), and the National Academy of Sciences, parent organization of TRB and NRC. The program provides for competitive, merit-based selection of research contractors; independent research project oversight; and dissemination of research results.

SHRP 2 Report S2-C10B-RW-2

ISBN: 978-0-309-27398-5

Copyright Information

Authors herein are responsible for the authenticity of their materials and for obtaining written permissions from publishers or persons who own the copyright to any previously published or copyrighted material used herein.

The second Strategic Highway Research Program grants permission to reproduce material in this publication for classroom and not-for-profit purposes. Permission is given with the understanding that none of the material will be used to imply TRB, AASHTO, or FHWA endorsement of a particular product, method, or practice. It is expected that those reproducing material in this document for educational and not-for-profit purposes will give appropriate acknowledgment of the source of any reprinted or reproduced material. For other uses of the material, request permission from SHRP 2.

Note: SHRP 2 report numbers convey the program, focus area, project number, and publication format. Report numbers ending in "w" are published as web documents only.

Notice

The project that is the subject of this report was a part of the second Strategic Highway Research Program, conducted by the Transportation Research Board with the approval of the Governing Board of the National Research Council.

The members of the technical committee selected to monitor this project and to review this report were chosen for their special competencies and with regard for appropriate balance. The report was reviewed by the technical committee and accepted for publication according to procedures established and overseen by the Transportation Research Board and approved by the Governing Board of the National Research Council.

The opinions and conclusions expressed or implied in this report are those of the researchers who performed the research and are not necessarily those of the Transportation Research Board, the National Research Council, or the program sponsors.

The Transportation Research Board of the National Academies, the National Research Council, and the sponsors of the second Strategic Highway Research Program do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of the report.

SHRP 2 Reports

Available by subscription and through the TRB online bookstore:
www.TRB.org/bookstore

Contact the TRB Business Office:
202-334-3213

More information about SHRP 2:
www.TRB.org/SHRP2

THE NATIONAL ACADEMIES

Advisers to the Nation on Science, Engineering, and Medicine

The **National Academy of Sciences** is a private, nonprofit, self-perpetuating society of distinguished scholars engaged in scientific and engineering research, dedicated to the furtherance of science and technology and to their use for the general welfare. On the authority of the charter granted to it by Congress in 1863, the Academy has a mandate that requires it to advise the federal government on scientific and technical matters. Dr. Ralph J. Cicerone is president of the National Academy of Sciences.

The **National Academy of Engineering** was established in 1964, under the charter of the National Academy of Sciences, as a parallel organization of outstanding engineers. It is autonomous in its administration and in the selection of its members, sharing with the National Academy of Sciences the responsibility for advising the federal government. The National Academy of Engineering also sponsors engineering programs aimed at meeting national needs, encourages education and research, and recognizes the superior achievements of engineers. Dr. C. D. (Dan) Mote, Jr., is president of the National Academy of Engineering.

The **Institute of Medicine** was established in 1970 by the National Academy of Sciences to secure the services of eminent members of appropriate professions in the examination of policy matters pertaining to the health of the public. The Institute acts under the responsibility given to the National Academy of Sciences by its congressional charter to be an adviser to the federal government and, on its own initiative, to identify issues of medical care, research, and education. Dr. Victor J. Dzau is president of the Institute of Medicine.

The **National Research Council** was organized by the National Academy of Sciences in 1916 to associate the broad community of science and technology with the Academy's purposes of furthering knowledge and advising the federal government. Functioning in accordance with general policies determined by the Academy, the Council has become the principal operating agency of both the National Academy of Sciences and the National Academy of Engineering in providing services to the government, the public, and the scientific and engineering communities. The Council is administered jointly by both Academies and the Institute of Medicine. Dr. Ralph J. Cicerone and Dr. C. D. (Dan) Mote, Jr., are chair and vice chair, respectively, of the National Research Council.

The **Transportation Research Board** is one of six major divisions of the National Research Council. The mission of the Transportation Research Board is to provide leadership in transportation innovation and progress through research and information exchange, conducted within a setting that is objective, interdisciplinary, and multimodal. The Board's varied activities annually engage about 7,000 engineers, scientists, and other transportation researchers and practitioners from the public and private sectors and academia, all of whom contribute their expertise in the public interest. The program is supported by state transportation departments, federal agencies including the component administrations of the U.S. Department of Transportation, and other organizations and individuals interested in the development of transportation. **www.TRB.org**

www.national-academies.org

SHRP 2 STAFF

Ann M. Brach, *Director*
Stephen J. Andrle, *Deputy Director*
Neil J. Pedersen, *Deputy Director, Implementation and Communications*
Cynthia Allen, *Editor*
Kenneth Campbell, *Chief Program Officer, Safety*
JoAnn Coleman, *Senior Program Assistant, Capacity and Reliability*
Eduardo Cusicanqui, *Financial Officer*
Richard Deering, *Special Consultant, Safety Data Phase 1 Planning*
Shantia Douglas, *Senior Financial Assistant*
Charles Fay, *Senior Program Officer, Safety*
Carol Ford, *Senior Program Assistant, Renewal and Safety*
Jo Allen Gause, *Senior Program Officer, Capacity*
James Hedlund, *Special Consultant, Safety Coordination*
Alyssa Hernandez, *Reports Coordinator*
Ralph Hessian, *Special Consultant, Capacity and Reliability*
Andy Horosko, *Special Consultant, Safety Field Data Collection*
William Hyman, *Senior Program Officer, Reliability*
Linda Mason, *Communications Officer*
Reena Mathews, *Senior Program Officer, Capacity and Reliability*
Matthew Miller, *Program Officer, Capacity and Reliability*
Michael Miller, *Senior Program Assistant, Capacity and Reliability*
David Plazak, *Senior Program Officer, Capacity and Reliability*
Rachel Taylor, *Senior Editorial Assistant*
Dean Trackman, *Managing Editor*
Connie Woldu, *Administrative Coordinator*

ACKNOWLEDGMENTS

This work was sponsored by the Federal Highway Administration in cooperation with the American Association of State Highway and Transportation Officials. It was conducted in the second Strategic Highway Research Program (SHRP 2), which is administered by the Transportation Research Board of the National Academies. This project was managed by Stephen J. Andrle, Deputy Director, SHRP 2.

The primary authors of this report are Drs. Yi-Chang Chiu, Alireza Khani, Hyunsoo Noh, and Brenda Bustillos of the University of Arizona and Dr. Mark Hickman, currently of the University of Queensland, who performed this work while with the University of Arizona. Key contributions were made by Michalis Xyntarakis of Cambridge Systematics, Inc. The principal investigator for SHRP 2 Project C10B is Thomas Rossi of Cambridge Systematics.

Copyright National Academy of Sciences. All rights reserved.

FOREWORD

Stephen J. Andrle, *SHRP 2 Deputy Director*

This report will be of interest to professionals who use travel demand and network assignment models as part of the transportation planning process. The goal of this research was to improve urban-scale modeling and network procedures to address operations or spot improvements that affect travel-time choice, route choice, mode choice, reliability, or emissions. Such improvements may include traveler information, expanded transit service, pricing, reversible lanes, or improved bottlenecks. Operational improvements like these are difficult to model on an urban area scale using existing tools. A secondary goal was to facilitate further development and deployment of these or similar procedures. The goals were addressed by building a proof-of-concept dynamic integrated model in two urban areas: Jacksonville, Florida, and Sacramento, California.

A Volume 1 report describes the Sacramento, California, integration of the activity-based demand model DaySim; a Dynamic Traffic Assignment (DTA) model, DynusT; and a transit network simulation model, FAST-TrIPs. This report describes the application of DynusT and FAST-TrIPs in the Sacramento region. All of the highway and transit network data required to run the models were assembled and are available with the software, which is open source.

A companion report and model set (SHRP 2 C10A) are available for the application in Jacksonville, Florida. This work has the same objective and uses DaySim as the demand model but uses TRANSIMS for the highway network assignment. Both model sets and software start-up guides are available from the Federal Highway Administration.

Travel demand models have been used for more than half a century to determine the need and estimate the usage of proposed new highway and transit systems. The majority of such models use Traffic Analysis Zones to aggregate demographic data and estimate interzonal travel demand for large time blocks (such as morning peak period). The interzonal demand is assigned to a link and node network to estimate likely roadway volumes.

Activity-based travel demand models are based on the disaggregate travel activity of individual travelers, not the aggregate behavior of all the travelers in a zone. They have the potential to better simulate behaviors such as time of day choice, route choice, mode choice, and trip chaining. As with real travelers, information on the state of the network is needed to make choices. The feedback loop from the network assignment may cause a simulated “traveler” to change route, time of day, or mode in response to network congestion. The model set iterates until convergence is reached—travel volumes and modes are stable after successive iterations.

Activity-based models have been available for some time, but are not widely used in production planning work. Dynamic Traffic Assignment models are network simulation tools that represent network travel conditions. Such simulation models are used for subarea traffic analysis but have not been linked to a demographically based demand model and used at the urban area scale. This project integrated the supply-and-demand side of transportation demand forecasting in order to test operational improvements to the highway system as well as capacity enhancements.

The Sacramento Area Council of Governments (SACOG) used the model set to test transportation alternatives. The results are proof-of-concept in nature. The integrated model works and demonstrates potential improved sensitivity to policies that affect regional travel. However, the model did not converge as hoped so it was not possible to fully calibrate it prior to testing by SACOG. The integrated model sets built for this project are available as a basis for implementing a similar approach in other urban areas.

CONTENTS

Executive Summary

The second Strategic Highway Research Program (SHRP 2) Project C10B, Partnership to Develop an Integrated, Advanced Travel Demand Model with Fine-Grained, Time-Sensitive Networks: Sacramento-Area Application, is an important step in the evolution of travel modeling from an aggregate, trip-based approach to a completely dynamic, disaggregate methodology. In this project, an existing disaggregate activity-based model (ABM) was integrated with an existing traffic simulation model to create a new, completely disaggregate model.

This report describes the DynusT (Dynamic Urban Systems for Transportation) dynamic traffic assignment (DTA) model and the FAST-TrIPs (Flexible Assignment and Simulation Tool for Transit and Intermodal Passengers) DTA package and how these two model systems interact with each other and with the DaySim ABM implemented in Sacramento, California, to compose an integrated ABM and DTA model. This work was performed as part of the SHRP 2 Project C10B, Partnership to Develop an Integrated, Advanced Travel Demand Model with Fine-Grained, Time-Sensitive Networks. DynusT is a simulation-based DTA model capable of performing daily regional simulations of large metropolitan areas that involve many millions of trips, a feature necessary for DTA and ABM integration. This document describes in detail the traffic simulation and assignment capabilities of DynusT that capture capacity constraints, congestion, and queue propagation for various types of vehicles, including transit vehicles, and allow the generation of time-dependent level-of-service (LOS) measures that are closer to traffic theory. It also describes in detail the methodology that is used internally by DynusT to determine the time-dependent least-cost path route for each driver, a concept that is described as “dynamic user equilibrium.” FAST-TrIPs is a region-wide DTA model that determines an individual-specific transit route for each transit traveler in the system, taking into account published transit schedules and transit vehicle run times that are congestion responsive and are provided by the traffic simulation component of DynusT. FAST-TrIPs deals with both transit-only and park-and-ride trips and is able to maintain multiple constraints associated with activity time-windows and the choice of modes in multimodal travel tours. This report describes how DynusT and FAST-TrIPs interoperate with each other to provide a model system in which the highway and transit assignments influence each other and are based on the same set of LOS variables.

CHAPTER 1

Introduction

The second Strategic Highway Research Program (SHRP 2) Project C10B, Partnership to Develop an Integrated, Advanced Travel Demand Model with Fine-Grained, Time-Sensitive Networks, is an important step in the evolution of travel modeling from an aggregate, trip-based approach to a completely dynamic, disaggregate methodology. In this project, an existing disaggregate activity-based model (ABM) was integrated with an existing traffic simulation model to create a new, completely disaggregate model. Both models were implemented using open source software.

At the same time that travel demand models have been evolving, traffic simulation models, which simulate the movements of vehicles through a highway network, have become more sophisticated due to improvements in computing. The product of SHRP 2 C10B is an integrated model that simulates individuals' activity patterns and travel and their vehicle and transit trips as they move on a real-time basis through the transportation system. It produces a true regional simulation of the travel within a region, for the first time using individually simulated travel patterns as input rather than aggregate trip tables to which temporal and spatial distributions have been applied to create synthetic patterns. A unique feature of this model is the simulation of transit vehicles as well as individual person tours using transit.

The new integrated model has been developed and implemented for the entire Sacramento, California, region. The integrated model components include SACSIM, the regional travel model maintained by the Sacramento Area Council of Governments (SACOG), the regional metropolitan planning organization (MPO), and DynusT (Dynamic Urban Systems for Transportation), a mesoscopic traffic simulation model developed by the University of Arizona. SACSIM includes an activity-based demand model, DaySim. The transit simulation is performed by FAST-TrIPs (Flexible Assignment and Simulation Tool for Transit and Intermodal Passengers), also developed by the University of Arizona. The integrated model also includes the ability to run MOVES, the air quality analysis program developed by the U.S. Environmental Protection Agency.

While the C10B integrated model produces reasonable results for regional travel patterns and behavior, the true value of the model is its ability to provide analysis results that demonstrate sensitivity to policy variables more accurately than models that use aggregate demand or assignment procedures. This sensitivity was tested through a series of policy and project tests conducted by SACOG using the new integrated model and the existing SACSIM model with aggregate assignment.

The SHRP 2 C10B project has been documented in a series of four reports:

- SHRP 2 C10B summary report,
- User's manual for the integrated model,
- Network report on SHRP 2 C10B version of DynusT and FAST-TrIPs, and
- Software start-up guide.

This report, the third in the series, describes the theoretical background and methodology for the integration of DynusT, a mesoscopic dynamic traffic assignment (DTA) model, and FAST-TrIPs, a public transit passenger assignment and simulation model. The DynusT and FAST-TrIPs integrated system is designed to be a "loosely coupled" system. This design allows the DynusT and FAST-TrIPs teams to develop and test components in parallel following separate software development cycles. In the SHRP 2 C10B project, the travel demand data are simulated by the DaySim ABM implemented in Sacramento, California.

The overall FAST-TrIPs model structure is illustrated in Figure 1.1, which depicts the software components as well as the files needed for the DynusT and FAST-TrIPs communication. First, data from the General Transit Feed Specification (GTFS), originally known as the Google Transit Feed Specification, are processed and converted to transit route layouts, stops, and

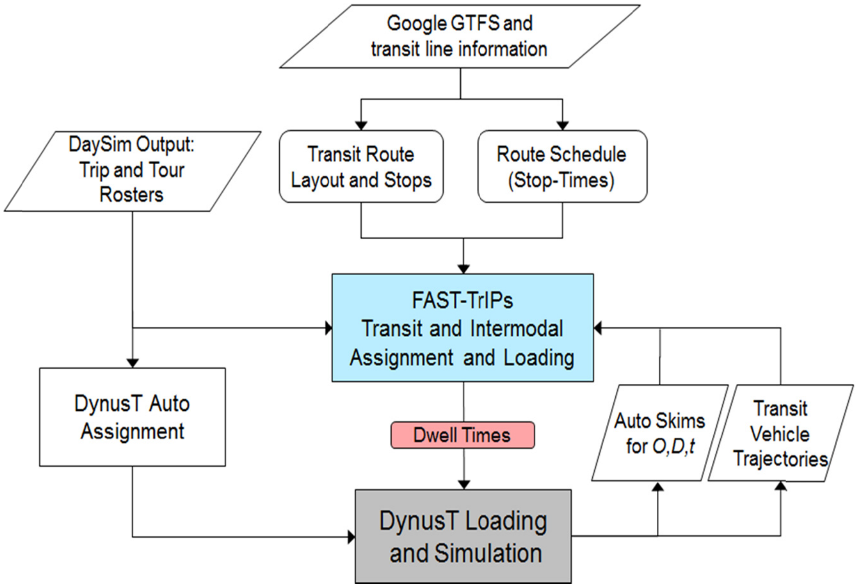


Figure 1.1. DynusT–FAST-TrIPs integration framework.

route schedule input files used by FAST-TrIPs. The FAST-TrIPs assignment procedure accepts demand data from DaySim and roadway link travel times from DynusT and assigns travelers to transit paths and transit vehicles. The DynusT DTA model simulates the DaySim trip and tour rosters and the transit vehicles from the FAST-TrIPs component. The network-wide LOS measures that result from the DynusT assignment become inputs to the FAST-TrIPs model in an iterative process until convergence is reached.

