Enhancements to Transportation Analysis and Simulation System (TRANSIMS)

Mansoureh Jeihani Koohbanani

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Civil and Environmental Engineering

Graduate Committee:
Antoine G. Hobeika, Ph.D., Co-Chair
Hanif D. Sherali, Ph.D., Co-Chair
Dusan Teodorovic, Ph.D.
John Collura, Ph.D.
Antonio Trani, Ph.D.
Hojong Baik, Ph.D.

November 30, 2004
Blacksburg, Virginia

Keywords: Traffic Assignment, Dynamic User Equilibrium, Convex-Simplex, TRANSIMS, Emissions Estimation

Enhancements to Transportation Analysis and Simulation System (TRANSIMS)

Mansoureh Jeihani Koohbanani

Abstract

Urban travel demand forecasting and traffic assignment models are important tools in developing transportation plans for a metropolitan area. These tools provide forecasts of urban travel patterns under various transportation supply conditions. The predicted travel patterns then provide useful information in planning the transportation system. Traffic assignment is the assignment of origin-destination flows to transportation routes, based on factors that affect route choice.

The urban travel demand models, developed in the mid 1950s, provided accurate and precise answers to the planning and policy issues being addressed at that time, which mainly revolved around expansion of the highway system to meet the rapidly growing travel demand. However, the urban transportation planning and analysis have undergone changes over the years, while the structure of the travel demand models has remained largely unchanged except for the introduction of disaggregate choice models beginning in the mid-1970s. Legislative and analytical requirements that exceed the capabilities of these models and methodologies have driven new technical approaches such as TRANSIMS.

The Transportation Analysis and Simulation System, or TRANSIMS, is an integrated system of travel forecasting models designed to give transportation planners accurate, and complete information on traffic impacts, congestion, and pollution. It was developed by the Los Alamos National Laboratory to address new transportation and air quality forecasting procedures required by the Clean Air Act, the Intermodal Surface Transportation Efficiency Act, and other regulations.

TRANSIMS includes six different modules: Population Synthesizer, Activity Generator, Route Planner, Microsimulator, Emissions Estimator, and Feedback. This package has been under development since 1994 and needs significant improvements within some of its modules. This dissertation enhances the interaction between the Route Planner and the Microsimulator modules to improve the dynamic traffic assignment process in TRANSIMS, and the Emissions Estimator module.

The traditional trip assignment is static in nature. Static assignment models assume that traffic is in a steady-state, link volumes are time invariant, the time to traverse a link depends only on the number of vehicles on that link, and that the vehicle queues are stacked vertically and do not traverse to the upstream links in the network. Thus, a matrix of steady-state origin-destination (O-D) trip rates is assigned simultaneously to shortest paths from each origin to a destination. To address the static traffic assignment problems, dynamic traffic assignment models are proposed. In dynamic traffic assignment models, the demand is allowed to be time varying so that the number of vehicles passing through a link and the corresponding link travel times become time-dependent. In contrast with the static case, the dynamic traffic assignment problem is still relatively unexplored and a precise formulation is not clearly established. Most models in the literature do not present a solution algorithm and among the presented methods, most of them are not suitable for large-scale networks. Among the suggested solution methodologies that claim to be applicable to large-scale networks, very few methods have been actually tested on such large-scale networks. Furthermore, most of these models have stability and convergence problem.

A solution methodology for computing dynamic user equilibria in large-scale transportation networks is presented in this dissertation. This method, which stems from the convex simplex method, routes one traveler at a time on the network and updates the link volumes and link travel times after each routing. Therefore, this method is dynamic in two aspects: it is time-dependent, and it routes travelers based on the most updated link travel times. To guarantee finite termination, an additional stopping criterion is adopted.

The proposed model is implemented within TRANSIMS, the Transportation Analysis and Simulation System, and is applied to a large-scale network. The current user equilibrium computation in TRANSIMS involves simply an iterative process between the Route Planner and the MicroSimulator modules. In the first run, the Route Planner uses free-flow speeds on each link to estimate the travel time to find the shortest paths, which is not accurate because there exist other vehicles on the link and so, the speed is not simply equal to the free-flow speed. Therefore, some paths might not be the shortest paths due to congestion. The Microsimulator produces the new travel times based on accurate vehicle speeds. These travel times are fed back to the Route Planner, and the new routes are determined as the shortest paths for selected travelers. This procedure does not necessarily lead to a user equilibrium solution. The existing problems in this procedure are addressed in our proposed algorithm as follows.

TRANSIMS routes one person at a time but does not update link travel times. Therefore, each traveler is routed regardless of other travelers on the network. The current stopping criterion is based only on visualization and the procedure might oscillate. Also, the current traffic assignment spends a huge amount of time by iterating frequently between the Route Planner and the Microsimulator. For example in the Portland study, 21 iterations between the Route Planner and the Microsimulator were performed that took 33:29 hours using three 500-MHZ CPUs (parallel processing). These difficulties are addressed by distributing travelers on the network in a better manner from the beginning in the Route Planner to avoid the frequent iterations between the Route Planner and the Microsimulator that are required to redistribute them. By updating the

link travel times using a link performance function, a near-equilibrium is obtained only in one iteration. Travelers are distributed in the network with regard to other travelers in the first iteration; therefore, there is no need to redistribute them using the time-consuming iterative process. To avoid problems caused by link performance function usage, an iterative procedure between the current Route Planner and the Microsimulator is performed and a user equilibrium is found after a few iterations. Using an appropriate descent-based stopping criterion, the finite termination of the procedure is guaranteed. An illustration using real-data pertaining to the transportation network of Portland, Oregon, is presented along with comparative analyses.

TRANSIMS framework contains a vehicle emissions module that estimates tailpipe emissions for light and heavy duty vehicles and evaporative emissions for light duty vehicles. It uses as inputs the emissions arrays obtained the Comprehensive Modal Emissions Model (CMEM). This dissertation describes and validates the framework of TRANSIMS for modeling vehicle emissions. Specifically, it identifies an error in the model calculations and enhances the emission modeling formulation. Furthermore, the dissertation compares the TRANSIMS emission estimates to on-road emission-measurements and other state-of-the-art emission models including the VT-Micro and CMEM models.

To my husband, Soheil

whose love and constant support made this possible

Acknowledgements

I am indebted to my advisors Dr. Antoine Hobeika and Dr. Hanif Sherali for their guidance, support, encouragement, and patient through my Ph.D. study. I have been fortunate in benefiting from the friendship, enlightening advice, and exemplary work ethic and commitment to quality and excellence of them.

I am grateful to Dr. Hesham Rakha for his educational discussions, advice, and experience during my dissertation. I would like to thank Dr. Dusan Teodorovic, Dr. John Collura, Dr. Antonio Trani, and Dr. Hojong Baik, members of my dissertation committee, for their participation and useful suggestions.

Finally, my heartfelt gratitude to my husband, Soheil, whose support and encouragement made this effort possible. This research work is a tribute to his constant support and love throughout my endeavors. I would also like to thank my beloved brothers and my parents whose firm belief in the value of good education leaded me into this way.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Problem Definition

The classical or traditional urban travel demand modeling process that has been used over the past forty years is called the "four-step" process. This was developed in the mid 1950s and has been used widely for decades for urban transportation planning. The planning purpose addressed at that time was an expansion of the highway system to meet rapidly growing travel demands and these models were accurate enough for that purpose. The computing facilities were inadequate at that time to make a more precise or a disaggregated model.