As shown in Figure 1.2, DynusT and FAST-TrIPs maintain a loose coupling integration architecture. This integration architecture is of particular advantage in reducing development

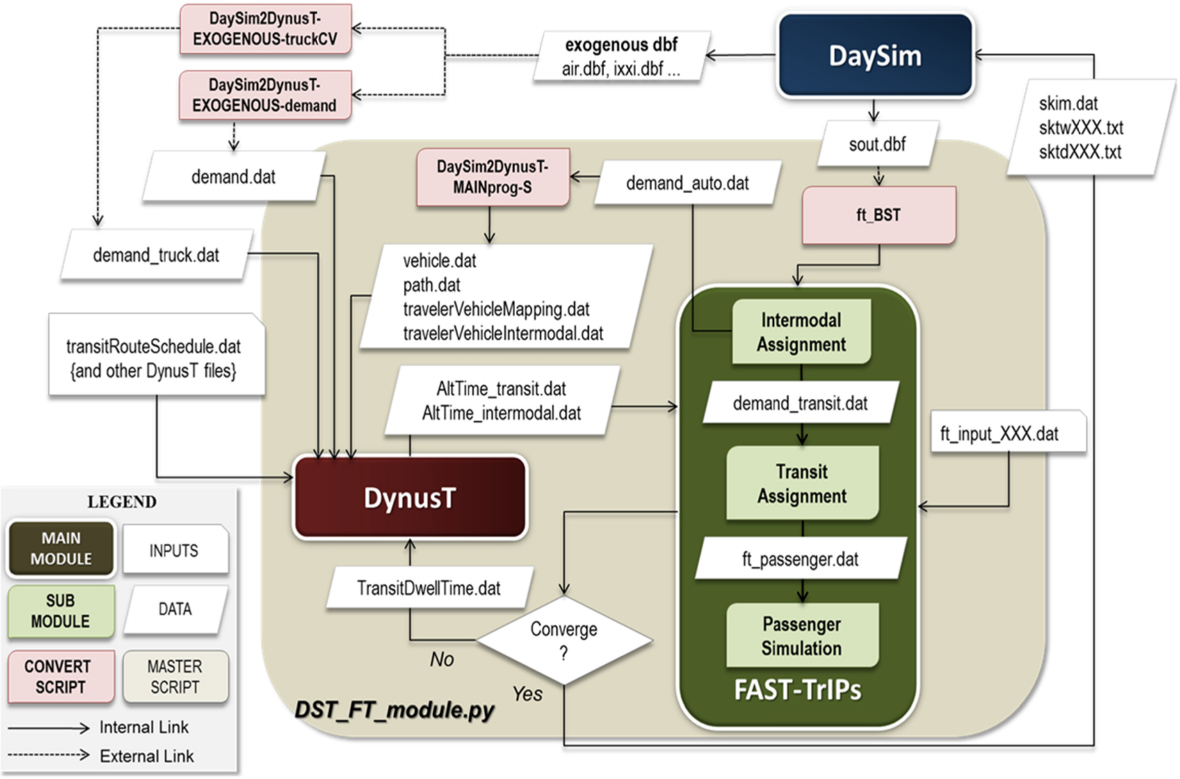


Figure 1.2. Interfacing mechanism in DynusT–FAST-TrIPs integrated system.

risk. This method allows FAST-TrIPs, a completely new development from the C10B project, to be prototyped, developed, and tested as a stand-alone module, allowing robust and controlled testing and debugging along the development process. As a result, communications between DynusT and FAST-TrIPs are conducted via flat text files.

In addition to the GTFS files, other inputs to the entire system include the DaySim output files containing trip and tour rosters (sout.dbf) and non-ABM trips, such as airport or exogenous origin–destination (OD) (exogenous.dbf and air .dbf, ixxi.dbf, etc.). The ft_BST script processes sout.dbf and produces demand_auto.dat. Demand_auto.dat contains the auto trip roster and is further processed into vehicle.dat and path.dat as the main input files for DynusT. The transit route and trip schedule information is organized and prepared as transitRouteSchedule.dat file from GTFS file format.

DynusT takes both the airport demand (processed into demand.dat) and external freight traffic (processed into demand_truck.dat) as the trip roster (vehicle.dat and path .dat) and performs simulation and dynamic assignment until reaching convergence in auto demand.

Once DynusT reaches convergence transit vehicle trajectories (AltTime_transit.dat and AltTime_Intermodal.dat), combining demand_transit.dat are processed and fed into FAST-TrIPs for transit assignment. Essentially AltTime_transit.dat and AltTime_intermodal.dat describe the transit vehicle movements, which can be considered as the supply-side information, whereas demand_transit.dat contains the demand-side information. The transit assignment procedure essentially determines the assignment of passengers to various transit route and transit stop choices.

The outcome of FAST-TrIPs procedure determines how many passengers will get on and off at each stop, and consequently determines the transit vehicle dwell time at each stop (TransitDwellTime.dat). If FAST-TrIPs is deemed not converged, DynusT is rerun again with the updated transit dwell time as the important input to DynusT.

On the other hand, if the FAST-TrIPs convergence criterion is met, the entire DynusT–FAST-TrIPs integrated procedure is considered completely converged, and the system will output necessary skim information in terms of skim.dat, skimXXX .dat. Such information will be fed into DaySim ABM.

This report is structured as follows: A detailed description of the DynusT model is presented in Chapter 2; Chapter 3 describes the enhancements that were made to DynusT to work with the DaySim ABM model; and Chapter 4 provides an in-depth exposition of the theoretical models used in FAST-TrIPs.

CHAPTER 2

The DynusT DTA Model

DynusT (Dynamic Urban Systems in Transportation) is a model system that is designed and implemented to perform simulation-based DTA and associated analysis. Due to its unique algorithmic structure and software implementation, it is capable of performing DTA on regional-level networks over a long simulation period. This makes DynusT particularly well-suited for regional-level modeling such as regional transportation planning, corridor studies, integration with activity-based models, and mass evacuation modeling. The purpose of this chapter is to present an overview of the theoretical and algorithmic innovations in DynusT.

As shown in Figure 2.1, DynusT consists of two main modules: traffic simulation and traffic assignment. Vehicles are created and loaded into the network based on their respective origins and follow a specific route to their intended destinations. The large-scale simulation of network-wide traffic is accomplished through the mesoscopic simulation approach that omits inter-vehicle car-following details while maintaining realistic macroscopic traffic properties (e.g., speed, density, and flow). More specifically, the traffic simulation is based on the Anisotropic Mesoscopic Simulation (AMS) (Chiu et al. 2010) technique that calculates a vehicle's speed from the traffic conditions ahead of the vehicle. Specifically, at each simulation interval, a vehicle's speed is determined by a speed–density curve, the density being the number of vehicles per mile per lane within a limited forward distance designated as SIR (speed-influencing region).

What is referred to as the traffic assignment module of DynusT consists of two algorithmic components: a time-dependent shortest path (TDSP) algorithm and a time-dependent traffic assignment, or routing. The TDSP algorithm determines the TDSP for each origin, destination, and departure time, and the traffic assignment component selects a route for each driver following some heuristic rules that are proven to lead to approximate user equilibrium, a condition in which each driver has selected the least-cost path available to him.

In DynusT, the assignment algorithm maintains the balance of computational efficiency and solution algorithm quality. Innovations in computational efficiency allow DynusT to perform 24-h assignment, which is a requirement for ABM and DTA integration.

After shortest paths have been calculated and a route choice has been made, all the vehicles are simulated. DynusT uses the time gap between a vehicle's simulated travel time and the vehicle's available shortest path time to assess the level of convergence. If the average time gap for all the vehicles in the simulation is small enough (usually 1% to 2%), DynusT terminates and outputs network-wide LOS measures; otherwise, it continues iterating between its two models until convergence is achieved.

The rest of this chapter is structured as follows: the first section presents the AMS model that powers the traffic simulation component of DynusT, and the second section discusses the traffic assignment algorithm that is deployed to achieve convergence.

Anisotropic Mesoscopic Simulation Model

The AMS model was based on research published by Chiu et al. (2010) and developed based on two empirical traffic rules: (1) At any time, a vehicle's prevailing speed is influenced only by the vehicles in front of it, including those that are in the adjacent lanes, and (2) the influence of downstream traffic decreases with increasing distance. These two characteristics define the anisotropic property of the traffic flow and provide the guiding principle for AMS model design. Based on the above, for any vehicle *i,* only the downstream vehicles within a certain distance influence vehicle *i*'s speed. This is a similar concept to a stimulus-response type of car-following model, with the distinction that in AMS, the stimulus of a vehicle's speed response is represented in a macroscopic manner instead of using intervehicle distance or speed as in microscopic models.

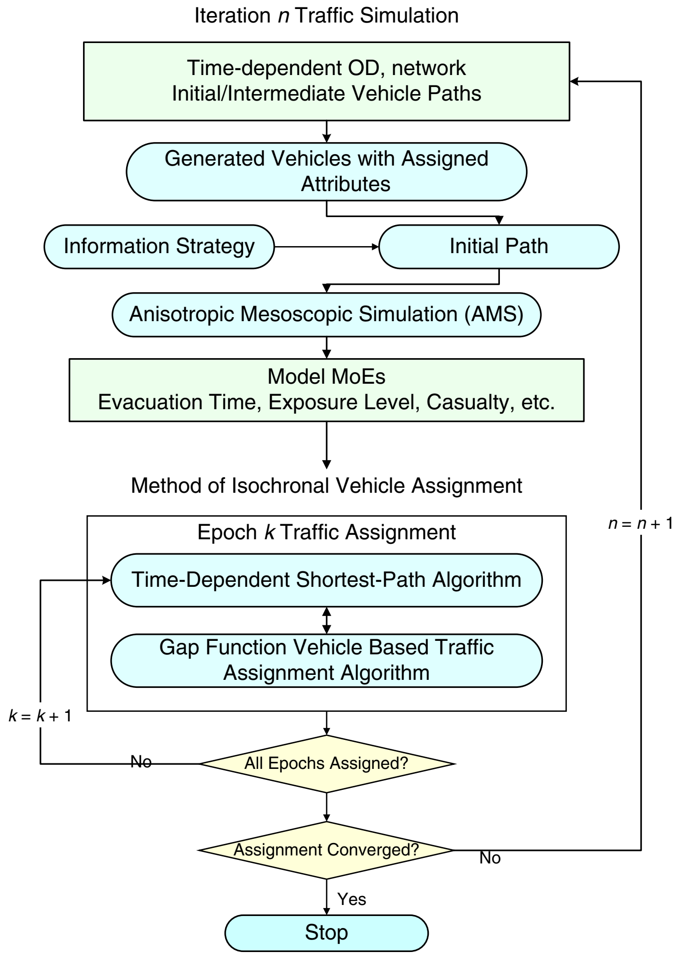


Figure 2.1. Traffic simulation, assignment, and link volume estimation framework in DynusT.

For modeling purposes, the SIR for vehicle *i* (*SIR*$_i$) is defined as vehicle *i*'s immediate downstream roadway section in which the stimulus is significant enough to influence vehicle *i*'s speed response. This concept is further depicted in Figure 2.2, AMS model concept, in which a multilane homogeneous roadway segment is considered. The SIR for vehicle *i* is defined as the area (including the lane in which vehicles reside and all the adjacent lanes) in front of vehicle *i*, where the traffic condition (represented by the density) affects vehicle *i*'s speed response. At each simulation clock tick, vehicle *i*'s speed is influenced by the density in the SIR. The upstream traffic and downstream traffic outside the SIR does not influence vehicle *i*. The traffic density in *SIR*$_i$, denoted as k_i, is calculated as the number of vehicles present in *SIR*$_i$ divided by the total lane-miles of the *SIR*$_i$. As such, the unit of k_i becomes the number of vehicles per mile per lane.

At the beginning of a simulation interval *t*, the prevailing speed of vehicle *i* during the simulation interval *t* is determined by Equation 2.1, which is a non-increasing speed–density relationship function with the following properties: density does not influence speed up to a certain density threshold and, for density greater than the threshold, value increases in density result in decreases in speed. The maximum density in DynusT is the "bumper-to-bumper" density observed in a long, standing-still queue.

The algorithmic steps of an AMS model during simulation are as follows: At each clock tick *t* (the beginning of a simulation interval), each vehicle's speed is evaluated based on its SIR density, which is obtained from the previous clock tick $t - 1$ through the speed–density (v-k) relationship of Equation 2.1. The SIR density is calculated based on Equation 2.2 or Equation 2.3, depending on whether the SIR spans over a roadway segment with a different capacity. If the SIR spans a heterogeneous highway section, Equation 2.2 applies; otherwise, the relationship is simplified to Equation 2.3. Vehicle *i*'s traveling distance at the end of the current simulation interval is obtained by multiplying the prevailing speed with the duration of the simulation interval.

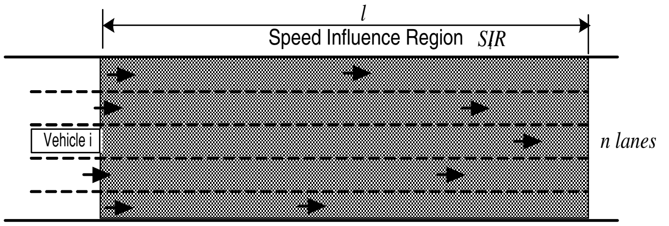
$$v_i^t = \wp\left(k_i^{t-1}\right) \tag{2.1}$$

$$k_i^{t-1} = \min\left[k_{queue}, \frac{N_i^{t-1}}{m x_i^{t-1} + n\left(l - x_i^{t-1}\right)}\right] \tag{2.2}$$

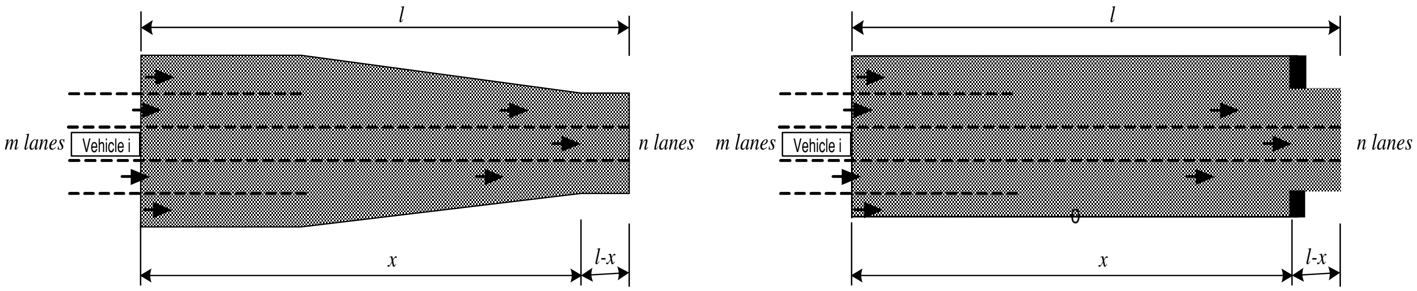
$$k_i^{t-1} = \min\left[k_{queue}, \frac{N_i^{t-1}}{nl}\right] \tag{2.3}$$

i = Subscript denoting a vehicle. The index i decreases with vehicles traveling in the same direction on the same link,
t = Superscript denoting a simulation interval,
n = Number of lanes downstream of lane add/drop,
m = Number of lanes upstream of lane add/drop,
l = SIR length,
v_i^t = Prevailing speed of vehicle i during simulation interval t,
x_i^{t-1} = Distance between vehicle i and lane-drop (open) at clock tick $t - 1$,
k_i^{t-1} = Density of the SIR for vehicle i,
N_i^{t-1} = Number of vehicles present in SIR, excluding vehicle i,
v_f = Free-flow speed in the speed–density relationship,
$\wp : k \rightarrow v$ = Non-increasing speed–density function specifying the v-k relationship, where $\wp(0) = v_f$ and $\wp(k_{queue}) = 0$, and
k_{queue} = Queue density, $\wp(k_{queue}) = 0$.