Since the mid 1980s, the transportation policy has not been to construct additional major highways; rather the policy has been to increase the efficiency of the current transportation network, which requires the models to be more precise and more responsive. Thus, transportation planning techniques suffered a revolution in 1980s and only small improvements were performed in the traditional models. By the end of 1980s, great advances in computing technology encouraged modelers to make basic improvements to the current models or develop new generation models. All the issues required to improve the current four-step model are not addressed by these models. Therefore, a new framework is needed to derive a good transportation demand forecasting model.

The Transportation Analysis and Simulation System, TRANSIMS, which served the foregoing purposes, is part of the Travel Model Improvement Program (TMIP) sponsored by the U.S. Department of Transportation, the Environmental Protection Agency (EPA), and the Department of Energy. TMIP is a multi-year, multi-agency program designed to improve analytical tools as well as the integration of these tools into the transportation planning process. TMIP was established in order to increase the ability of existing travel forecasting procedures to respond to emerging policy and technology issues. It also redesigns the travel forecasting process to reflect changes in behavior, and responds to greater information needs placed on the forecasting process, and takes advantage of changes in data collection technology. In addition, TMIP integrates the forecasting techniques and the decision making processes to provide a better understanding of effects of transportation decisions. TMIP has focused on both short-term and long-term

improvements to the models and planning procedures. The short-term improvements concentrated on changes to the existing four-step modeling process. TRANSIMS is the long-term effort to redesign the modeling process from the ground up.

The objective of TRANSIMS is to develop technologies that can be used by transportation planners in any urban environment. TRANSIMS will offer transportation planning agencies increased policy sensitivity, more detailed vehicle-emission estimates, and improved analysis and visualization capabilities. The primary philosophy in TRANSIMS is to simulate travel in a study area with a rather fine temporal and spatial resolution. Other research and development efforts have also come to the conclusion that the next generation of urban travel models should be based on microanalytic simulation and they should employ the activity-based approach to travel demand modeling.

TRANSIMS solves most of the existing problems in the current travel demand forecasting models. It is a microscopic transportation modeling framework which consists of several modules: Population Synthesizer, Activity Generator, Route Planner, Microsimulator, Feedback, and Emissions Estimator.

TRANSIMS not only synthesizes individuals and their activities, but also creates the transportation network and individual trip plans to carry out these activities. TRANSIMS builds a model of households and activity demands. The model forecasts how changes in transportation policy or infrastructure might affect activities and trips. TRANSIMS tries to capture every important interaction between travel subsystems, such as an individual's activity plans and congestion on the transportation system. For instance, when a trip takes too long, people find other routes, change from car to bus or vice versa, leave at different times, or decide not to do a given activity at a given location.

The modules developed for TRANSIMS contain many significant advances beyond the four-step models. For example, the simulation module in TRANSIMS observes the movements of individuals and vehicles second by second throughout the entire day rather than just the total travel for various periods. This movement represents realistic traffic dynamics produced from interactions of individual vehicles. The regional microsimulation uses vehicle interactions to produce operating speeds, intersection

operations, and vehicle operating conditions for each vehicle in the system instead of deterministic equations. The microsimulation process in TRANSIMS permits a very detailed analysis of traffic operations on the transportation network. This capability could be used to evaluate improvements such as traffic signal plans and ramp metering.

To identify traffic congestion and emission concentration, every vehicle in the study area should be monitored. Existing emission models use average speed estimates from the current models. The resulting emission estimates are insensitive to traffic conditions and lack precision. In contrast, TRANSIMS estimates vehicle pollutant emissions and fuel consumption based on the operation of individual vehicles as they interact in the roadway traffic.

Underlying the development of the different modules in TRANSIMS has been an effort to tightly couple the functions and data flow among the four modules in the existing four-step modeling process. While it is currently possible to add a microsimulation or an air quality emission calculation into current forecasting models, such capabilities require considerable post-processing. Also, feedback mechanisms and the careful scrutiny of policy or infrastructure actions are currently difficult to achieve.

TRANSIMS has been under development since 1994 and it is in the deployment stage. The first-level deployment activities are to complete the core of TRANSIMS, to develop user-friendly software, and to support local agencies in the implementation process. The goal of this effort is to operate TRANSIMS in several locations in order to identify problems so that these problems can be fixed. Another goal of the deployment program is to begin training consultants and transportation planning agency staff to gain experience with operating TRANSIMS. The Los Alamos National Laboratory has completed experimental transportation studies at Albuquerque and Dallas and has completed a microsimulation of auto traffic patterns in 25 square miles of Dallas that represent about 200,000 vehicles over a five-hour period. A model for the Portland transportation system using TRANSIMS is under development and is scheduled for completion by end of 2004.

Although TRANSIMS solves most of the problems in the existing four-step models, it still needs some improvements within some of the modules. In this dissertation, some of the problems that exist in the current version of TRANSIMS regarding the Emissions Estimator module as well as the interaction between the Route Planner and the Microsimulator are addressed and improvements are developed.

1.2 Objectives

The fundamental objective of this dissertation is to address some of the problems in the current version of TRANSIMS. The focus is in the Route Planner-Microsimulator feedback and the Emissions Estimator module. Besides proposing improvements in these modules of TRANSIMS, this dissertation computes dynamic user equilibria for large-scale transportation networks, which can be implemented within any traffic assignment software package. A new heuristic method is presented to compute dynamic user equilibria and this method is implemented within TRANSIMS and is tested on a large-scale network.

The Route Planner module routes one traveler at a time on the transportation network but it is insensitive to each routing, because it does not update the link volumes and link travel times after each routing. This drawback is solved by updating the link volumes and link travel times after each loading. Since each person is routed based on the most updated travel times, travelers are distributed better in the network as compared with the current process.

Transportation activities contribute to carbon monoxide, excessive ozone, and particulate matter concentration in urban areas. The TRANSIMS Emissions Estimator module translates traveler behavior into consequent energy consumption and pollutant emissions of nitrogen oxides, hydrocarbons, carbon monoxide, and carbon dioxide. The Emissions Estimator produces estimates for tailpipe emissions from light-duty vehicles (LDVs), tailpipe emissions from heavy-duty vehicles (HDVs), and evaporative emissions as a function of vehicle fleet composition, fleet status, and fleet dynamics. Specifically, an error in the existing model calculations is identified and the emission modeling formulation is enhanced. Furthermore, this dissertation compares the TRANSIMS

emission estimates to on-road emission-measurements and other state-of-the-art emission models including the VT-Micro and CMEM models.

1.3 Organization of the Dissertation

The reminder of this dissertation is divided into five chapters. Chapter 2 reviews dynamic traffic assignment models in the planning context. It surveys the existing literature on dynamic user equilibrium models and discusses their shortcoming in the planning context for large-scale transportation networks.

Chapter 3 presents a comprehensive explanation of the Feedback module in TRANSIMS. This Feedback module uses an iteration script provided by the user to control the overall framework of TRANSIMS. A typical study using TRANSIMS involves repeated iterations between the different modules. One important example of this feedback is in solving the traffic assignment problem. The simplest version of this uses a loop between the Route Planner and the Traffic Microsimulator modules to determine the actual travel times from the Microsimulator and consequently, to feed back this information to the Route Planner to compute the shortest routes for the travelers in the network.