During the AMS simulation, each vehicle maintains its own desired speed and SIR. Individual vehicles' traveling distances are therefore likely to differ, even though they are on the same link. This feature is different from certain previous models

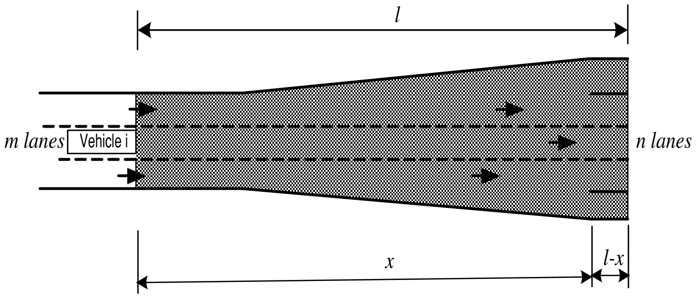


(a) Homogeneous Highway

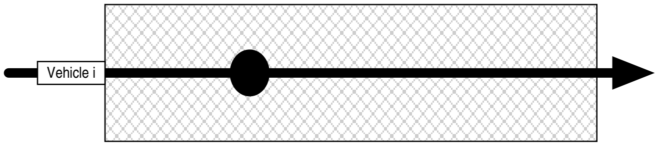


(b-1) Non-Homogeneous Highway (lane drop)

(b-2) Non-Homogeneous Highway (point bottleneck)



(c) Non-Homogeneous Highway (bottleneck discharge)



(d) AMS with SIR extending beyond current link boundary

Figure 2.2. AMS model concept.

(Jayakrishnan et al. 1994, Balakrishna et al. 2005), in which all moving vehicles on the same link travel at the same speed. This characterizes the AMS model as a vehicle-based mesoscopic model having a greater degree of resemblance with car-following–based microscopic models. The major difference between AMS and car-following models is that in AMS, a vehicle's speed adjustment at each simulation time interval is governed by the SIR density, which is a macroscopic measure of all the vehicles present in the SIR, instead of an intervehicle measure between the target vehicle and the leading vehicle(s).

Since the SIR moves with each vehicle during simulation, it can be anticipated that in the AMS model, the vehicle advancing mechanism is generally independent of the mathematical representation of the roadway network (size/length of cell/segment/link). AMS simulation results generally remain stable regardless of whether link lengths are shorter than the simulation interval multiplied by the speed.

AMS handles queue formation/discharge in a natural and straightforward manner. When the maximum value of (k_{queue}) is reached, vehicles get zero speed: $v = \wp(k_{queue}) = 0$. When density in the SIR region decreases, affected vehicles speed up. This mechanism allows for clear representations of substantial or transient queue formation or discharge. When a free-moving vehicle approaches the end of a queue, its speed gradually approaches the speed of the vehicles at the queue's tail.

Equation 2.1 is further extended to simulate traffic flow in uninterrupted flow facilities under various configurations, such as homogeneous highways, non-homogeneous highways, and temporary blockage, by incorporating different SIR and density k_i^t calculations. In the case of homogeneous highways, k_i^t is calculated as the number of vehicles present in the SIR divided by the total lane-miles of the SIR (Equation 2.2). When lane drops or lane additions occur within the SIR, the total lane-miles of SIR is the sum of lane-miles of separate sections, as shown in Equation 2.3 and depicted in Figure 2.2, AMS model concept (where m = the number of lanes at the beginning of the section and n = the number of lanes at the end of the section).

In the case of a lane drop or a point bottleneck ($n < m$), the SIR density of a vehicle gradually increases (and, hence, speed reduces) as it approaches the bottleneck. When $n = 0$, a complete blockage occurs; this can be applied to either the point blockage or red-light signal indication. On the other hand, in the case of discharging from a bottleneck, as a vehicle approaching the open-up of the bottleneck, the density reduces and speed increases gradually.

In this project, the speed–density function takes the following two Greenshield types of function forms.

$$v^{cal} = \begin{cases} v_f, & k \le k_b \\ v_f\left[1-\left(\dfrac{k-k_b}{k_q-k_b}\right)^{\beta}\right]^{\alpha}, & k_b \le k \le k_q \end{cases} \tag{2.4}$$

$$v^{cal} = \begin{cases} v_f, & k \le k_b \\ v_f\left[1-\left(\dfrac{k-k_b}{k_q-k_b}\right)\right]^{\alpha}, & k_b \le k \le k_q \end{cases} \tag{2.5}$$

Gap Function Vehicle-Based Traffic Assignment

In the assignment, the proposed gap function vehicle-based (GFV) algorithm adopts a gradient projection concept, where path flow updates are composed of both a gradient and a step size. The algorithm also takes advantage of the vehicle-based simulation, allowing reassigned selected individual vehicles with better paths to improve their travel times. The gradient projection approach is common in constrained nonlinear programming and has been applied in many classical static and DTA works (Dafermos and Sparrow 1969; Florian and Nguyen 1974; Sheffi 1985; Smith 1993). The route-swapping heuristics developed in more recent years have been found to be related to the same concept, except that in the route-swapping heuristics the step size is usually a pre-determined swapping rate, and the direction is linearly proportional to the travel time difference between the current vehicle travel time and the shortest path travel time (Smith and Wisten 1995; Huang and Lam 2002; Szeto and Lo 2006). In a more recent study the swapping rate was proposed to be the ratio of the difference of the individual path travel time to the shortest path travel time (Lu 2007; Lu et al. 2008a).

In the GFV method used in the SHRP 2 C10B model, the step size is in relation to the relative gap value calculated for all the paths between each origin–destination–departure time (i, j, τ) triplet at iteration l. Similar in concept to prior studies, the GFV method leads to a smaller step size with a smaller relative gap value. The gradient determines the search direction, where "search direction" means those paths to be updated with more or fewer vehicles. For each (i, j, τ) triplet, paths are sorted according to the average travel time, and vehicles traveling on each path are also sorted according to decreasing experienced travel time. Note that the assignment interval is normally much longer than the simulation interval. Therefore, vehicles departing within the same assignment interval, while subject to the same path set, would experience different travel times due to their different departure times during the simulation interval.

Furthermore, at each iteration, once the assignment is completed, the path set $K^{l+1}(i, j, \tau)$ $\forall$ all (i, j, τ) is released from memory. At the beginning of each iteration, the path set is reconstructed by scanning through each vehicle and assigned path. This may cause a slight increase in computation, but it eliminates the need to retain the path set $K^{l+1}(i, j, \tau)$ $\forall$ all (i, j, τ) when such information is not used (e.g., in simulation), thus reducing the peak memory needed during the entire simulation-assignment procedure. It should also be noted that this strategy is likely to deplete high-travel-time paths of vehicles. As a result, such paths will not appear in the next iteration path set.

Different from methods that explicitly solve for the optimal flow distribution among paths within $K^l(i, j, \tau)$ (Chang and Ziliaskopoulos 2004; Lu et al. 2008b), the determination of the step size and gradient are based on a simple joint approximation of the descent direction and step size. This strategy is adopted after a careful consideration of the trade-offs between the solution quality and computational tractability. The methods that explicitly solve for the optimal distribution given $K^l(i, j, \tau)$ require significantly more computation time that may make the algorithm computationally intractable unless a parallel computing scheme is used. Overall, the GFV method used in SHRP 2 C10B exhibits satisfactory convergence quality for a reasonable computation time.

A schematic representation of the GFV algorithm as part of DynusT is illustrated in Figure 2.3. The iterative simulation assignment is initialized with the primary inputs of network loading: demand patterns, time-dependent OD matrices, and initial path assignments. As the AMS model simulates vehicles within the network, evaluations of time-varying link densities, link flows, travel times, and speeds are made (Chiu

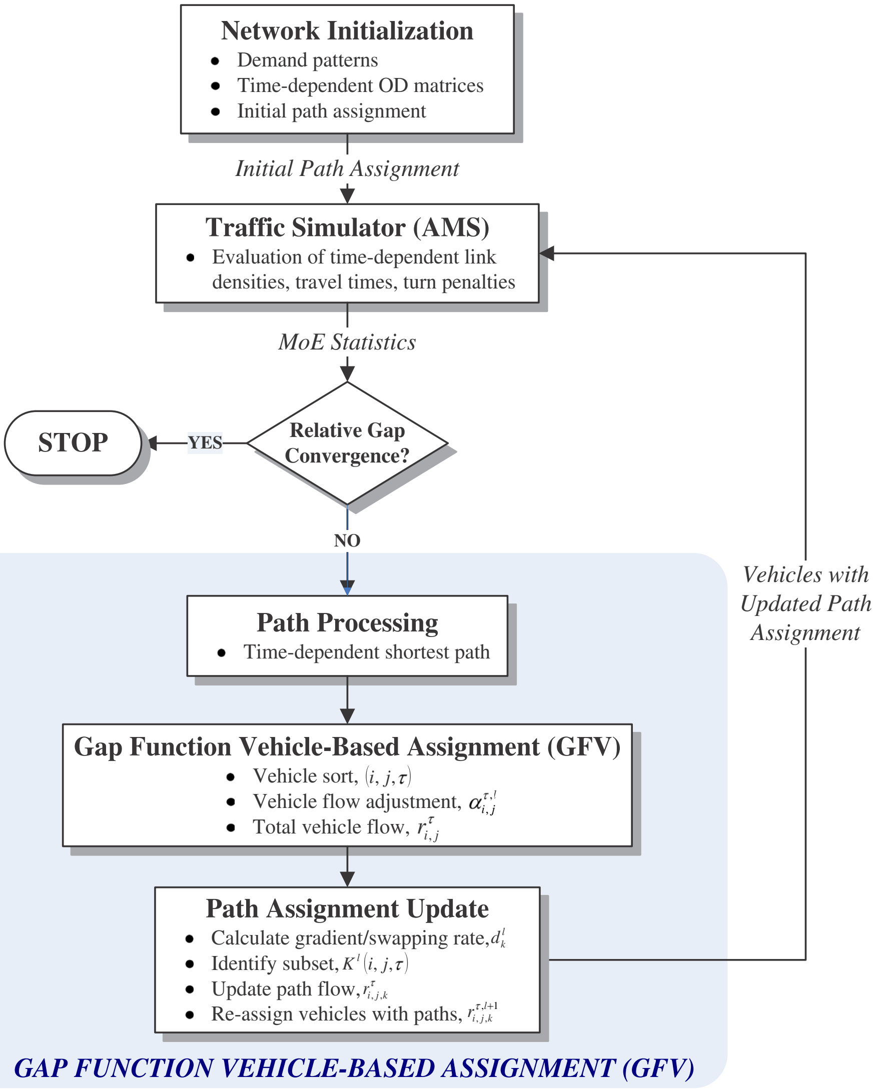


Figure 2.3. GFV method in conjunction with DynusT.

et al. 2010). After the initial network loading, the interplay between GFV and the AMS simulation continues until the convergence criterion is met. The convergence criterion is the gap function value, which is further discussed below.

The GFV procedure starts by sorting vehicles for all $K^l(i, j, \tau)$ based on vehicles' experienced travel times. Note that vehicles are loaded into the network on links; therefore, the origin node i for a vehicle refers to the downstream node of the link on which the vehicle is loaded.

In the GFV algorithm, the step size is in relation to the relative gap (RG) value, calculated $\forall k \in K^l(i, j, \tau)$. Furthermore, the relative gap RG_k for path k defined in Equation 2.6 indicates that q_v is the experienced travel time of vehicle v (from the downstream node of origin link i to destination node j) and $u^{\tau,l}_{i,j}$ is the calculated time-dependent shortest path travel time for (i, j, τ) at iteration l solved by the TDSP algorithm. RG_k measures the travel time deviation of path k in comparison with the shortest path.

$$RG_k = \frac{\sum_{v\in V^l(i,j,\tau,k)} q_v - r^{\tau,l}_{i,j,k} \cdot u^{\tau,l}_{i,j}}{r^{\tau,l}_{i,j,k} \cdot u^{\tau,l}_{i,j}}, \forall k \in K^l(i,j,\tau) \tag{2.6}$$

The stopping criterion follows Equation 2.7.

$$\overline{RG} \le RG^0 \tag{2.7}$$

where RG^0 is the user-specified threshold and $\overline{RG}$ follows Equation 2.8.

$$\overline{RG} = \frac{\sum_{i,j,\tau,k} \sum_{v\in V^l(i,j,\tau,k)} q_v - \sum_{i,j,\tau}\left(r^{\tau}_{i,j} \cdot u^{\tau,l}_{i,j}\right)}{\sum_{i,j,\tau}\left(r^{\tau}_{i,j} \cdot u^{\tau,l}_{i,j}\right)} \tag{2.8}$$

Next, the time-dependent shortest path is solved. At each iteration *l*, the flow for each $k \in K^l(i, j, \tau)$ to be shifted at this iteration is the step size $\alpha_{i,j}^{\tau,l}$ times the total flow $r_{i,j}^{\tau}$ between the (i, j, τ) triplet. $\alpha_{i,j}^{\tau,l}$ is defined as the minimal of two candidate step sizes as shown in Equation 2.9. α' is the *RG*-based step size, calculated based on Equation 2.10; α^0 is the maximum step size. The step size is determined by Equation 2.7 as the *RG*-based step size is calculated as average RG value for all path $k \in K^l(i, j, \tau)$:

$$\alpha_{i,j}^{\tau,l} = \min\{\alpha^0, \alpha'\} \tag{2.9}$$

$$\alpha' = \left\{ \frac{\sum_k RG_k}{|K^l(i, j, \tau)|} \right\} \tag{2.10}$$

$|K^l(i, j, \tau)|$ is the cardinality of the set of non-zero flow paths between criterion (i, j, τ) for iteration *l*. Paths in $K^l(i, j, \tau)$ are ordered with decreasing travel time. Note $K^l(i, j, \tau)$ includes the TDSP solved at the current iteration.

Based on Equation 2.6 through Equation 2.10, one should expect that since the algorithm starts from an all-or-nothing (AON) assignment, $|K^l(i, j, \tau)|$ is small and RG_k may be large; therefore, the step size is initially capped at α^0. As iterations increase, $|K^l(i, j, \tau)|$ also increases while RG_k decreases. With the step size capped by α^0 at initial iterations, as iterations increase, the step size then will be handled by α' as the size of path sets $K^l(i, j, \tau)$ would eventually stabilize. The value of α' generally continues to decrease as the time-dependent user equilibrium (TDUE) solution is iteratively improved.

If the assignment starts from an initial simulation in which vehicles are being loaded following TDUE paths solved from a prior TDUE run, the step size in the initial iteration is likely to be capped by α'. This reduces excess fluctuation of the initial assignment away from the TDUE condition since the initial solution is already close to the TDUE condition. Only those $K^l(i, j, \tau)$ exhibiting large gaps are to be shifted with a larger flow amount. This observation implies a relatively stable and quick convergence for a warm-start assignment.

Next, the descent direction d_k^l updates the direction of flow adjustment at iteration *l* using what are considered increasing-flow and decreasing-flow path subsets. $K^l(i, j, \tau)^+$ defines the path subset that will have the increased flow after assignment while $K^l(i, j, \tau)^-$ is the path subset with decreased flow, where

$$K^l(i, j, \tau)^+ \cup K^l(i, j, \tau)^- = K^l(i, j, \tau) \tag{2.11}$$

After determining the step size, the amount of the total shifted flow is $\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}$, by definition. It is important to note that in $K^l(i, j, \tau)$, paths are always ranked in the order of increasing path average travel time $\overline{q}_{i,j,k}^{\tau,l}$, which is

$$\overline{q}_{i,j,k}^{\tau,l} = \frac{\sum_{v \in V^l(i,j,\tau,k)} q_v}{|V^l(i, j, \tau, k)|}, \forall k \in K^l(i, j, \tau) \tag{2.12}$$

and

$$\overline{q}_{i,j,1}^{\tau,l} \le \overline{q}_{i,j,2}^{\tau,l} \le \cdots \le \overline{q}_{i,j,|K^l(i,j,\tau)|}^{\tau,l} \tag{2.13}$$

The decreasing-flow path set $K^l(i, j, \tau)^-$, as defined in Equation 2.14, is determined by scanning paths $k = 1, 2, \ldots, \hat{k}$ until the condition defined in Equation 2.15 is met. $\hat{k}$ is the cutoff path in which vehicles in any path $k \in K^l(i, j, \tau)$ whose $\overline{q}_{i,j,k}^{\tau,l} \ge \overline{q}_{i,j,\hat{k}}^{\tau,l}$ will be considered for reassignment.

$$K^l(i, j, \tau)^- = \left\{ k = \hat{k}, \ldots, |K^l(i, j, \tau)| \,\middle|\, \sum_{k=1}^{\hat{k}-1} r_{i,j,k}^{\tau,l} < \alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \le \sum_{k=1}^{\hat{k}} r_{i,j,k}^{\tau,l} \right\} \tag{2.14}$$

Equation 2.14 only determines the decreasing-flow path set; however, because the GFV algorithm is vehicle based, the step size $\alpha_{i,j}^{\tau,l}$ specifies a certain number of vehicles that would be reassigned to the increasing-flow path set. For that reason, all vehicles on path $k \in K^l(i, j, \tau)^- \backslash \hat{k}$ will be assigned to the increasing-flow path set, but only part of those on the cutoff path $\hat{k}$ will be assigned. $\hat{v}$ is defined as the cutoff vehicle on path $\hat{k}$ such that any vehicle beneath $\hat{v}$ is not considered for reassignment.

$$\hat{v} = \sum_{k=1}^{|K^l(i,j,\tau)^-|} r_{i,j,k}^{\tau,l} - \alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \tag{2.15}$$

Those vehicles belonging to $k \in K^l(i, j, \tau)^-$ will be reassigned with one of the paths in the set $K^l(i, j, \tau)^+$, which may also include the latest solved TDSP. The redistribution scheme is depicted in Equation 2.16:

$$d_k^l = \begin{cases} \dfrac{\left(RG_{\hat{k}} - RG_k\right)^{\theta}}{\sum_{k \in K(i,j,\tau)^+} \left(RG_{\hat{k}} - RG_k\right)^{\theta}}, & \forall k \in K^l(i,j,\tau)^+ \\[2ex] -\dfrac{r_{i,j,k}^{\tau,l}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}} & , \forall k \in K^l(i,j,\tau)^- \backslash \hat{k} \\[2ex] -\dfrac{\hat{v}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}} & , \hat{k} \in K^l(i,j,\tau)^- \end{cases} \tag{2.16}$$

where $RG_{\hat{k}}$ is the relative gap value for cutoff path $\hat{k}$.

For those paths $k \in K^l(i, j, \tau)^+$, the number of vehicles is increased by the amount

$$\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot \frac{\left(RG_{\hat{k}} - RG_k\right)^{\theta}}{\sum_{k \in K(i,j,\tau)^+} \left(RG_{\hat{k}} - RG_k\right)^{\theta}} \tag{2.17}$$

For a vehicle to be assigned to a path $k \in K^l(i, j, \tau)^+$, the probability can be expressed as

$$P(k) = \frac{\left(RG_{\hat{k}} - RG_k\right)^{\theta}}{\sum_{k \in K(i,j,\tau)^+} \left(RG_{\hat{k}} - RG_k\right)^{\theta}}$$

For those paths $k \in K^l(i, j, \tau)^- \backslash \hat{k}$ the change in the number of vehicles is

$$-\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot \frac{r_{i,j,k}^{\tau,l}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}}$$

For $k = \hat{k}$, the decrease amount is

$$-\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot \frac{\hat{v}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}}$$

Note that θ is a scaling factor which can be kept as a predefined constant or can be an increasing function of iteration number. The larger θ is, the more aggressive the assigned flow to the first best paths will be. The value used in the C10B project is 2.0.

Lastly, all flow determined to be reassigned are applied to all paths $k \in K^l(i, j, \tau)^+$ following a nonlinear proportional scheme:

$$r_{i,j,k}^{\tau,l+1} = r_{i,j,k}^{\tau,l} + \alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot d_k^l, \ \forall k \in K^l(i,j,\tau)$$

CHAPTER 3

DynusT Enhancement for the Integrated Model

Transit Simulation

Vehicle simulation under the presence of transit vehicles needs to properly differentiate the situation with or without bus pullouts. As illustrated in Figure 3.1, when a bus pullout is present and a transit vehicle resides in the pullout, passerby vehicles' SIR area remains unchanged. On the other hand, without the pullout the stopped transit vehicle typically blocks one traffic lane, creating a temporal blockage to the following traffic steam. The departure from each stop involves different rules for frequency or schedule-based transit. The main difference is that for schedule-based transit operation, a transit vehicle needs to be held until the scheduled departure time if the transit vehicle is still ahead of schedule after boarding and alighting. Such vehicle holding is unnecessary in frequency-based operation.

As previously mentioned, demand is generated from a tour-based travel demand model. Before this information can be used for traffic simulation and assignment purposes it must be manipulated to meet the traffic simulation and assignment model's specific network and demand inputs. In terms of demand, DynusT demand inputs take two forms: the typical OD table for specified time periods, and vehicle and path files. In general, the exogenous travel components (truck, external, and airport vehicle trips) yield OD demand files given diurnal factors while tour/trip records yield vehicle and path demand files.

Generating DynusT's vehicle demand file is a more involved process because it requires detailed trip information as opposed to OD demand files simply requiring OD and diurnal factors. Examples of this mandatory information include household identification (ID), traveler/person ID, tour/trip ID, OD parcels/points, OD zones, mode choice, travel time, value of time, and arrival/departure time. The purpose of this information is to represent a trip as realistically as possible within DynusT's node-link based network and context. Examples of this realistic representation not only entail correct zone vehicle generation or destinations but most importantly also ensure that a specific person's trip reaches its destination before his or her next trip (tour) is generated. This instance is usually prevalent in networks with congestion or disruption or where trips are sequenced immediately after one another.

Temporal Domain Decomposition Algorithmic Scheme for Large-Scale DTA Implementation

In order to support ABM integration, DynusT needs to perform 24-h simulation and assignment. This large spatial and temporal scale brings forth great computational challenges in algorithm design and implementation. The DynusT research team developed a temporal decomposition scheme called the Method of Isochronal Vehicle Assignment (MIVA) that has successfully overcome this computational challenge and enabled DynusT to perform megascale spatial- and temporal-scale dynamic traffic assignment.

The design concept is to divide the entire analysis period into epochs (evenly or based on computing loads). Vehicle assignment is performed sequentially and parallelly in each epoch. Vehicles generated in each epoch are assigned in a parallel (multi-threaded) fashion. This scheme has two principal advantages. First, it improves the model scalability by confining the peak runtime memory requirement as the maximum memory usage for each individual epoch regardless of the total analysis period, thus improving the memory efficiency. Secondly, the parallel processing of vehicle assignment further improves the runtime efficiency. A self-turning scheme adaptively searches for the runtime-optimal epoch setting during iterations regardless of the characteristics of the modeled network. The following explanations are the brief excerpt from Nava and Chiu (2012).

As shown in Figure 3.2, the common algorithmic framework of a typical simulation-based DTA model includes the iterative execution of network loading (simulation), the path

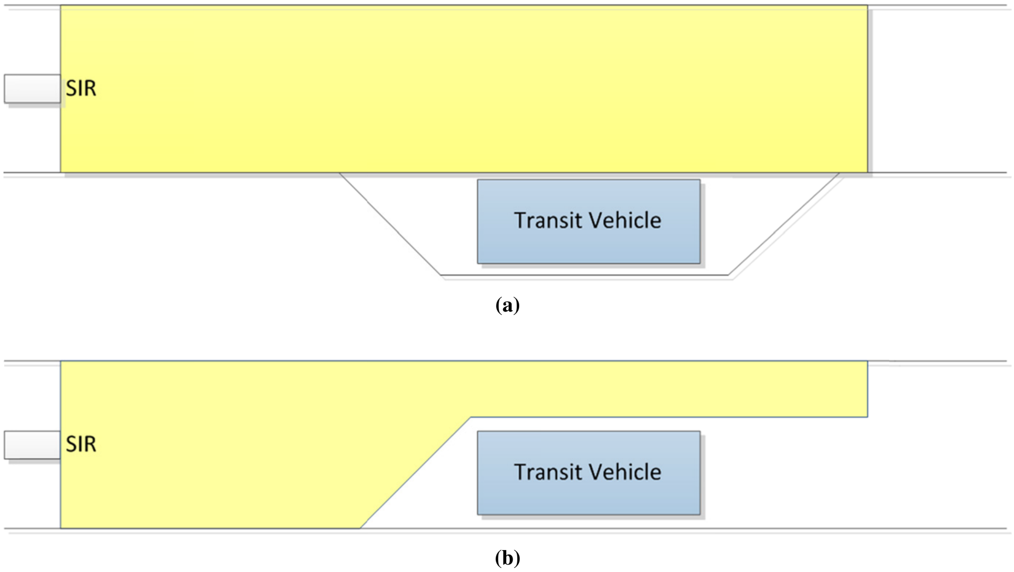
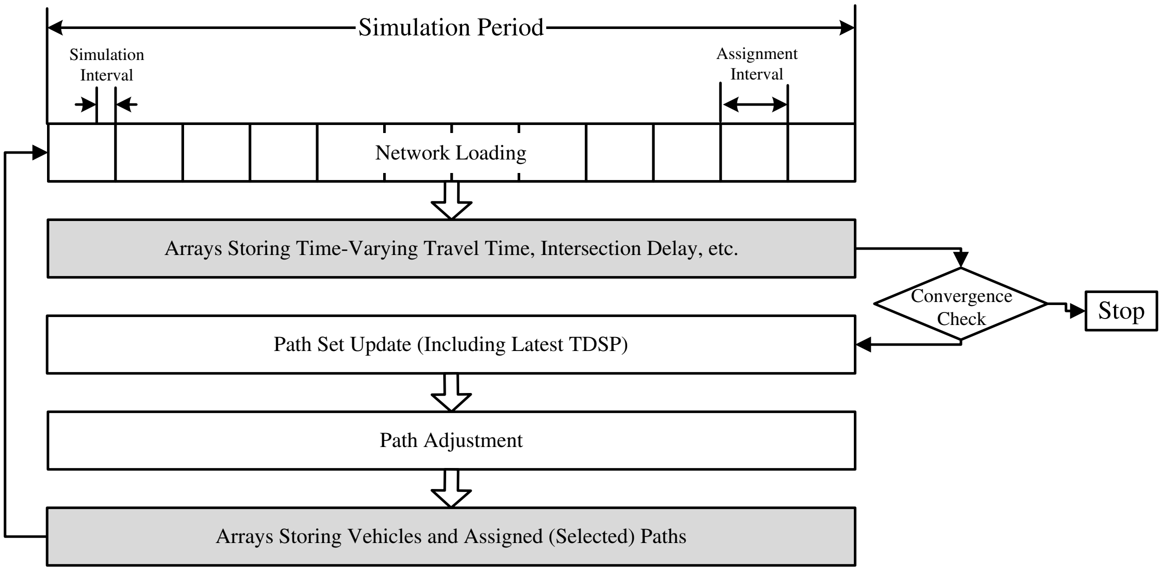


Figure 3.1. SIR areas with and without bus pullouts.

set update procedure (time-dependent travel time cost of routes for all origin, destination, and departure time triplets), and the path set adjustment (DTA) procedure to update vehicle paths. To determine how close the current solution is to the dynamic user equilibrium (DUE) condition, the evaluation of path assignments, by means of simulation, requires checking a defined convergence criterion. The algorithm terminates if the stopping criterion is met.

The proposed MIVA decouples the time domain between simulation and both the path set update and path adjustment procedures (composed of the TDSP and assignment solution algorithm) into forward-sliding time periods, which allows the memory requirement for both the path set update and path adjustment to be bounded solely on the length of the determined temporal segment instead of the entire analysis period.

The memory usage for storing time-varying link travel times is of memory size $|I| \times |T|$, where I is the set of links and T is the set of assignment intervals. Memory usage for storing the time-varying node (intersection) delay is of size $|I| \times |M| \times |T|$, where M is the set of movements. These arrays are of modest size, even for a large network and long analysis period. The TDSP memory requirement, depending on the algorithmic implementation, generally requires the memory for storing many arrays with dimension $|I| \times |M| \times |J| \times |T|$, where J is



Source: Chiu et al. 2011.

Figure 3.2. The simulation-based DTA algorithmic framework.

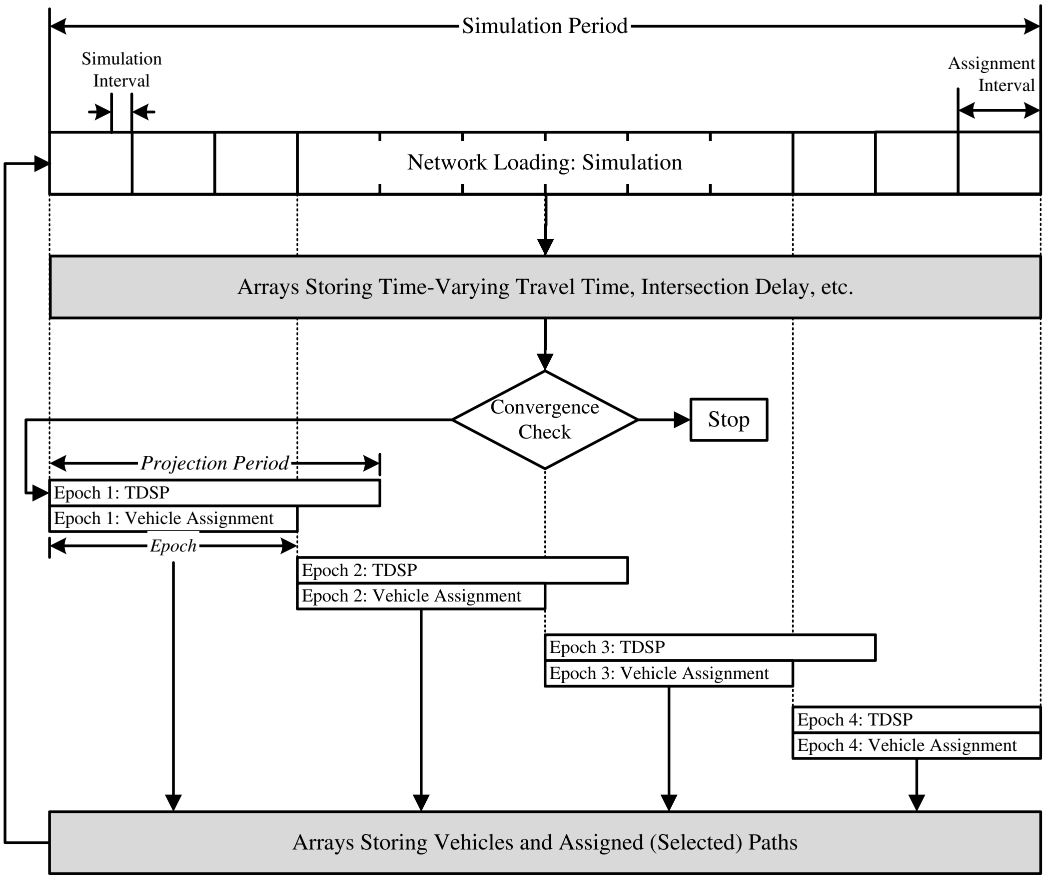
the set of destinations (or zones if the destination is the centroid of the zone). For example, a network with 20,000 links, 1,000 zones, 5 movements per intersection, and 100 departure intervals would need $1 \times E10$ elements to store information for one array, let alone the multiple arrays typically needed during full network TDSP computation. The applied TDSP algorithm for this research is label-correcting with complexity $O(nmT^2)$, where n is the number of nodes, m is number of links, and T is number of assignment intervals. It is apparent that the TDSP computational time will grow polynomially with network size and the analysis period. The memory usage for the assignment procedure, although varied by implementation, typically would require a significant amount of memory to store a time-dependent path set for each origin–destination–departure time triplet. The memory usage is linear in the temporal domain, but could be large if each path is stored in terms of individual nodes/links composing the path.