The dynamic traffic assignment model in TRANSIMS is compared to that in other simulation packages in Chapter 4. A solution algorithm to calculate dynamic user equilibria for large-scale transportation networks and an implementation of this algorithm within TRANSIMS is presented in Chapter 5. Chapter 6 describes our enhancement to the Emissions Estimator module of TRANSIMS as well as the validation test of the emission estimations in TRANSIMS using an on-road emissions measurement device. Finally, Chapter 7 concludes this dissertation.

Chapter 2

Background Review

2.1 Introduction

Increasing urban traffic congestion has motivated researchers to improve the performance of traffic networks. Drivers, who travel from their current locations to a desired destination at specific departure or desired arrival times, require a best routing for their trips. The travel time between an origin and a destination in an urban area is the result of many individuals' decisions. Travelers choose if and when to take a trip, which mode of transportation to use, where to go, and which way to get there. These decisions partly depend on the amount and location of congestion. Congestion at any point, however, depends on the amount of travel through that point. A driver's behavior is the result of a complex process involving human judgment, decision-making and learning in a dynamic environment.

The current chapter reviews the existing literature regarding two concepts: Traffic assignment and driver's behavior.

2.2 Review of Traffic Assignment Concepts and Models

This section deals with traffic assignment and equilibrium concepts. Traffic assignment is the assignment of origin-destination flows to transportation routes, based on factors that affect route choice. It is divided into three main categories: Static, quasi-dynamic, and dynamic traffic assignment, which are explained in the next three sub-sections. The existing literature dealing with dynamic equilibrium problems under system optimal and user equilibrium categories is reviewed in the dynamic traffic assignment section. In the fourth sub-section, we discuss different related mathematical formulations and in the fifth sub-section, we review the existing solution methodologies. The sixth sub-section contains a summary and conclusion.

2.2.1 Static Assignment Concepts

Static assignment models assume that link flows and link trip times remain constant over the planning horizon. Thus, a matrix of steady-state origin-destination (O-D) trip rates are assigned to the network links. Although static equilibrium models are adequate for long-term planning analyses, they fail to capture the essential features of traffic

congestion. The first mathematical programming formulation for the static user equilibrium problem was presented by Beckmann et al. (1956). In the static case, the link performance function is assumed to be positive, increasing, and convex. An unrealistic and restrictive assumption of these functions is that the travel time on each link is independent of flows on other links in the network. Static assignment models are inappropriate for real-time traffic control applications because congestion cannot be modeled adequately in static models.

2.2.2 Dynamic Assignment Models

Dynamic network assignment is a subject that has been recently considered by many researchers. The dynamic analysis of network flow patterns consists of three principal dimensions (Mahmassani, 1990):

(1) Time-dependent flow patterns within a given day: Almost all existing time-dependent traffic assignment models address single user class problems and can be classified into descriptive and normative models.

(2) Day-to-day dynamics of peak period traffic flow: This is concerned with the evaluation of time-dependent flow patterns from day to day. This approach is appropriate for situations where users adjust their departure times and route decisions from one day to the next and the system might not be in equilibrium. The stability of possible equilibrium under these conditions depends on the behavioral rules.

(3) Real-time dynamics of flow patterns resulting from real-time decisions of motorists: This focuses on flow patterns resulting from the real-time decisions of travelers in response to supplied real-time information and perceived traffic conditions.

There are two different equilibrium problems. The system optimal (SO) approach that minimizes the total system travel time over the planning horizon, and the user equilibrium problem (UE) that seeks time-dependent user path assignments that satisfy the extension of Wardropian user equilibrium conditions: No user can improve his/her experienced travel time by unilaterally switching routes for a given departure time. While

user equilibrium formulations are descriptive in nature, the system optimal objectives are normative. The system optimal problem objective function minimizes the total system travel time.

There are some other classifications of traffic assignment models such as:

- Stochastic/ Deterministic: The route choice behavior of travelers is assumed to have a random component in the stochastic models.
- Path-based / Link-Based
- Flow-based / vehicle-based: Flow-based models provide a snapshot of the network for each time interval (not a continuous record of network operation) while vehicle-based models require a starting point in time from which they proceed on a virtually continuous basis.

The operational definition of the user equilibrium is suggested by Wardrop (1952): For each OD pair, at an user equilibrium, the travel time on all used paths is equal, and less than or equal to the travel time that would be experienced by a single vehicle on any unused path. In the dynamic space, the departure time is added to the above definition.

Most existing time-dependent traffic assignment (TDTA) formulations assume convex, continuous, and non-decreasing link performance functions to represent link costs. This assumption simplifies the structure of the problem but ignores essential aspects of the time-dependent nature of the problem. To minimize system-wide travel delays, it may often be useful to favor certain traffic streams or movements over others, which is not probably acceptable. Some mathematical programming SO/UE papers use the concept of a concave exit function specified as an upper bound on the number of vehicles exiting a particular link in a given period to model link congestion (e.g. see Li et al.(2000)). The behavior of traffic on a link is first-in-first-out (FIFO) that creates a vexing difficulty in the solution of mathematical programming formulations. This problem does not arise in static assignment problems nor in TDTA with a single destination. However, in TDTA with multiple destinations, vehicles on different paths that share one or more links may violate the FIFO protocol. For example, the downstream arc along one path might be blocked, but not for another path. This problem occurs regardless of whether SO or UE models are being used. There are some techniques that

have been proposed to address the FIFO problem in the literature (e.g. Peeta and Mahmassani (1995)).

2.2.2.1 System Optimal Traffic Assignment

Most studies on system optimal traffic assignment can be accommodated in the following five classes of problems. (1) Optimizing the departure patterns in a commuting corridor with a single route and one bottleneck; (2) minimizing total system cost for deterministic time-dependent O-D flows from multiple origin to a single destination; (3) the joint assignments of peak-period users to departure times and routes in a corridor with a single destination; (4) minimizing total system cost for stochastic time-dependent O-D flows from multiple origins to a single destination, and (5) minimizing total system cost for deterministic time-dependent O-D flows from multiple origins to multiple destinations.

The first mathematical programming approach to the SO problem was developed by Merchant and Nemhauser (1978). Their model considered the assignment of a known time varying O-D trip pattern, and was formulated as a non-linear, non-convex, discrete time mathematical program.

Some researchers extended and generalized this work while others considered constrained optimal control theory in which the O-D trip rates are assumed to be known continuous functions of time. The link flows are also considered as continuous functions of time.

Friesz et al. (1989) analyzed the dynamic system optimization problem for networks in which network transportation costs are minimized over a planning horizon. Their model is a dynamic extension of static system and user optimization models for network traffic assignment. They also discussed optimal control formulations for the single destination case for the system optimal time-dependent traffic assignment (SOTDTA) problem. The flow patterns from a static system optimization process are shown to be equilibrated for all utilized paths connecting a given O-D pair. Although this result is not surprising, it provides a framework for dynamic user equilibria. They considered a fixed planning horizon of length T.

Some early researches such as Hendrickson and Kocur (1981), have addressed

optimal congestion toll pricing for a single O-D, single bottleneck case with time-varying flows that satisfy some system-wide objectives. The problem is to determine a time-varying toll pricing scheme so that the flow pattern resulting from drivers' decisions satisfies the controller's objective.

Birge and Ho (1993) extended the Merchant and Nemhauser model to the stochastic case by relaxing the assumption that the O-D choices are known. Here, O-D demands are assumed to be random variables. They developed a multistage stochastic mathematical programming formulation, which is non-linear and non-convex.

Wie et al. (1994) solved the discrete time dynamic system optimal traffic assignment model of Merchant and Nemhauser as a discrete time optimal control problem. The planning horizon was divided into a finite number of discrete time periods having a uniformly small length.

Mahmassani was the first researcher who used simulation in traffic assignment. Shen, Chen, and Mahmassani (1993) described a new simulator for laboratory studies of the dynamics of commuter behavior under real-time traffic information. They investigated the morning peak period commuters in congested traffic corridors. They performed three different experiments: (i) Pre-trip and en-route path selection only; (ii) pre-trip departure time and path choice and en-rote path selection, and (iii) pre-trip departure time and path choice, real-time departure time adjustments, and en-route path selection. In each experiment, each subject was asked to drive to the CBD. Each driver was provided with real-time traffic information before each trip. Then the driver determined his/her departure time and path decisions. These decisions was fed into the simulator. At each junction where the driver had the opportunity to switch to another route, he/she was given real-time traffic information and was asked to decide to continue the current route or to switch to another route. Feedback was supplied to the user at the end of the trip. The process was repeated until system convergence was achieved.

Peeta and Mahmassani (1995) formulated two dynamic models in which O-D choices were assumed to be known a priori: The system optimal time-dependent traffic assignment (SOTDTA) and the user equilibrium time-dependent traffic assignment (UETDTA) formulations. Peeta and Mahmassani used a path-based formulation rather than a link-based model. In the system optimal model, they used a non-linear mixed-

integer model that incorporates binary time-dependent link path incidence variables to relate the number of vehicles assigned to each path to those on each link. The difficulty in solving TDTA is that the time-dependent incidence variables are themselves a function of the assignment. There are no explicit constraints for ensuring the FIFO protocol or for holding of vehicles, but in the simulation-based solution algorithm used by them, the traffic simulator implicitly satisfies these constraints. The objective function is convex and/or differentiable whereas the constraint set is non-convex. Therefore, the SO assignment solution is obtained by assigning vehicles to the minimum time-dependent marginal travel time paths that are consistent with the first-order conditions. They also extended this approach to the time-dependent case using a Lagrangian strategy on the SO formulation. They presented a heuristic iterative procedure in which a special purpose simulation model, DYNASMART was used, to evaluate system performance and to represent the traffic interaction.

Sandholm (2002) stated that at equilibrium, each commuter chooses a path that minimizes his/her travel time, but does not consider the consequence of his/her choice on others. Thus, the total travel time experienced by all users at an equilibrium exceeds the total expected.

Dafermos and Sparrow (1969) proposed a condition on link costs, which ensures that equilibrium behavior generates a system optimum. Sandholm interpreted their condition as a global equality of cost elasticities. This condition implies that the costs associated with each link in the network must be equally sensitive to changes in the link flow. Sandholm then established a local condition on cost elasticities, which implies the optimality of an equilibrium behavior. The local equality of cost elastisities implies that if the cost elasticities of all links are equal at a system optimum, then this system optimum is an equilibrium solution. This condition for optimality only needs equal sensitivities of link costs at a single flow configuration.

Laborczi and Nowotny (2003) proposed a fast heuristic method for finding a dynamic system optimal in large-scale networks using a time-dependent form of the original Davidson link performance function developed by Akcelik.

2.2.2.2 User Equilibrium Traffic Assignment

Most studies define a dynamic user equilibrium (DUE) problem by extending Wardrop's UE conditions to the choice of departure time in addition to route. These models seek a time-dependent UE pattern covering a peak traffic period. The UE solution is obtained by equilibrating the travel time of simultaneous route choice and departure time choice decisions.

The traditional solution in an urban planning scenario is static deterministic assignment. It assumes a steady-state demand and allocates the resulting traffic streams in an optimal way, which minimizes some costs such as time for each individual user.

There are many possible dynamic generalizations of the static user optimal traffic assignment model. Most dynamic user equilibrium assignment models have a correspondence with the static user equilibrium. For stochastic assignment, the route choice behavior of individual travelers is assumed to have a random component.

Fisk (1980) formulated an optimization problem for which the resulting flow between each O-D pair is distributed among different paths according to a logit formula where link costs are determined as functions of the assigned flows. His model is a general approach for which the user optimized equilibrium model (Wardrop, 1952) and the Dial assignment model (Dial, 1971) are special cases, where the first stochastic route choice model was proposed by Dial.

Most of the analytical formulations consider an idealized commuting corridor with a single O-D pair connected by a single route with a single bottleneck. De Palma et al. (1983) formulated a stochastic user equilibrium by considering a random utility model for the departure time choice. Extensions to incorporate parallel routes were proposed by Mahmassani and Herman (1984), and further extended by Ben-Akiva et al. (1986) assuming elastic demands in a corridor framework with queuing bottlenecks.

Most studies of traffic equilibrium that proceeded Smith (1984) considered route choices and mode choices but not the departure time. Smith suggested that each commuter has a desired time for leaving the bottleneck. If the commuter leaves at that time, a penalty associated with the arrival time at work is minimized. If it is impossible for all commuters to leave the bottleneck at their desired times, a queue forms and each commuter tries to minimize the sum of his/her queuing delay at the bottleneck and also

the penalty associated with his/her actual time of leaving the bottleneck. Friesz et al. (1989) focused on route choices rather than departure time choices and formulated the model as a continuous time optimal control problem.

In all the above models, a generalization of the DUE to the network level is not possible. Mahmassani and Jayakrishnan (1991) proposed determining the time-dependent departure pattern that satisfies dynamic user equilibrium conditions. This extends the usual static UE to include the trip timing decisions. They computed a stochastic DUE in a corridor network where traffic performance is obtained using a traffic simulation model using the Method of Successive Average (MSA). Friesz et al. (1993) reviewed one class of models, the Boston traffic equilibrium, and presented a new class of models, path integrals, which is also a dynamic generalization of the static Wardropean user equilibrium. They developed a variational inequality formulation of the DUE path integrals. Wie et al. (1995) extended Friesz's work. They formulated the dynamic user equilibrium problem in terms of unit path cost functions as a variational inequality problem. The formulation is based on simultaneous route-departure choices by users in discrete time intervals. Their method needs a partial enumeration of paths and is therefore not applicable for large networks. Wie et al. (2002) expanded this work to preclude a path enumeration. They formulated elastic demands as a link-based non-linear complementary problem on congested traffic networks. They actually extended the path-based formulation used in Friesz et al. (1993). Their link-based model is developed in a discrete time space to avoid both the complexity of the mathematical analysis required to continuous-time models and also to take advantage of the simplicity of discrete-time numerical analysis. Ran and Boyce (1996) also presented a discrete variational inequality model for the ideal dynamic user equilibrium route choice.

Cantarella and Cascetta (1995) addressed the day-to-day dynamic modeling of transportation networks using two approaches: Deterministic and stochastic. In the stochastic approach, the distribution of the forecasted path costs around the expected value is considered through random residuals. Thus the forecasted path costs are modeled through deterministic variables. The number of commuters following each path was modeled as a discrete random vector. When O-D demand and capacity values are large enough, deterministic and stochastic descriptions of system evolution are similar and the

equilibrium flows become closer to the stochastic process model. Ran et al. (1993) presented a dynamic user optimal route choice model that considers the equilibration of instantaneous travel time. Mahmassani and Peeta (1993) expanded on the time-dependent savings of SO over UE. They suggested based on their experiments that the saving of SO over UE are not accumulated uniformly over time. They are dependent on network congestion conditions. The greatest saving occurs when the network is well-congested.