The MIVA Temporal Decomposition Framework

Shown in Figure 3.3, the MIVA scheme is denoted by two inter-associated time periods: epoch and projection period. The MIVA scheme decouples the temporal domain of the analysis period (also termed simulation period) into sequential segments of equal length called epochs. For vehicles departing within a single epoch, the arrival times to their destinations are used to estimate the time period, known as the projection period, in which the T domain for the TDSP algorithm is defined for the path set update for vehicles departing within the current epoch. At the end of one epoch, all TDSP and assignment-related memory is de-allocated, and then re-allocated for the next epoch. The MIVA scheme then slides the path set update and adjustment operations from one epoch to the next until completing all epochs. As a result, the memory usage during the entire path set update and adjustment operation is only a function of the epoch length.

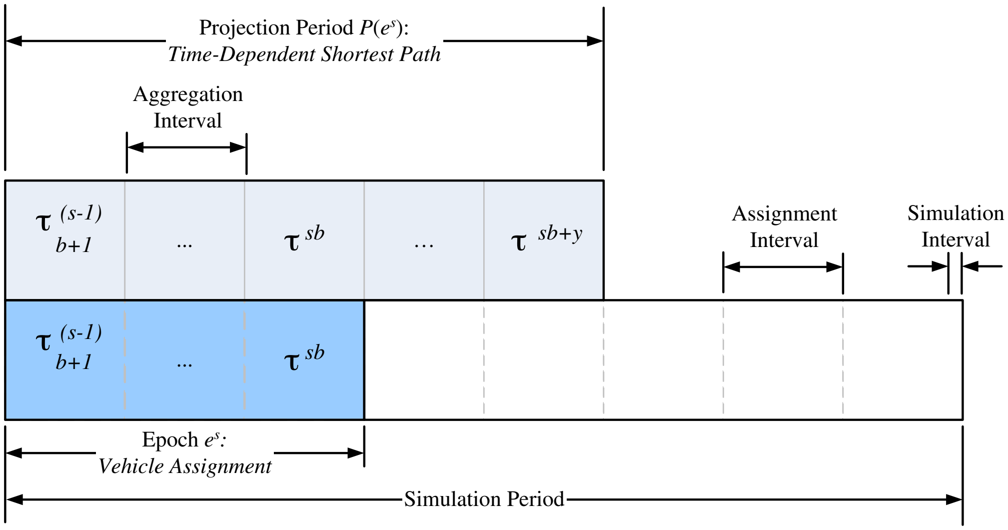
Epoch

The epoch is the partitioned period that acts as the temporal segment for the path adjustment procedure, meaning the TDSP and assignment procedure is bounded solely by the length of the epoch. An epoch consists of multiple assignment intervals (interchangeably termed as departure intervals as assignment is performed for vehicles departing at the same



Source: Nava and Chiu 2012.

Figure 3.3. The MIVA computational scheme implemented within the simulation-based DTA algorithmic framework.



Source: Nava and Chiu 2012.

Figure 3.4. The epoch and projection period of the MIVA structure.

departure interval). An aggregation interval pertains to the time interval in which traffic data (e.g., time-dependent link travel times and intersection delays) are averaged to be the input for the TDSP. The assignment interval is bounded by the number of simulation intervals. A simulation interval is defined as the time resolution that traffic simulation states are updated. An assignment interval is a multiple of simulation intervals, and, in the same manner, an epoch is a multiple of assignment intervals, as shown in Figure 3.4.

Let H be the simulation period and h be the length of the assignment/departure time interval in which H is discretized, resulting in a time discrete model. Let $T = \{\tau^1, \tau^2, \ldots, \tau^{H/h}\}$ be the set of departure time intervals. Let the length of each epoch (number of assignment intervals) be in terms of the integer number of assignment intervals $b = H/h/n$ where n is the prespecified total number of epochs within H. Let $E = \{e^1, e^2, \ldots, e^n\}$ be the set of epochs. Let $e^s = \{\tau^{(s-1)b+1}, \tau^{(s-1)b+2}, \ldots, \tau^{sb}\}, \forall\, e^s \in E$ be the set of assignment intervals for epoch e^s containing b number of assignment intervals. In each epoch domain there is a set of departing vehicles $V^{e^s}(i, j, \tau) \subseteq V$, where $V = \{v^1, v^2, \ldots, v^{|V|}\}$. Those vehicles $v \in V^{e^s}(i, j, \tau)$ are assigned based on the TDSP solved over the projection period associated with e^s, which will be further explained in the next section.

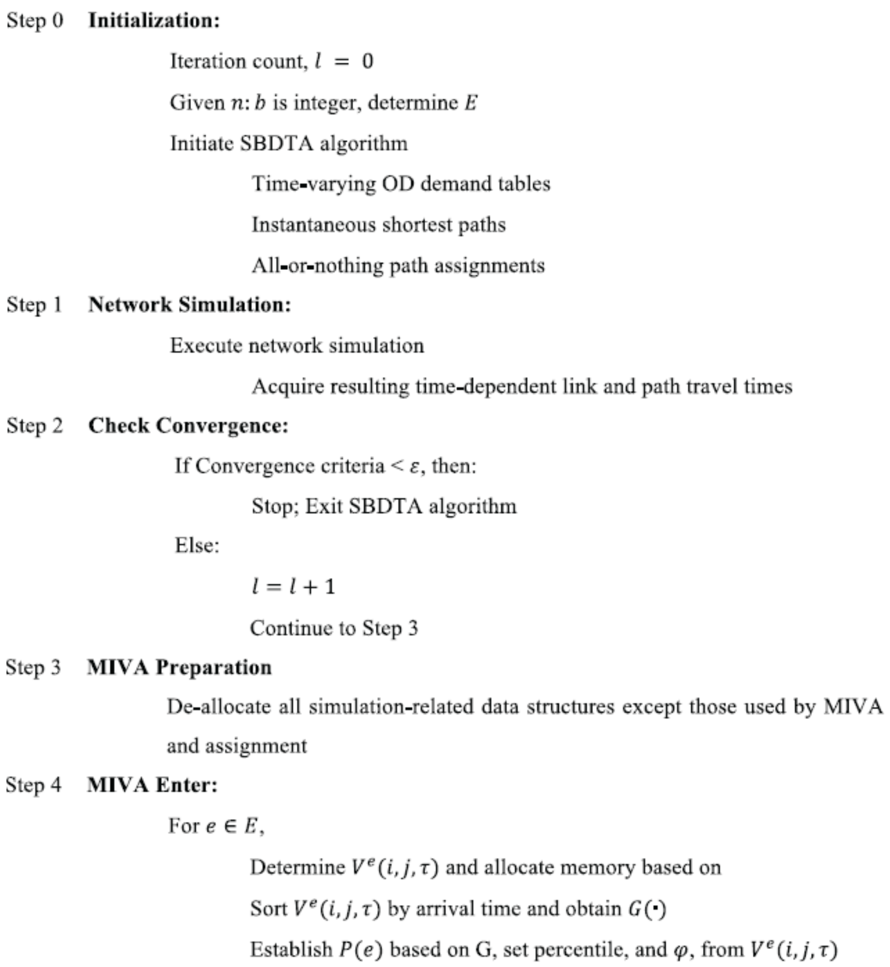
Projection Period

The projection period, (e^s), is defined as the set of assignment intervals for each epoch e^s. Let $P(e^s) = \{\tau^{(s-1)b+1}, \tau^{(s-1)b+2}, \ldots, \tau^{sb}, \tau^{sb+1}, \ldots, \tau^{sb+y}\}, \forall\, e^s \in E$ be the set of assignment intervals contained in the project period for epoch e^s. By definition, the start of the projection period is synchronized with each associated epoch. However, the projection period is extended beyond the end of the epoch by $\{\tau^{sb+1}, \ldots, \tau^{sb+y}\}$ as shown in Figure 3.4. This temporal extension is to allow the TDSP to solve for the later arrival times of those vehicles departing toward the end of the epoch.

It is intuitive to set the limit of the projection period based on the latest arrival time of all vehicles departing within the epoch, as beyond this limit link travel times and intersection delays would not be needed for the current epoch's TDSP calculation. However, binding the project period limit based on the latest arrival time may be too conservative and thus include too many additional assignment intervals if the latest arriving vehicle's travel time is likely to improve in the next iteration because it is assigned with a new path, and/or other vehicles assigned with a better path would also improve the overall traffic condition and improve all vehicles' travel times. With this being recognized, the length of the projection period can be defined as a ratio of total vehicles based on ranked experienced arrival times. This means vehicles belonging to $V^e(i, j, \tau)$ are sorted by increasing experienced arrival times. The projection period length is then defined based on a predefined ratio. For example, a 0.95 means that the end of the projection period is set at the 95th percentile of all arrival times. In other words, $P(e^s) = \{\tau^{(s-1)b+1}, \tau^{(s-1)b+2}, \ldots, \tau^{sb}, \tau^{sb+1}, \ldots, \tau^{sb+y}\}$, where $h \cdot \tau^{sb+y} \geq G(\varphi)$, where G is the increasing arrival time profile and φ is the predefined ratio, $0.0 \leq \varphi \leq 1.0$. One should also expect that $P(e^s)$ may vary due to different levels of congestion experienced by vehicles in different epochs. Vehicles departing in an epoch corresponding to off-peak hours would have a shorter projection period than those corresponding to peak hours due to more severe congestion during peak hours even though the epoch lengths are identical.

After solving the TDSP, the path becomes available over the domain $P(e^s) = \{\tau^{(s-1)b+1}, \tau^{(s-1)b+2}, \ldots, \tau^{sb}, \tau^{sb+1}, \ldots, \tau^{sb+y}\}$. The computational requirement to maintain $V^e(i, j, \tau)$ in memory rather than for V is the major advantage of the MIVA computational scheme. Memory is allocated only to the TDSP of size $P(e)$ rather than the entire simulation period. For the given $V^e(i, j, \tau)$ set, the path set adjustment (i.e., the DTA solution algorithm procedure) is performed, which is bounded by a given e. Once path adjustment is completed $\forall v \in V^e(i, j, \tau)$, the traffic data and TDSP calculations for the current epoch e are de-allocated from memory, and the MIVA scheme continues on to the next epoch.

Figure 3.5 describes the MIVA algorithmic structure within the simulation-based DTA (SBDTA) framework.



```
Step 0  Initialization:
            Iteration count, $l = 0$
            Given $n$: $b$ is integer, determine $E$
            Initiate SBDTA algorithm
                Time-varying OD demand tables
                Instantaneous shortest paths
                All-or-nothing path assignments
Step 1  Network Simulation:
            Execute network simulation
                Acquire resulting time-dependent link and path travel times
Step 2  Check Convergence:
            If Convergence criteria < $\varepsilon$, then:
                Stop; Exit SBDTA algorithm
            Else:
                $l = l + 1$
                Continue to Step 3
Step 3  MIVA Preparation
            De-allocate all simulation-related data structures except those used by MIVA
            and assignment
Step 4  MIVA Enter:
            For $e \in E$,
                Determine $V^e(i, j, \tau)$ and allocate memory based on
                Sort $V^e(i, j, \tau)$ by arrival time and obtain $G(\cdot)$
                Establish $P(e)$ based on G, set percentile, and $\varphi$, from $V^e(i, j, \tau)$
```

Source: Nava and Chiu 2012.

Figure 3.5. The pseudo code of the MIVA scheme.

CHAPTER 4

FAST-TrIPs Overview

For modeling the transit component, FAST-TrIPs is divided into two main submodules: transit assignment and simulation. Above all, the transit assignment submodule plays the role of passenger assignment for given OD pairs. For assigning transit passengers for the OD pairs, a trip-based shortest path model (Noh et al. 2011; Khani et al. 2012a; Khani et al. 2012b; Noh et al. 2012a; Noh et al. 2012b) is used by searching for a feasible path on each OD pair. The assigned passengers, including their paths, are given to and simulated through the transit simulation submodule in FAST-TrIPs. During the simulation, experienced arrival and departure times of transit vehicles are used to simulate boarding and alighting of passengers, considering transfers and other components (e.g., walking and waiting). Each passenger's trajectory (i.e., experienced path) is recorded, and dwell time for each transit route is calculated as a function of the boardings and alightings at each stop. Results of the simulation are used in the next iteration of auto-transit vehicle simulation and are also fed back to the activity-based model in the next global iteration for updating the demand.

FAST-TrIPs has an intermodal functionality, which is embedded in the two submodules mentioned above. It is capable of assigning and simulating the intermodal passengers in a mixed environment, modeling these movements for auto and transit passengers. The intermodal model consists of a park-and-ride assignment model for individual tours, a transit assignment and transit simulation model for the transit portion of the tour, and an interface with DynusT for the auto assignment and simulation.

FAST-TrIPs in the Integrated Model

In the big picture, in terms of integration, FAST-TrIPs interfaces with DynusT as well as connecting with the DaySim ABM. Before a discussion of the FAST-TrIPs model in detail, the overall functional location of the FAST-TrIPs tool in the integrated architecture should be noted. The high-level modular architecture for transit in the C10B project is given in Figure 1.1; Figure 4.1 briefly depicts the FAST-TrIPs operation.

To operate the FAST-TrIPs module, or DTA module in a higher level, several critical inputs to FAST-TrIPs must be made. To support the transit and intermodal travel within DynusT, FAST-TrIPs reads as input the database of traveler tours and trips from DaySim. This processing of the trip and tour rosters allows FAST-TrIPs to identify both the transit trips with walk access and egress and the transit trips with auto access or egress. The latter are intermodal trips in the sense that they require simulation of both modes of travel (transit and auto) within FAST-TrIPs and DynusT.

Specifically, for the purposes of FAST-TrIPs, the output from DaySim includes information for each trip segment and subsequent activity for each individual. It contains origin, destination, and travel mode, including differentiation for different transit access modes, and time at which travel begins, for trips leaving from the primary tour destination, or time at which travel ends, for trips destined to the primary tour destination.

This last choice, of departure time or arrival time, makes use of DaySim's protocol that trips going toward the primary tour destination are constrained to arrive by the start of the activity at that destination. Also, trips departing from the primary tour destination cannot depart before the scheduled end of the activity at the primary tour destination.