Peeta and Mahmassani (1995) developed a method for determining an approximate UE using a traffic simulator (DYNASMART) to update the traffic-related variables. Each simulation time interval was taken to be six seconds and it was assumed that traffic conditions do not change during this small period. They did not represent a complete mathematical formulation of this problem and hence did not find an exact equilibrium. They stated that simulation implicitly satisfies constraints to solve the traffic assignment problem. The traffic simulator circumvents the need for link performance functions and link exit functions, ensures FIFO, captures link interactions, and prevents holding of traffic. They also used a path-based assignment in which path flows are the decision variables. This approach is good from the ATIS point of view because the controller needs to provide paths to the users.

Raney and Nagel (2002) used TRANSIMS to simulate all Switzerland transportation network and find a user equilibrium.

2.2.3 Formulation Approaches

Current studies on dynamic traffic assignment can be grouped into mathematical programming, optimal control, and variational inequality approaches.

Mathematical Programming

In mathematical programming approaches, all the state and control variables are considered as decision variables and also, restrictions are modeled as algebraic constraints. Therefore, the large number of decision variables and constraints can make the computation very expensive when the number of links and the number of time periods increases.

A general optimization problem is to maximize or minimize an objective function, and in the case of a constrained problem, subject to a set of constraints.

$$\underset{x}{\text{Minimize}} \quad Z(x) \qquad (1)$$

$$\text{subject to} \quad x \in G.$$

Merchant and Nemhauser (1978) were the first to formulate the dynamic traffic assignment problem as a discrete time, non-linear, and non-convex problem based on system optimization. Carey (1986, 1987) proved that the Merchant-Nemhauser model satisfies the linear independence constraint qualification and modified their model as a convex program.

A typical system optimal formulation using mathematical programming is as follows (Peeta and Mahmassani, 1995).

$$\text{Minimize} \quad \sum_t \sum_i \sum_j \sum_k (r_{ijk}^t . T_{ijk}^t)$$

or

$$\text{Minimize} \quad \left[T\left(r_{ijk}^t, \quad \forall i, j, k, t \right) \right]$$

subject to

$$r_{ij}^t = \sum_k r_{ijk}^t, \qquad \forall i, j, t,$$

$$\sum_b d^{tb} = \sum_c m^{tc} + I_n^t - o_n^t, \qquad \forall t, n, b \in B(n), c \in c(n),$$

$$x^{ta} = x^{t-1a} + d^{t-1a} - m^{t-1a}, \qquad \forall t, a,$$

$$x^{ta} = \sum_k \sum_t \sum_i \sum_j (r_{ijk}^t . d_{ijk}^{tta}), \qquad \forall t, a,$$

$$T_{ijk}^t = \sum \sum \left[d_{ijk}^{tta} \Delta \right], \qquad \forall i, j, k, t,$$

$$d_{ijk}^{tta} = F\left[\left(r_{ijk}^{t}\right), \qquad \forall i, j, kt\right] \qquad \forall i, j, k, t, t, a,$$

$$d^{ta} = \sum_{k}\sum_{t}\sum_{i}\sum_{j} d_{ijk}^{tta}, \qquad \forall t, a,$$

$$m^{ta} = \sum_{k}\sum_{t}\sum_{i}\sum_{j} m_{ijk}^{tta}, \qquad \forall t, a,$$

$$I_{n}^{t} = \sum_{j} r_{nj}^{t}, \qquad \forall t, n \in I$$

$$o_{n}^{t} = \sum_{k}\sum_{t}\sum_{i}\sum_{c} m_{ink}^{ttc}, \qquad \forall t, n, \in J, c \in C(n),$$

$$t \le t,$$

$$d_{ijk}^{tta} = 0 \text{ or } 1, \qquad \forall i, j, k, t, t, a,$$

$$\text{all variables} \ge 0,$$

where,

i: Subscript for origin nodes,

j: Subscript for destination nodes,

n: Index for a node in the network,

a: Subscript for a link in the network,

k: Subscript for a path in the network,

t: Superscript denoting departure time interval,

t: Superscript denoting current time interval,

T': Total duration for which assignments are to be made,

? : Length of a time interval (equal to *T'/T*),

r_{ij}^{t} : Number of vehicles that depart from i to j in period t,

r_{ijk}^{t}: Number of vehicles that depart from i to j in period t along path k,

T_{ijk}^{t}: Experienced path travel time for vehicles going from i to j that are assigned to path k at time t,

d_{ijk}^{tta}: Time-dependent link-path incident indicator, equal to 1 if vehicles going from i to j assigned to path k at time t are on link a in period t, i.e.,

$[d_{ijk}^{tta} = 1,$ if r_{ijk}^{t} is on arc a during period t,

$= 0,$ if arc a does not belong to path k,

$= 0,$ if $t > t$,

$= 0,$ if r_{ijk}^{t} is not on arc a during period t],

x_{ijk}^{tta}: number of vehicles going from i to j along path k in period t that are on link a at the beginning of period t,

d_{ijk}^{tta}: number of vehicles going from i to j along path k in period t that enter link a in period t.

The constraints represent the conservation of O-D demand choices at the origin, conservation of vehicles at nodes, conservation of vehicles on links, the number of vehicles on a link in terms of path-vehicle assignments, definition of path travel times using the incidence variables, the definitional constraints for the number of vehicles entering and exiting links during various time intervals, the definitional constraints for the number of vehicles entering and exiting the network at a particular node and time interval, temporal correctness constraints that restrict the start (or departure) time constraint, restriction of the time-dependent incidence variables to take 0 or 1 values, and the non-negativity constraints.

The time-dependent path marginal travel time t_{ijk}^{t} is the effect of an additional vehicle on path k at time t on the system travel time. A path-based definition of the global path marginal t_{ijk}^{t} for a particular $i = m$, $j = w$, $k = p$ and $t = k$ is:

$$t_{mwp}^{k} = T_{mwp}^{k} + \sum_{t}\sum_{i}\sum_{j}\sum_{k} r_{ijk}^{t} \frac{\partial}{\partial r_{mwp}^{k}} (T_{ijk}^{t}).$$

The first-order conditions for the static SO problem state that at an optimal solution, the marginal travel times on all used paths for a given O-D pair are equal, and less than or equal to the marginal travel times on any unused routes. Therefore, the SO assignment solution is obtained by assigning vehicles to the minimum time-dependent marginal travel time paths that are consistent with the first-order conditions.

Variational Inequality Problem

The variational inequality problem (VI) is a general problem formulation that includes a set of mathematical problems such as non-linear equations, optimization problems, complementarity problems, and fixed-point problems. There is a vector of decision variables $x = (x_1, x_2, \ldots, x_n)$ and a vector of cost functions $f(x) = [f_1(x), f_2(x), \ldots,$

$f_n(x)$]. Define G as a given closed convex set of the decision variables x and f as a vector of given continuous functions in R^n. The infinite-dimensional VI problem is to determine a vector $x^* \in G \subset R^n$, such that $f[x^*].[x - x^*] \geq 0$ (2)"

In geometric terms, Equation (2) states that $f(x^*)$ is orthogonal to the feasible set G at the point x*.

Friesz et al. (1993) proposed a VI formulation of the Dynamic User Equilibrium with simultaneous path and departure time choices for the first time. Wie et al. (1995, 2002) extended Friesz's work. They formulated the dynamic user equilibrium problem in terms of unit path cost functions as a variational inequality problem. Ran and Boyce (1996) and Li et al. (2000) also presented a discrete variational inequality model for the dynamic user equilibrium route choice.

An example of a VI formulation is Ran and Boyce (1996)'s formulation :

$$\sum_{n=1}^{N}\sum_{rs}\sum_{a}\Omega_a^{rj*}(n)\left\{u_a^{rs}[n+p^{ri*}(n)]-u_a^{rs*}[n+p^{ri*}(n)]\right\}\geq 0$$

where, $\Omega_a^{rj*}(n)$ is the difference of the minimal travel time from r to j and the minimal travel time from r to j via minimal travel time route from r to i and link a for vehicles departing from origin at time t:

$$\Omega_a^{rj*} = p^{ri*} + t_a[t+p^{ri*}(t)] - p^{rj*}(t) \qquad \forall a, r; a=(i,j),$$

u_a^{rs} is the inflow into link a, and p_a^{rs} is the minimum actual route travel time over link a at time interval n from r to s. To solve this problem, they reformulated it as the following non-linear program (NLP), which is equivalent to the discrete VI under relaxation. The relaxation is:

- the actual travel time $t_a(n)$ in the link flow propagation constrains is fixed as $\bar{t}_a(n)$,
- minimal travel times $p^{ri*}(n)$ are fixed as $\bar{p}^{ri*}(n)$ and $p^{rj*}(n)$ are fixed as $\bar{p}^{rj*}(n)$.

The objective function of the NLP problem is as follows.

$$\underset{u,v,x,E}{\text{Minimize}} \quad Z = \sum_{k=1}^{K} \sum_{a} \left\{ \int_{0}^{u_a(k)} t_a[x_a(k), w, v_a(k)]dw + \sum_{r} u_a^r(k)[\bar{p}^{ri}(x_a^r) - \bar{p}^{rj}(x_a^r)] \right\}$$

subject to $\quad x_{ap}^{rs}(k+1) = x_{ap}^{rs}(k) + u_{ap}^{rs}(k) - v_{ap}^{rs}(k), \qquad \forall a, p, r, s; k = 1,...,K;$

$$E^{rs}(k=1) = E^{rs}(k) + \sum_{a \in B(s)} \sum_{p} v_{ap}^{rs}(k), \qquad \forall r; s \neq r; k = 1,...,K;$$

$$\sum_{a \in A(r)} \sum_{p} u_{ap}^{rs}(k) = f^{rs}(k), \qquad \forall r \neq s; k = 1,...,K;$$

$$\sum_{a \in B(j)} v_{ap}^{rs}(k) - \sum_{a \in A(j)} u_{ap}^{rs}(k) = 0, \qquad \forall j, p, r, s; j \neq r; k = 1,...,K;$$

$$x_{ap}^{rs}(k) = \sum \left\{ x_{bp}^{rs}[k + \bar{t}_a(k)] - x_{bp}^{rs}(k) \right\} + \left\{ E_p^{rs}[k + \bar{t}_a(k)] - E_p^{rs}(k) \right\}, \quad \forall a \in B(j); p, r, s; j \neq r; k = 1,...,K;$$

$$u_{ap}^{rs}(k) \geq 0, v_{ap}^{rs}(k) \geq 0, x_{ap}^{rs}(k+1) \geq 0, \qquad \forall a, p, r, s; k = 1,...,K;$$

$$E_p^{rs}(k+1) \geq 0, \qquad \forall p, r, s; k = 1,...,K;$$

$$E_p^{rs}(1) = 0, \qquad \forall p, r, s;$$

$$x_{ap}^{rs}(1) = 0, \qquad \forall a, p, r, s;$$

where, $t_a(n)$ is the actual travel time over link a at time interval n, $x_a(n)$ is the number of vehicles on link a at time interval n, $u_a(n)$ is inflow into link a during interval n, $v_a(n)$ is the exit flow into link a during interval n, $f_p^{rs}(n)$ is inflow into route p during interval n, $h_p^{rs}(n)$ is the actual travel time on route p at time interval n from origin r to destination s, $e_p^{rs}(n)$ is flow over route p at time interval n from r to s during interval n, and $E_p^{rs}(n)$ is cumulative number of vehicles arriving at s from r during interval n.

Optimal Control Problem

The usual definition of optimal control implies the optimization of systems described by differential equations and at least one control function. The many-to-one problems have been solved by optimal control theory by Friesz et al. (1989), Wie et al. (1990), and Lam and Hauang (1995). They have developed formulations for both the system optimal and the user equilibrium problem, assuming known continuous time-dependent O-D demand flows. Wie et al. (1994) solved the discrete time dynamic system optimal traffic

assignment model of Merchant and Nemhauser (1978) as a discrete time optimal control problem, rather than as a mathematical programming formulation. A typical system optimal formulation using optimal control is as follows (Wie et al. (1994)), where the total transportation cost in the network is minimized over the planning horizon:

$$\text{Minimize} \quad J(x) = \sum_{i=1}^{I} \sum_{a \in A} C_a[x_a(i)]$$

subject to

$$x_a(i+1) = x_a(i) + u_a(i) - g_a[x_a(i)], \qquad \forall a \in A, i \in T$$

$$S_k(i) + \sum_{a \in B(k)} g_a[x_a(i)] - \sum_{a \in A(k)} u_a(i) = 0, \qquad \forall k \in M, i \in T$$

$$x_a(0) = x_a^0 \geq 0, \qquad \forall a \in A$$

$$u_a(i) \geq 0, \qquad \forall a \in A, i \in T$$

$$x_a(i) \geq 0, \qquad \forall a \in A, i \in T'$$

where $x_a(i)$ is the number of vehicles traversing link a at the beginning of time period i ($x_a(i)$ is considered as state variable), $u_a(i)$ is the number of vehicles entering link a in period i (this is considered as the control variable), $g_a[x_a(i)]$ is the number of vehicles leaving link a in period t (this is an exit function), and $C_a[x_a(i)]$ is the total transportation cost incurred by the traffic volume $x_a(i)$ on link a in period i. The input is the number of vehicles generated at node k in period i and destined for node n, $S_k(i)$. The first constraint described the dynamics of traffic flow on each link and is called the state difference equation. The second constraint represents the conservation of traffic and is called the state-control equality constraint. Wei et al. (1994) solved this optimal control problem using Augmented Lagrangian methods.

2.2.4 Solution Methodologies

Most of the studies reviewed in this chapter do not explain how an equilibrium can be found for large-scale networks and how we can solve the mathematical formulations efficiently. Some algorithms that can be used for relatively large networks are reviewed in this section. The solution methodologies are classified into two groups, algorithmic

and heuristic methods. Since most of these algorithms use the Frank-Wolfe method, this is reviewed here first.