Passengers are assigned to a path in the transit network according to the protocol discussed later. In addition to the assignment of transit trips, FAST-TrIPs also performs an intermodal assignment, for which an estimate of auto travel times is necessary. This allows FAST-TrIPs to choose the optimal combination of an auto segment and a transit segment for an intermodal trip. The auto travel times in this assignment are obtained from either a time-dependent auto shortest path or the most recent auto skims of the network loading

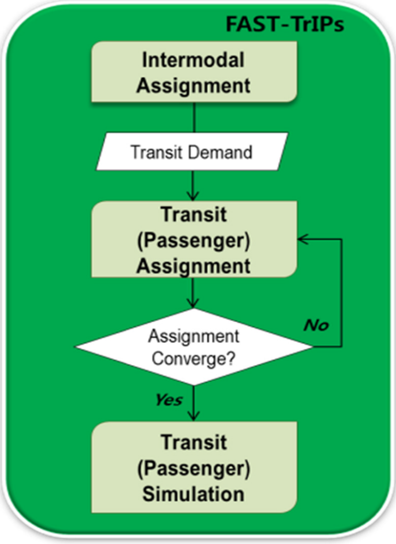


Figure 4.1. FAST-TrIPs algorithmic structure.

and simulation in DynusT. This is indicated on the right side of Figure 1.1, DynusT–FAST-TrIPs integration framework.

Once the intermodal assignment has been performed, FAST-TrIPs provides DynusT with the auto portion of any intermodal trip, allowing the simulation of the auto trip through either

- The assignment of an OD (a park-and-ride lot), and time of arrival at the park-and-ride lot, for trips transferring to transit headed to the primary tour destination; or
- The assignment of an origin (park-and-ride lot), destination, and time of departure from the park-and-ride lot, for trips returning from the primary tour destination.

The auto portion of these intermodal trips is simulated in DynusT. In addition, the trip and tour records from DaySim are updated to enforce sequential trip behavior: In the simulation, transit trips cannot begin before the auto actually arrives at the park-and-ride lot, and vice versa.

Of course, in parallel, DynusT reads the same trip and tour rosters from the intermodal assignment and determines the assignment of pure auto trips to associated paths and travel times in the network. Especially for intermodal cases, intermodal assignment will create the auto and transit part of these intermodal trips as deciding the optimal park-and-ride lot. The auto portion of the intermodal assignment is added in the DynusT simulation, and the transit portions are loaded to the transit assignment in FAST-TrIPs. These assigned auto trips from the intermodal demand are then simulated in DynusT.

As a second input shown in Figure 1.1, FAST-TrIPs uses Google's GTFS files. The GTFS files currently allow a transit agency to provide its routes and schedules to Google Maps. However, this same route and schedule data are often made publicly available by transit agencies, allowing others to develop applications using these data. The Sacramento Regional Transit District is one of the many transit agencies providing GTFS data to Google and to the public.

The GTFS files contain the geographic representation of routes and stops (typically in geographic information system, or GIS, shape files). The data also contain either the formal schedule of service (in the case of GTFS), or the frequency information (in the case of traditional line files) associated with each transit route and direction. The GTFS data or line files are converted into route networks that are compatible with the DynusT road network. This process is partly automated, using existing shape files for the road and transit networks, but considerable manual processing may be necessary to adjust the network to ensure that road segments are consistent and that transit stop locations are placed at appropriate locations in the road network. Finally, the schedule (the so-called stop-times in GTFS) also serves as input to the transit assignment. The transit network and service data should ideally be based on the GTFS. This is useful because the actual service schedule and individual stops can be modeled explicitly. This provides a more dynamic modeling of the transit passenger behavior than a traditional four-step model. The GTFS data can be used to represent the base year, perhaps by making some manual adjustments to the existing (2010 for the C10B implementation) GTFS schedule data to make it comparable to the base year.

The detail by which transit routes are defined in a schedule-based format such as GTFS may pose a problem when dealing with future-year forecasts. A transit network for the future will require designating routes, stops, and schedules. This may be easily adapted from existing schedules if only more modest changes are envisioned. However, for more significant changes in the transit network, this could require significant effort to develop appropriate GTFS data. Whether to develop this GTFS input or simply to use a future line file (from a four-step model) would be a decision likely made jointly by the local MPO (like SACOG) or the local transit agency (like Sacramento Regional Transit).

For the auto and transit vehicle simulation, as shown at the bottom of Figure 1.1, the auto, transit, and intermodal assignments are fed directly into a network loading and simulation, using DynusT's simulation capability. The simulation of vehicle movements in the network, at a fine level of temporal and spatial resolution, is handled directly within DynusT; this simulation includes specific movements of both auto and transit vehicles in the network. There is appropriate logic in DynusT for the management of mixed vehicle types in the traffic stream, as explained earlier, with each vehicle type having its own particular speed–density characteristics. The simulation also allows for the appropriate movements of transit vehicles at stops, including the following: a vehicle pulls out of the traffic stream at bus pullouts but remains in the traffic stream for curbside stops; dwell times allow for passengers to alight and to board, depending on passenger type (pedestrian, bicycle, wheelchair); a bus may bypass a stop if no passengers want to board or alight (so-called hail stop operation); and the vehicle may hold until the scheduled departure time if a vehicle is running ahead of schedule ("running hot").

The simulation of passenger movements in the transit network, however, is handled directly within FAST-TrIPs. The interaction between DynusT and FAST-TrIPs is through the link travel times that DynusT provides to FAST-TrIPs and the bus arrival, dwell, and departure times that FAST-TrIPs provides to DynusT. At the beginning of the simulation, FAST-TrIPs provides an estimate of the dwell times for each transit vehicle at each stop. During the simulation in DynusT, the transit vehicle dwells at the stop for this period of time, and then, if holding is needed, the vehicle is held until the scheduled departure time. The final simulation outputs from DynusT include the actual vehicle arrival and departure times from each stop. These outputs, in turn, can be used in the assignment to adjust passengers' path choices, to reflect issues with service reliability, or to confirm vehicle adherence to the schedule.

The assignment and simulation of transit vehicles and passengers is described in the following sections.

Passenger Assignment

Transit assignment is the process that determines a path, or set of connecting subpaths, for a passenger to get from origin to destination. DaySim determines the origins and destinations for the travelers and feeds this information into the assignment. FAST-TrIPs handles transit passenger and intermodal passenger assignment, and DynusT handles auto assignment.

From the output of DaySim, intermodal and transit trips are read directly by FAST-TrIPs. The required inputs for transit assignment in FAST-TrIPs include the passenger origin, the passenger destination, and the time of departure at the origin or the time of arrival at the destination. FAST-TrIPs has a transit shortest path algorithm, called trip-based shortest path (TBSP), which finds the path from the origin to the destination of the passengers.

Algorithmically, transit networks have an important feature in the types of nodes. That is, in transit networks, rerouting (i.e., a transfer to another route) can be done only at certain stops. For example, when a passenger is on board, he or she may not consider alighting from a vehicle to transfer to another route at every stop. More particularly, it does not happen where there is no other route available at a stop or where other stops are not within walking distance from that stop. Therefore, one may consider any variation in the path at transfer stops only. This property leads to the generation of a hierarchical transit network with transfer stops at the higher level, where the non-transfer stops in the path search algorithm may be disregarded. In this case, instead of having multiple stops and multiple links between two transfer stops, the transfer stops can be connected by a single transit trip, with associated departure and arrival times. However, the non-transfer stops are maintained in the network, as they can be used for access and egress points.

In the network preparation phase, transfer links are generated between a pair of stops if

1. The distance between the stops is less than a certain value (e.g., 0.25 mile), and
2. There is at least one route that serves one of the stops but not the other.

After generating transfer links, transfer stops are defined to construct the network hierarchy. A transfer stop is defined as a stop from which the passenger has the option of transferring to another route. With this definition, a stop is defined as a transfer stop if it is located on more than one route, or it has at least one inbound or outbound transfer link.

Using this simple model, the hierarchical transit network, which can be very useful in transit path algorithms, is generated.

A new network representation that is suitable for modeling schedule-based transit systems was used. This network structure is called trip based and is defined by a graph $G(N, P, T)$ where N is the set of nodes (or stops), P is the set of transit trips where each trip belongs to a route r in R (set of transit routes), and T is the set of transfer links. For each trip there is a list $S(p)$, which contains the stops served by the trip as well as the associated arrival and/or departure time of the transit vehicle at that stop. Also, for each stop, there is a set $A(n)$ containing the trips serving the stop. The main advantage of the trip-based network representation over node- or link-based structures is that the connection between two stops can be established by a single trip p if they are located on the same trip, while in both node-based and link-based networks the connection between any two stops is made using a sequence of links.

Transit Trip-Based Shortest Path

The trip-based network has the advantages of stop connectivity, dynamic representation of service, and hierarchical structure of transfer stops. On the basis of these properties, together with the data availability through GTFS and the behavior of transit users, a TBSP is developed. TBSP is a labeling algorithm in the schedule-based transit systems, exploiting the trip-based network format. Therefore, it has the advantage of processing a subset of stops and finds the shortest path in a shorter amount of time compared to traditional shortest path algorithms. The general form of the algorithm is shown in Figure 4.2; the form can be either label-setting or label-correcting, but the label-correcting form is used for the application in this project.

The variables in Figure 4.2. TBSP algorithm are defined as follows:

PAT_i = Preferred arrival time to stop *i*;
seq_{ip} = Sequence number of stop *i* on trip *p*;
d_{pi} = Departure time of trip *p* at stop *i* (usually the same as arrival in printed schedules and GTFS data);
v_{ijp} = In-vehicle time from stop *i* to stop *j* using trip *p*;
t_{ij} = Transfer time from stop *i* to stop *j* (typically equal to the walking time between two stops);
a_i = Arrival/departure time label of stop *i*;
w_{ip} = Waiting time at stop *i* for trip *p*, equal to the difference between the departure time of trip *p* at stop *i* and the arrival time label of stop *i*;
l_i = Label of stop *i*, equal to the travel time (or cost) from the origin stop to stop *i* in forward algorithms, and the travel time (or cost) from stop *i* to the destination stop in backward algorithms;
p_i = Predecessor stop of *i*;
m_i = Mode (trip number or transfer link) used to reach stop *i* in forward algorithms, or to leave stop *i* in backward algorithms;
c_{pi} = Utility (a function of travel time or cost) of trip *p* at stop *i* to reach the destination;
$T(i)$ = Set of transfer links at stop *i*;
$R(i)$ = Set of routes at stop *i*;
$p(i)$ = Set of trips at stop *i;* and
SE = Scan eligible list, containing the stops with temporary labels.

The TBSP algorithm in Figure 4.2 is a forward-labeling algorithm starting from the origin with τ as the planned departure time (PDT). With very few modifications, a backward algorithm can be developed with search from the destination in a backward direction. In this backward case, a preferred arrival time (PAT) is used for the destination. In the proposed forward algorithm, the label of each stop is the earliest arrival time.

In general, a generalized cost function can be used to generate different variations of the shortest path algorithm. In this case, different weights are applied to different components of the trip. This approach is used to model the inconvenience of transfers and waiting times compared with in-vehicle time. Assuming the weight α_k is applied to the *k*th element of the passenger trip, the least-cost path algorithm is developed based on the shortest path algorithm with changes as shown below:

13- If $(l_i + \alpha_t . t_{ij} < l_j)$:

14- $l_j = l_i + \alpha_t . t_{ij}$, $a_j = a_i + t_{ij}$, $p_j = i$, $m_j =$ "T"

19- If $l_i + \alpha_w . w_{ip} + \alpha_v . v_{ijp} < l_j$:

20- $l_j = l_i + \alpha_w . w_{ip} + \alpha_v . v_{ijp}$, $a_j = d_{jp}$, $p_j = i$, $m, = p$

```
1-   Initialization:
2-   Get the origin (O) and the departure time ($\tau$)
3-   $i=O$, $l_i=0$, $p_i=\Phi$, $a_i=\tau$, $m_i=\Phi$, $SE=\{i\}$
4-   $l_j=\infty$, $p_j=-$, $a_j=\infty$, $m_i=\Phi$;  $j\neq i$
5-   Termination Criterion:
6-   If $SE=\Phi$, stop!
7-   Stop Selection:
8-   Select $i=Argmin_j \{l_j | j \;\; SE\}$, if Label Setting
9-   Select i=The first stop in SE, if Label Correcting
10-  $SE=SE\backslash\{i\}$
11-  Updating the labels:
12-  $t \;\; T(i)$:
13-          If $(l_i+t_{ij}<l_j)$:
14-                  $l_j=l_i+t_{ij}$, $a_j=a_i+t_{ij}$, $p_j=i$, $m_j=$ "T"
15-                  $SE=SE \;\; \{j\}$
16-  $r \; R(i)$:
17-          Select $p=Argmin_q \{w_{iq}|d_{qi}\geq a_i\}$
18-          $j \; S(p)$ with $seq_j>seq_i$:
19-                  If $l_i+w_{ip}+v_{ijp}<l_j$:
20-                          $l_j= l_i+w_{ip}+v_{ijp}$, $a_j=d_{jp}$, $p_j=i$, $m_j=p$
21-                          If $j \; N_t$:
22-                                  $SE=SE \;\; \{j\}$
23-                                  Break the loop!
```

Figure 4.2. TBSP algorithm.

The result of the TBSP is a shortest path tree from the origin to all destinations. So the path to a specific destination will be found by tracking the predecessors from the destination stop to the origin stop. The path is then attached to the passenger and is passed to the simulation model.

Passenger Simulation

The passenger simulation model is a high-resolution model, capable of simulating the path taken by individuals in the transit and intermodal networks. The main inputs are the paths generated in the transit assignment submodule. In fact, there are three categories of data inputs to the passenger simulation:

- Transit network, including stops, routes, and schedule;
- Transit vehicle simulation results, including the actual arrival/departure of transit vehicles at each stop; and
- Passengers, including information about each passenger and the passenger's assigned path.

There are two main modules to capture the behavior of passengers and their interactions with transit vehicles. The first module captures access, egress, and transfer behavior of passengers. In the same way the simulation captures the movement of a passenger from his alighting stop to either his destination or the next boarding stop (in case of a transfer). The detailed information of the passenger's trip is recorded. The second module takes care of the boarding and alighting of passengers whenever a transit vehicle arrives to a stop. Therefore, an event-based simulation is used for this part, when an event is the simulated arrival of a transit vehicle to a transit stop. By considering factors such as the number of boarding and alighting passengers and type of transit vehicle, a dwell time value is calculated for the transit vehicle at each stop. For each transit vehicle, based on the type of route, a capacity is assumed. All of the information regarding the boarding, alighting, and passenger load of the vehicles is written in the output files and can be used in the next iteration. The model also has post-processing functions for calculating skim tables, summary statistics, and convergence measures.

Passenger Movements in Simulation

The primary engine that performs simulations is the simulation engine within DynusT, which handles auto and transit vehicle movements. However, FAST-TrIPs handles passenger movements within transit networks.

During the simulation itself, FAST-TrIPs and DynusT's mesoscopic simulator do not communicate directly. Rather, the simulation of the passenger movements is done through post-processing of the vehicle trajectories from DynusT. This design decision was made so as to maintain the computational efficiencies of DynusT yet still have a means of determining passenger movements. This passenger simulation process is illustrated in Figure 4.3.

As described previously, DynusT mesoscopic simulator handles all vehicle movements, including autos and transit vehicles. Among many other outputs, DynusT generates vehicle trajectories for the transit vehicles that are simulated, and it also records the times that autos depart their origins and arrive at their destinations. These outputs, shown on the left side of Figure 4.3, serve as the critical inputs to the transit and intermodal travel simulation in FAST-TrIPs.