Frank-Wolfe (F-W) Algorithm

The F-W algorithm was suggested by Frank and Wolfe (1956) as a way for solving quadratic programming problems with linear constraints. This method is also called convex combination method. It is widely used in determining equilibrium flows in static transportation network problems. In the static case, a minimization program with linear constraint is considered:

Minimize $Z(x)$

subject to

$$\sum_i a_{ij} x_i \geq b_j, \qquad \forall j.$$

The algorithm is a feasible descent direction method for which at iteration $(n+1)$, a point x^{n+1} is generated from x^n so that $Z(x^{n+1}) < Z(x^n)$. The algorithmic step can be written as:

$$x^{n+1} = x^n + a^n d^n$$

where d^n is a descent direction vector and a^n is non-negative scalar known as the step size. The above equation means that at each point x^n, a direction d^n is identified along which the function is decreasing. Then a^n determines the next point x^{n+1} will be along the direction d^n.

To find a descent direction, the algorithm finds an auxiliary feasible solution y^n such that the direction from x^n to y^n provides the maximum drop with respect to a first-order approximation. Therefore, the descent direction $d^n = (y^n - x^n)$ is found using the following mathematical program that determines y^n.

$$\text{Minimize} \quad \nabla Z(x^n)^T (y^n - x^n) = \sum_i \frac{\partial Z(x^n)}{\partial x_i}(y_i^n - x_i^n)$$

$$\text{subject to} \quad \sum a_{ij} y_i^n \geq b_j, \qquad \forall j$$

where $\nabla Z(x^n)$ is the gradient of Z at x^n.

The next step is to find the step size a^n using the following line search problem.

$$\underset{0 \le a^n \le 1}{\text{Minimize}} \quad Z[x^n + a^n(y^n - x^n)].$$

After finding the descent direction and the step size, the next point can be generated using the following formula:

$$x^{n+1} = x^n + a^n(y^n - x^n)$$

or

$$x^{n+1} = (1 - a^n)x^n + a^n y^n.$$

The last equation is a convex combination or weighted average of x^n and y^n. Figure-2.1 presents the flow-chart of the F-W algorithm. Starting with a feasible solution, the algorithm will converge after a finite number of iterations.

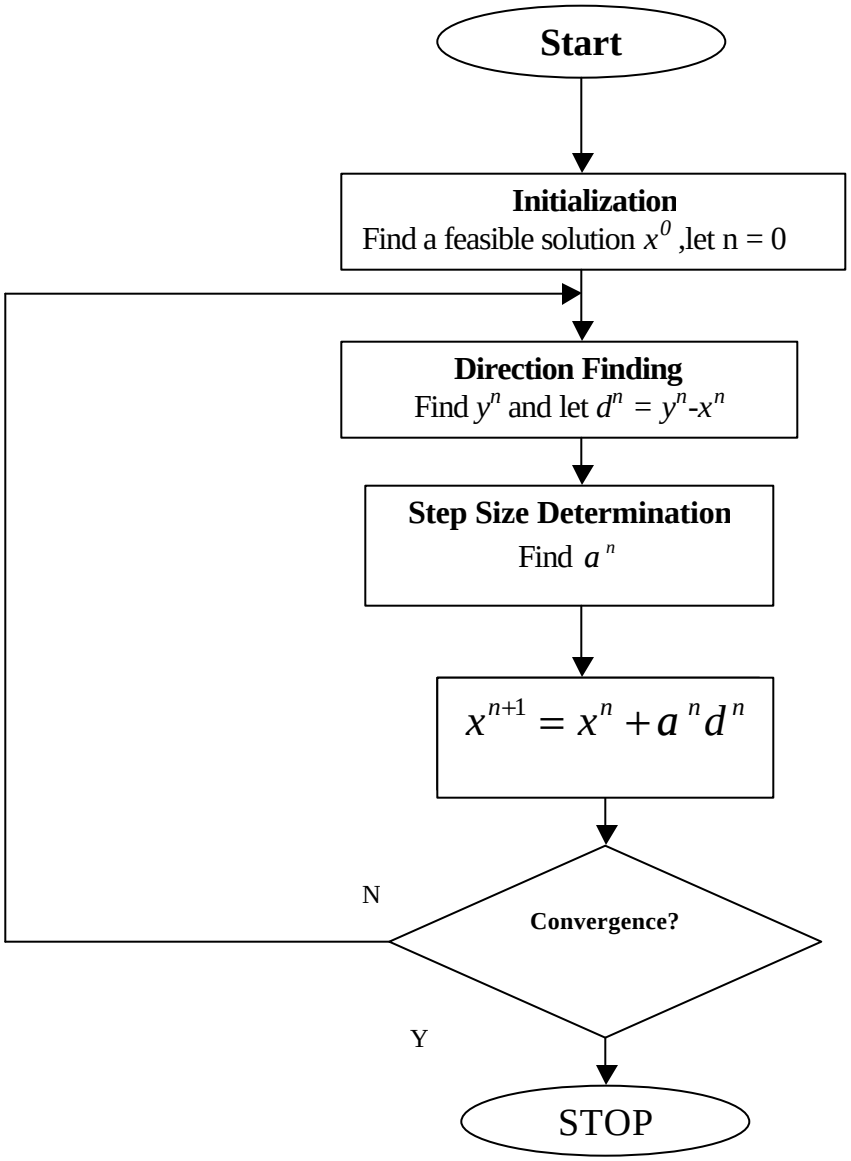


Figure-2.1: A Flow-chart for the Frank-Wolfe Algorithm.

2.2.4.1 Algorithmic Methods

Algorithmic methods are based on mathematical models and their analyses are typically theoretically guaranteed to find a solution.

Ran and Boyce (1996) proposed a method, in which the time period was divided into N small time increments. They then used two loops to find a dynamic user optimum. In the inner loop, they solved an NLP route choice model using the F-W algorithm. (They extended LeBlanc et al. (1975)'s method of using the F-W algorithm.) In the outer loop, they used a relaxation procedure to find a dynamic user optimum. The flow-chart of their overall solution algorithm is presented in Figure-2.2 and a more detailed flow-chart is presented in Figure-2.3. Here, each link is replaced with three separate artificial links for each time period to show the three variables associated with each physical link, by adding artificial nodes to define the new links. Each physical node is also expanded into *K* additional nodes without any extra link to provide a time-space expansion . Therefore, the expanded time-space network has [(3*A*+1)*K*] links and [(*N*+*A*)*K*+1][1] nodes and the original LP problem is transformed into a one-to-one minimal cost routing problem having flow propagation constraints.

In their example, they used the following link travel cost functions:

$$
\begin{aligned}
t_a(k) &= g_{1a}(k) + g_{2a}(k) \\
g_{1a}(k) &= b_{1a} + b_{2a}[u_a(k)]^2 + b_{3a}[x_a(k)]^2 \\
g_{2a}(k) &= b_{4a} + b_{5a}[v_a(k)]^2 + b_{6a}[x_a(k)]^2 .
\end{aligned}
$$

where, $x_a(k)$ represents vehicles on the link at the beginning of interval k, $u_a(k)$ and $v_a(k)$ represent inflow and exit flow during interval k, and b_{ia} (i :1...6) are the parameters for each link travel time function. The example is a 12-link and 9-node network. It required 20 incremental iteration for the initial solution, three inner F-W iterations per outer iteration, and 40 outer relaxation iterations to converge. Although their algorithm seems good, expanding the network would be questionable for large-scale problems.

[1] N is the total number of nodes and A is the total number of links in the network.