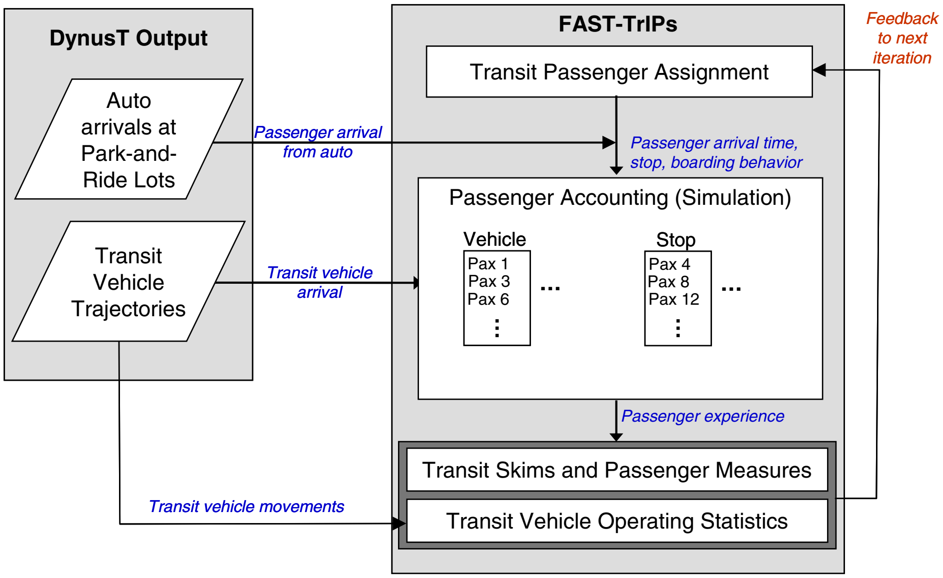


Figure 4.3. Transit passenger simulation in FAST-TrIPs.

In FAST-TrIPs, these data are used in a simulation of transit passenger travel. This simulation likewise follows a high temporal resolution, using a 1-second interval. With this as the simulation clock, the simulation methods in FAST-TrIPs handle the bookkeeping associated with the persons on board each vehicle and at the transit stops. This is done through a set of lists keeping track of passengers on a vehicle and passengers in a stop at any point in time.

The movement of passengers in each time interval follows the following transit assignment logic:

- A passenger who arrives at a stop during that time interval, from a given access mode (walk or auto access), is loaded into the queue at the stop. Note that passengers using auto access are not generated until the DynusT outputs indicate that the auto has arrived at the park-and-ride lot.
- A transit vehicle that arrives at a stop during that time interval is processed, in the sense that alighting passengers are removed from the vehicle and boarding passengers are added to the vehicle. The first method processes the passengers on the vehicle *k*. Those passengers on board whose final stop is *s* will be taken off this list and processed as an alighting passenger. Statistics for the alighting passenger (access time, egress time, in-vehicle time, transfer time, number of transfers, etc.) are written out to a file to compute experienced transit skims and summary statistics for the transit mode. A second method processes the list of passengers in the queue at the stop *s*. Those passengers desiring to board this vehicle *k* are then placed on board the vehicle, in a first-in–first-out basis, until either all passengers desiring to board have been admitted on board, or the vehicle capacity is reached. In the latter case, passengers are denied boarding due to the capacity constraints.
- An alighting passenger is sent to the destination, by a given egress mode, or to a transfer stop.

As these movements occur, FAST-TrIPs keeps track of the time associated with access, waiting, on board, transferring, and more in order to generate passenger statistics. Once passengers arrive at their destinations, their travel statistics are accumulated, and the passengers' experienced skims from the transit network are also generated. These skims and transit passenger summary statistics follow the structure of similar outputs for vehicle trips from DynusT. These in turn can be used to affect the transit passenger assignment in the next iteration. In addition, transit vehicle operating statistics (on-time performance, travel times, hours in operation, etc.) can also be derived from the trajectories as outputs from the DynusT simulation.

In the DynusT–FAST-TrIPs integration model, three types of convergence apply. DynusT and FAST-TrIPs, have their own individual convergence methods that use a relative gap measure to assess user equilibrium. DynusT uses the simulation-based relative gap measure. The convergence of FAST-TrIPs is estimated by the number of passengers who are denied from boarding due to capacity constraints. Finally, for the combined DynusT–FAST-TrIPs model, a relative gap measure comparing dwell time changes from one iteration to the next is applied.

Intermodal Assignment and Simulation

The intermodal assignment and simulation model in FAST-TrIPs models passenger tours with a combination of auto and transit modes; in other words, passengers who drive from their origin to access transit routes have the opportunity to select their transit path among a big set of transit options. Travelers are constrained to park their cars at the transit stations, something that limits their feasible choices to park-and-ride facilities only. Furthermore, they have to return to the same facility to pick up their car in the return portion of the tour. These constraints make the intermodal assignment a difficult problem containing different choice problems.

In the SHRP 2 C10B implementation, the intermodal problem has been decomposed to a tour-based park-and-ride choice (called the intermodal assignment model) and transit/auto path choice model (similar to auto or transit passengers). Given a passenger tour, including the activities and time windows between the activities, the intermodal assignment model finds the optimal park-and-ride location considering the travel cost in the whole tour. More specifically, the park-and-ride location has to

1. Be reachable in the time window between two consecutive activities, and
2. Have the minimum tour travel cost compared with other park-and-rides.

The algorithm that solves the intermodal problem is a combination of a shortest path in the auto network using the PDT from origin to all park-and-rides and a backward transit shortest path to the destination using the PAT. This gives the travel cost of the bimodal trip from the origin to the destination. With the similar concept, travel cost for the second half of the tour is calculated and the optimal park-and-ride is assigned to the tour with a specific arrival time to make the mode transfer. Then the trips with auto or transit mode are extracted and fed to the transit assignment model in FAST-TrIPs or DynusT. To make the algorithm computationally efficient, a TBSP is used for the transit network that takes into account the service schedule in GTFS.

In the transit assignment, intermodal passenger trips are modeled similarly to other passenger trips with the difference that either the origin or the destination is a park-and-ride location. The assignment model generates the path for passengers, and the passengers are simulated at the same time with other passengers. One important thing to be considered in simulating intermodal passengers is the actual arrival to the park-and-ride and the transfer to transit mode. Because the first part of an intermodal trip is traveled by auto, an auto trip is simulated in DynusT considering the roadway congestion, and the actual arrival of the passenger to the transit station is recorded. This time is used for the simulation of the transit part, and the feasibility of transit path is verified before starting the simulation. Finally, FAST-TrIPs produces the outputs of the intermodal tours in separate files from transit and sends them back to the demand model.

References

Balakrishna, R., H. N. Koutsopoulos, and M. Ben-Akiva. 2005. *Calibration and Validation of Dynamic Traffic Assignment Systems.* Elsevier, New York.

Chang, E., and A. Ziliaskopoulos. 2004. Formulation of Analytical Time-Varying Intermodal Person Trip Assignment Model. *Transportation Research Record: Journal of the Transportation Research Board, No. 1882,* Transportation Research Board of the National Academies, Washington, D.C. pp. 1–9.

Chiu, Y.-C., J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks. 2011. *Dynamic Traffic Assignment: A Primer.* Transportation Research Board of the National Academies, Washington, D.C. http://onlinepubs.trb.org/onlinepubs/circulars/ec153.pdf.

Chiu, Y.-C., L. Zhou, and H. Song. 2010. Development and Calibration of the Anisotropic Mesoscopic Simulation Model for Uninterrupted Flow Facilities. *Transportation Research Part B,* 44: 152–174.

Dafermos, S., and F. Sparrow. 1969. The Traffic Assignment Problem for a General Network. *Journal of Research of the National Bureau of Standards,* 73B: 91–118.

Florian, M., and S. Nguyen. 1974. A Method for Computing Network Equilibrium with Elastic Demands. *Transportation Science,* 8: 321–332.

Huang, H., and W. Lam. 2002. Modeling and Solving the Dynamic User Equilibrium Route and Departure Time Choice Problem in Network with Queues. *Transportation Research Part B,* 36: 253–273.

Jayakrishnan, R., H. Mahmassani, and T. Hu. 1994. An Evaluation Tool for Advanced Traffic Information and Management Systems in Urban Networks. *Transportation Research Part C,* (2C(3)): 129–147.

Khani, A., M. Hickman, and H. Noh. 2012a. Trip-Based Path Algorithms Using Transit Network Hierarchy (TBPATH). Presented at 4th International Symposium on Dynamic Traffic Assignment, Martha's Vineyard, Mass.

Khani, A., S. Lee, M. Hickman, H. Noh, and N. Nassir. 2012b. Intermodal Shortest- and Optimal-Path Algorithm Using a Transit Trip-Based Shortest Path. Presented at 91st Annual Meeting of Transportation Research Board, Washington, D.C.

Lu, C.-C. 2007. Multi-Criterion Dynamic Traffic Assignment Models and Algorithms for Road Pricing Applications with Heterogeneous Users. PhD dissertation, University of Maryland.

Lu, C.-C., H. Mahmassani, and X. Zhou. 2008a. A Bi-Criterion Dynamic User Equilibrium Traffic Assignment Model and Solution Algorithm for Evaluating Dynamic Road Pricing Strategies. *Transportation Research Part C: Emerging Technologies,* 16(4): 371.

Lu, C.-C., H. Mahmassani, and X. Zhou. 2008b. Equivalent Gap Function-Based Reformulation and Solution Algorithm for the Dynamic User Equilibrium Problem. *Transportation Research Part B: Methodological.* In Press, Corrected Proof.

Nava, E., and Y.-C. Chiu. 2012. A Temporal Domain Decomposition Algorithmic Scheme for Large-Scale Dynamic Traffic Assignment. *International Journal of Transportation Science and Technology,* 1(1): 1–24.

Noh, H., A. Khani, Y.-C. Chiu, M. Hickman, and B. Bustillos. 2012a. Integration of Dynamic Traffic Assignment with Schedule-Based Transit Assignment and Simulation of Congested Multimodal Networks. Presented at 4th International Symposium on Dynamic Traffic Assignment, Martha's Vineyard, Mass.

Noh, H., A. Khani, M. Hickman, and Y.-C. Chiu. 2011. Integrated Dynamic Traffic and Transit Assignment Considering Traffic Congestion and a Schedule-Based Transit Network. Presented at 13th TRB Transportation Planning Applications Conference, Reno, Nev.

Noh, H., A. Khani, M. Hickman, Y. Tian, Y.-C. Chiu, and J. Villalobos. 2012b. Integrated Vehicle and Passenger Simulation Model Considering a Multimodal Transportation Network. Presented at 4th TRB Conference on Innovations in Travel Modeling. Tampa, Fla.

Sheffi, Y. 1985. *Urban Transportation Networks.* Prentice-Hall, Inc., Englewood Cliffs, N.J. http://web.mit.edu/sheffi/www/selectedMedia/sheffi_urban_trans_networks.pdf.

Smith, M. 1993. A New Dynamic Traffic Model and the Existence and Calculation of Dynamic User Equilibrium on Congested Capacity-Constrained Road Networks. *Transportation Research Part B: Methodological,* 27(1): 49–63.

Smith, M., and M. Wisten. 1995. A Continuous Day-to-Day Traffic Assignment Model and the Existence of a Continuous Dynamic User Equilibrium. *Annals of Operations Research,* 60(1): 59–79.

Szeto, W., and H. Lo. 2006. Dynamic Traffic Assignment: Properties and Extensions. *Transportation,* 2(1): 31–52.

APPENDIX A

DynusT–FAST-TrIPs Integration Model Execution

Starting with DST_FT_module.py, successive sequential steps to run the DynusT–FAST-TrIPs integration model are processed as follows:

Step 0: ft_BST.py -> ft_Intermodal.exe
Step 1: DynusTx64.exe
Step 2: FAST_TrIPs.py -> ft_Assignment.exe -> ft_Simulation.exe

If not converged yet, go to Step 1. Else, stop.

Step 0: ft_BST.py

To initiate the main DynusT–FAST-TrIPs integration model, DST_FT_module starts with the boot-strapping process, ft_BST.py, which first reads veh_sout.dbf as input files and converts it to text file, to make the file reading more efficient in DynusT and FAST-TrIPs. Then, it runs intermodal assignment (ft_intermodal.exe) which

- Separates auto demand and transit demand (modes 6 and 3, respectively) and writes them down in separate files for DynusT and FAST-TrIPs use; and
- Runs an initial intermodal tour assignment (with static auto travel times and printed transit schedule) and splits intermodal tours into transit and auto trip segments, and appends them into the auto and transit demand files.

Therefore, after running the python script, demand files are generated for auto (including auto trips and auto part of intermodal tours) and transit (transit trips and transit part of intermodal tours), which are used for assignment in DynusT and FAST-TrIPs, respectively.

Step 1: DynusTx64.exe

The DynusT module mainly assigns and simulates auto demand, including ABM trips, exogenous trips, and auto trip in intermodal trip. For integrating with FAST-TrIPs module, critical transit vehicle arrival information (AltTime_transit.dat) is created.

Step 2: FAST_TrIPs.py

FAST-TrIPs module consists of three components: transit assignment, passenger simulation, and intermodal assignment. These components are designed in separate files (ft_Assignment.exe, ft_Simulation.exe, and ft_Intermodal.exe) and are executed sequentially as explained below:

- *Transit Assignment (ft_Assignment.exe)*

This component assigns passenger trips along the searched transit path, which typically shows boarding and alighting stops, used transit trips (or vehicles), and access and egress times. Transit paths are assigned to transit passengers so that they can be simulated in the next component (passenger simulation).

- *Passenger Simulation (ft_Simulation.exe)*

Using the passenger assignment information (ft_passenger.dat) and simulated transit vehicles' information (AltTime_transit.dat), passengers are simulated on the transit network from their origin to destination, so that passenger flow on each transit trip (vehicle) is determined. After running this component, transit-related outputs are generated and dwell times (in TransitDwellTime.dat) are fed back into DynusT module in an iterative process. Convergence of the system is also tested in this component and the result is printed in ft_convergence.dat.

- *Intermodal Tour Assignment (ft_Intermodal.exe)*

The intermodal tour assignment is the same function used in ft_BST.py, which assigns intermodal tours to optimal park-and-rides and splits transit and auto segments for the next iteration. The difference is that in this step, the results of simulation (time-dependent travel time and experienced transit vehicle trajectories) are used in the algorithm. In the latest iteration of DynusT–FAST-TrIPs, the intermodal assignment is not required and transit passenger simulation is the last step of the model.

APPENDIX B

Working with Data Set

Create Working Folder

For the integrated DynusT–FAST-TrIPs model, each executable file is assigned to a specific folder as shown in Figure B.1. Basically, the data folder (given any desired name) includes all the main functions (executable files) of the model and the data set input files except those related to demand from the activity-based model. More specifically, the python master code, called DST_FT_module.exe is in this folder along with DynusT executable and FAST-TrIPs executable files (ft_Intermodal.exe, ft_Assignment.exe, and ft_Simulation.exe). In addition, two FAST-TrIPs python scripts (ft_BST.py and FAST_TrIPs.py) that call the main functions also need to be placed in this folder.

A folder called VehicleDemandGen should be created in the main folder, in which all other python scripts are placed to generate demand files from the activity-based model. The two subfolders in VehicleDemandGen folder are assigned for the input files and output files.

The main input file for the DynusT–FAST-TrIPs model is veh_sout.dbf, which is located in the Input Files subfolder, and all the demand-related files are placed in the same folder after generation in the bootstrapping step. Under the same token, DynusT output files need to be copied to the Output Files subfolder. Output Files folder is temporarily activated in terms of file transfer. The outputs (i.e., vehicle.dat and path.dat) generated by DaySim2DynusT-MAINprog-S.py in VehicleDemandGen are temporarily stored in the Output Files folder, then transferred to the working folder and used by DynusT during runs.

Aside from hosting the demand file, the Input Files subfolder also hosts information mapping files for DynusT and DaySim attributes as well as defined corridors or paths of interest and several DynusT network files. The mapping files are of importance for the following reasons: they define DynusT to DaySim zone, node, and parcel equivalents and coordinate information is used for distance calculation. The defined corridor data provide for easy access information for the purpose of post-processing simulation data. Lastly, the DynusT origin and destination network files will provide information for trips.

As mentioned previously, the python files in the VehicleDemandGen folder read and translate DaySim trip information and prepare it as DynusT input files. To execute such processes, intermediate mapping files (found in Input Files subfolder) for DynusT and DaySim attributes must also be developed. In essence, this exercise ensures trips will start, originate, and terminate at the DaySim designated times and locations but within DynusT. Within the DynusT simulation a person making a trip will be assigned a vehicle; once this trip reaches its destination the assigned vehicle will exit the simulation. A person making multiple trips within the 24-h simulation period will be assigned into several vehicles; only one of those vehicles will be simulated at once.

Table B.1 illustrates typical files (not an exhaustive list) for DynusT and FAST-TrIPs.

Rancho Cordova Network Example

Rancho Cordova, California, is an eastern suburb of Sacramento, California. The DynusT Rancho Cordova network was created in late August in 2010 and is a “subarea network cut” from the regional Sacramento model. The DynusT network is shown in Table B.2.

Network

This network is composed of 121 zones, 452 nodes, and 864 links. Like the downtown model, the Rancho Cordova network must first be cleaned/fine-tuned in order to bring it from its original state, a macroscopic model network, to mesoscopic model compatibility.

For Rancho Cordova area, the transit network consists of 163 stops, 170 bi-directional transfer links, and 205 trips for

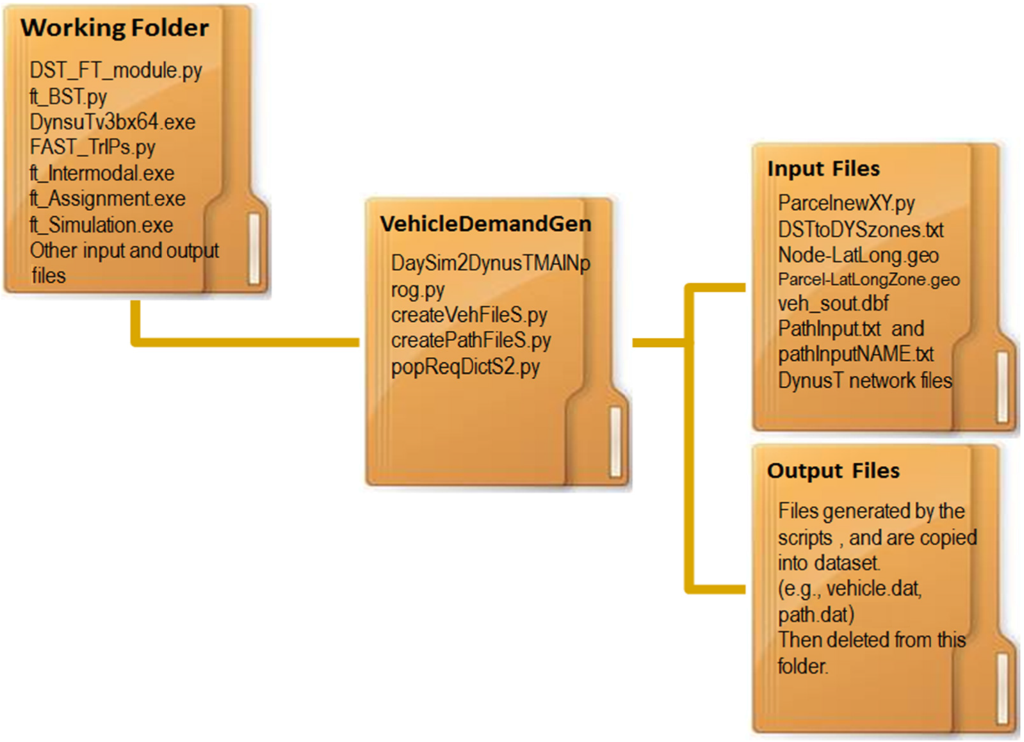


Figure B.1. Folder configuration.

Table B.1. Input File List in Working Folder

DynusT	Network Data (network.dat) Movement Data (movement.dat) Vehicle Generation Data (origin.dat) Vehicle Exit Data (destination.dat) Signal Data (control.dat) Simulation and Assignment Configuration Data (system.dat) Model Scenario Data (scenario.dat) Work Zone Data (workzone.dat) Incident Data (incident.dat) Vehicle Trip Schedule Data (vehicle.dat) Vehicle Route Data (path.dat) Advanced Parameter Data (parameter.dat)
FAST-TrIPs	ft_input_stops.dat ft_input_routes.dat ft_input_trips.dat ft_input_stopsTimes.dat ft_input_accessLinks.dat ft_input_transfers.dat ft_input_transitVehicles.dat ft_input_park-n-rides.dat ft_demand_auto.dat ft_demand_transit.dat ft_passengers.dat
DynusT–FAST-TrIPs	networks.dat AltTime_transit.dat AltTime_interModal.dat travelerVehicleMapping.dat travelerVehIntermodal.dat TransitRouteSchedule.dat TransitDwellTime.dat

five bus routes. The Sacramento light rail transit (Gold line) is excluded in this data set. The transit data are shown in Figure B.3.

Demand

Exogenous trip tables (airport, external, thru auto and truck, and 2- and 3-axle commercial vehicles) from the travel demand model were converted to DynusT origin–destination (OD) demand table input. The conversion produced two temporal OD files, demand.dat and demand_truck.dat, with defined DynusT formats (explained in demand.dat description memo). The following table summarizes the generated demands.

Table B.2. Input File List in "VehicleDemandGen–Input Files"

	Node-LatLong.geo Parcel-LatLongZone.geo parcelMOD.txt DSTtoDYSzones.txt PathInput.txt pathInputNAME.txt xy.dat network.dat origin.dat destination.dat veh_sout.dbf

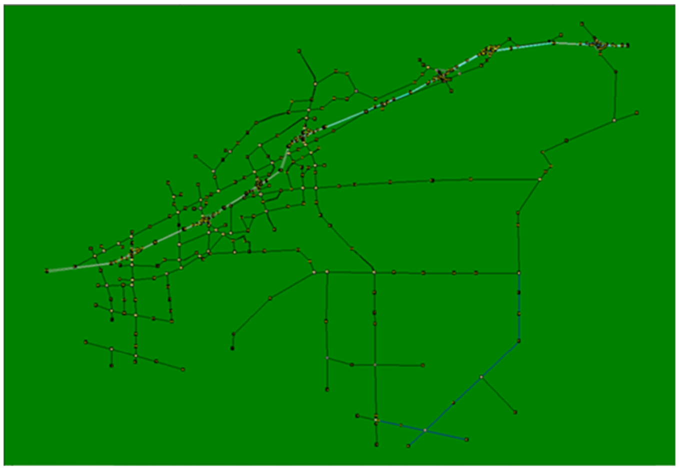


Figure B.2. Rancho Cordova transportation network in DynusT.

Similarly, trip information from the activity-based model (DaySim) is converted, generating a vehicle.dat file with defined DynusT formats (explained in vehicle.dat description memo). The following table summarizes the generated demands. As previously mentioned and observed, the DaySim and DynusT values are different, due to the filtering trips modes that were not solely auto (e.g., "shared ride" or "drive alone"), such as "drive-transit-walk," "bike," and "walk," along with other modes.

Table B.3. Rancho Cordova Exogenous Demand Statistics

File Name	Trips	Total Trips
demand.dat	• airport: 12,262 • thru auto: 220,489 • external: 202,886	435,637
demand_truck.dat	• 2- and 3-axle: 83,388 • thru truck: 5,878	89,266

Transit Route

The operating transit system in Rancho Cordova is a part of Sacramento Regional Transit. There are five bus routes and a light rail line that cover Rancho Cordova area and the connection between other parts of Sacramento regional area (i.e., some of the routes are extended out of Rancho Cordova). To construct the transit network, General Transit Feed Specification (GTFS) data of Sacramento Regional Transit are used, and the required data are extracted for Rancho Cordova. The data were accessed in November 2010 through http://www.gtfs-data-exchange.com, and 2011 transit service was used for the network preparation.

The transit routes serving Rancho Cordova are bus routes 21, 28, 72, 74, and 75, and a part of light-rail transit route number 507 (Gold line). By mapping the GTFS routes using GIS and comparing the data with the auto network and the

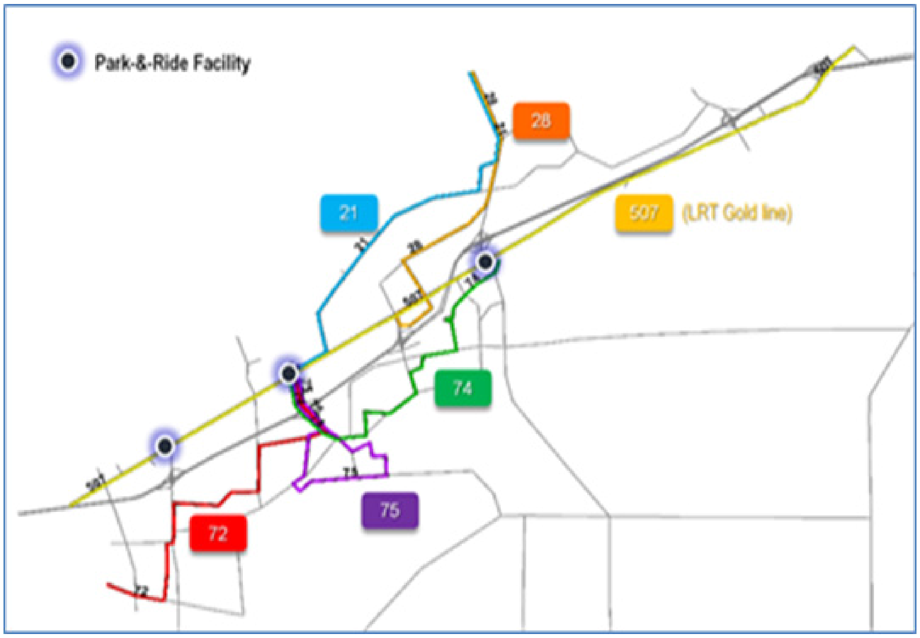


Figure B.3. Rancho Cordova transit network.

Table B.4. Rancho Cordova Trip Roster Demand Statistics

File Name	Total Trips
DaySim trip roster	140,099
vehicle.dat	117,795

area boundary, the transit routes were cut to keep the segments inside the area only. In the same way, transit stops inside the area were selected. Since the transit network is used both in DynusT (for transit vehicle simulation in congested traffic network), and FAST-TrIPs (for assignment and simulation of transit passengers), transit routes had to be coded into the DynusT network. In other words, the path for each route (including its directions and variations) were found and stops were mapped into DynusT links, so that DynusT could read the transit network, generate buses like a type of vehicle, and simulate them in the model. This process was done in a semiautomatic way, so that paths were found by mapping GTFS shape files (containing geographic information about routes and stops) into the auto network and finding the best path that can represent the transit route. Obviously, the results were not perfect and many checks and corrections were done manually to accommodate transit routes in the DTA network (e.g., adding some missing links in the DynusT model on where transit vehicles travel). At the same time, the GTFS data, including stops, routes, trips, and stop-times (detailed schedule of transit vehicle trips) were prepared in appropriate formats to be read by FAST-TrIPs.

TRB OVERSIGHT COMMITTEE FOR THE STRATEGIC HIGHWAY RESEARCH PROGRAM 2*

CHAIR: **Kirk T. Steudle,** *Director, Michigan Department of Transportation*

MEMBERS

H. Norman Abramson, *Executive Vice President (retired), Southwest Research Institute*
Alan C. Clark, *MPO Director, Houston–Galveston Area Council*
Frank L. Danchetz, *Vice President, ARCADIS-US, Inc.*
Malcolm Dougherty, *Director, California Department of Transportation*
Stanley Gee, *Executive Deputy Commissioner, New York State Department of Transportation*
Mary L. Klein, *President and CEO, NatureServe*
Michael P. Lewis, *Director, Rhode Island Department of Transportation*
John R. Njord, *Executive Director (retired), Utah Department of Transportation*
Charles F. Potts, *Chief Executive Officer, Heritage Construction and Materials*
Ananth K. Prasad, *Secretary, Florida Department of Transportation*
Gerald M. Ross, *Chief Engineer (retired), Georgia Department of Transportation*
George E. Schoener, *Executive Director, I-95 Corridor Coalition*
Kumares C. Sinha, *Olson Distinguished Professor of Civil Engineering, Purdue University*
Paul Trombino III, *Director, Iowa Department of Transportation*

EX OFFICIO MEMBERS

Victor M. Mendez, *Administrator, Federal Highway Administration*
David L. Strickland, *Administrator, National Highway Transportation Safety Administration*
Frederick "Bud" Wright, *Executive Director, American Association of State Highway and Transportation Officials*

LIAISONS

Ken Jacoby, *Communications and Outreach Team Director, Office of Corporate Research, Technology, and Innovation Management, Federal Highway Administration*
Tony Kane, *Director, Engineering and Technical Services, American Association of State Highway and Transportation Officials*
Jeffrey F. Paniati, *Executive Director, Federal Highway Administration*
John Pearson, *Program Director, Council of Deputy Ministers Responsible for Transportation and Highway Safety, Canada*
Michael F. Trentacoste, *Associate Administrator, Research, Development, and Technology, Federal Highway Administration*

---

* Membership as of July 2014.

CAPACITY TECHNICAL COORDINATING COMMITTEE*

CHAIR: **Mark Van Port Fleet,** *Director, Bureau of Highway Development, Michigan Department of Transportation*

MEMBERS

Kome Ajise, *Program Manager, Public-Private Partnership Program, California Department of Transportation (Caltrans)*
Michael Bruff, *Manager, Transportation Planning Branch, North Carolina Department of Transportation*
Jacquelyn D. Grimshaw, *Vice President for Policy, Center for Neighborhood Technology*
Kris Hoellen, *Director, Conservation Leadership Network, The Conservation Fund*
Carolyn H. Ismart, *Florida Department of Transportation (retired)*
Randy Iwasaki, *Executive Director, Contra Costa Transportation Authority*
Thomas J. Kane, *Thomas J. Kane Consulting*
Keith L. Killough, *Assistant Director, Travel Demand Modeling and Analysis, Multimodal Planning Division, Arizona Department of Transportation*
T. Keith Lawton, *Principal, Keith Lawton Consulting, Inc.*
Debra Nelson, *Strategic Policy Advisor, New York State Department of Transportation*
Bob Romig, *State Transportation Development Administrator, Florida Department of Transportation*
Joseph L. Schofer, *Professor of Civil Engineering and Environmental Engineering and Associate Dean, McCormick School of Engineering and Applied Science, Northwestern University*
Barry Seymour, *Executive Director, Delaware Valley Regional Planning Commission*
Brian J. Smith, *Washington State Department of Transportation*
John V. Thomas, *Office of Policy, Economics, and Innovation, Environmental Protection Agency*
Gary Toth, *Director, Project for Public Spaces*
Jeff Welch, *Director, Knoxville Regional Transportation Planning Organization*
Doug Woodall, *State Director, Turnpike Planning and Development, Texas Turnpike Authority Division, Texas Department of Transportation*

AASHTO LIAISON

Janet P. Oakley, *Director, Policy and Government Relations, American Association of State Highway and Transportation Officials*
Matthew Hardy, *Program Director, Planning and Policy, American Association of State Highway and Transportation Officials*

FHWA LIAISONS

James Cheatham, *Director, Office of Planning, Office of Planning HEPP-1, Federal Highway Administration*
Gary A. Jensen, *Team Leader, Byways, TCSP & Delta Programs, Office of Human Environment HEPH-30, Federal Highway Administration*
Spencer Stevens, *Community Planner, Office of Planning Oversight and Stewardship, Federal Highway Administration*

---

* Membership as of July 2014.

Related SHRP 2 Research

Improving Our Understanding of How Highway Congestion and Pricing Affect Travel Demand (C04)

Incorporating Greenhouse Gas Emissions into the Collaborative Decision-Making Process (C09)

Dynamic, Integrated Model System: Jacksonville-Area Application (C10A)

Transferability of Activity-Based Model Parameters (C10A)

Activity-Based Travel Demand Models: A Primer (C46)