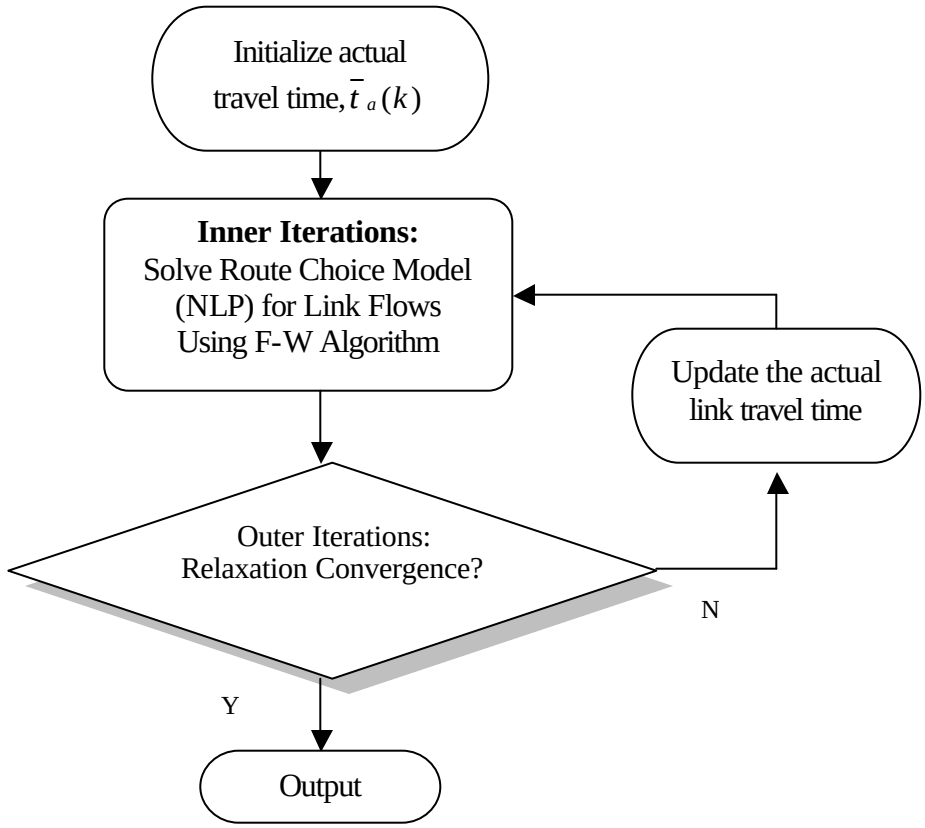


Figure-2.2: Flow-chart for the Ran and Boyce Solution Algorithm

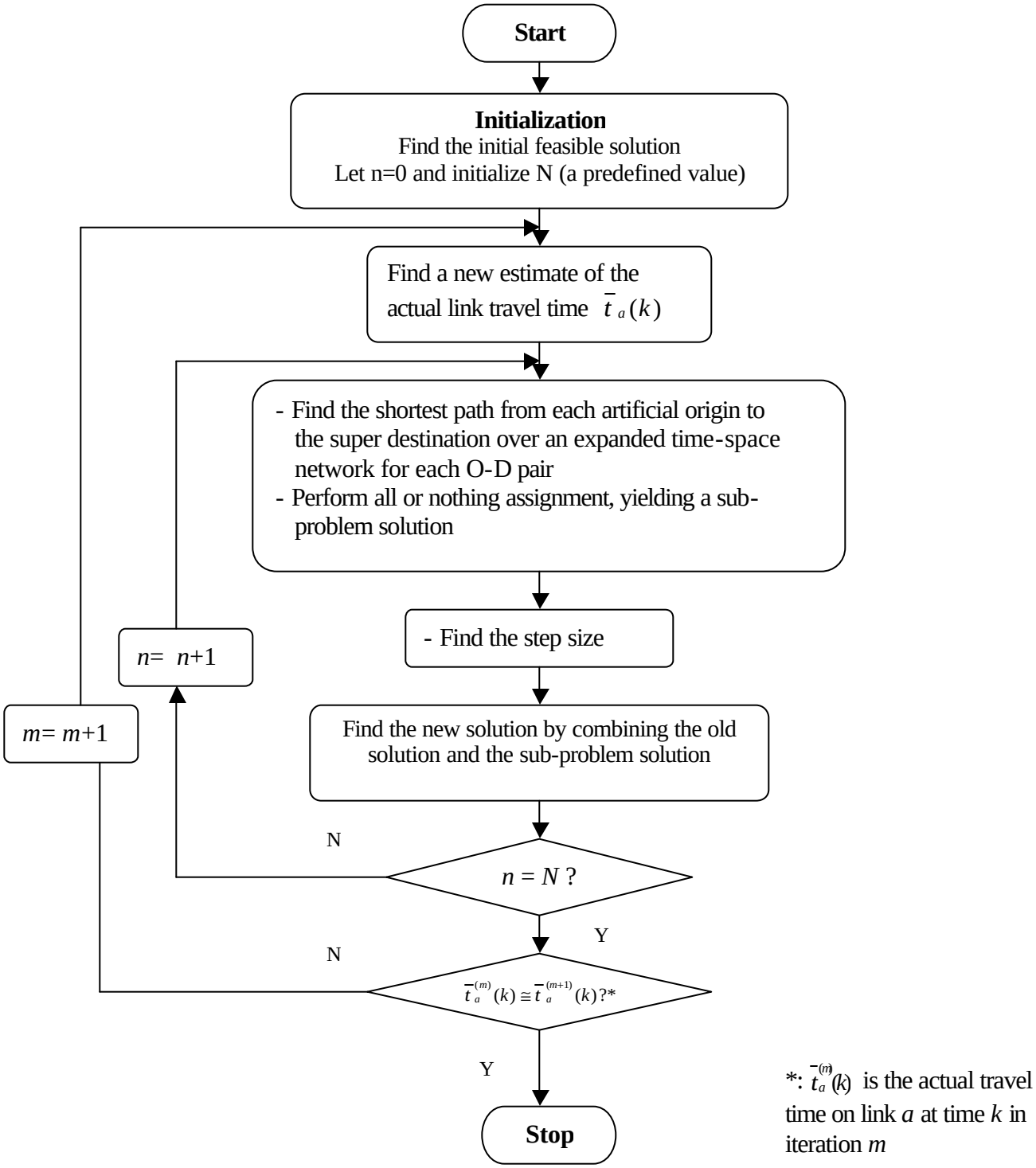


Figure-2.3: A detailed flow-chart of the Ran and Boyce Solution Algorithm.

Lie et al. (2000) formulated the DUE problem on networks with queues as a variational inequality (VI) problem over a polyhedral set by expanding the network so that the path enumeration is avoided. They assumed that the travel time on link a consists of two parts: The constant running time t_a and the delay time at the exit of the link due to the *point queue* (a queue having no physical length). The delay on link a is due to the limited departure capacity with maximum rate s_a. It is assumed that all vehicles have an identical speed on link a. The delay function and performance function over link a for vehicles entering at time t are:

$$d_a(t) = d_a(t-\Delta) + \frac{u_a(t) - v_a(t+t_a)}{s_a}\Delta, \qquad \forall a \in A$$

$$t_a(t) = t_a(t-\Delta) + \frac{u_a(t) - v_a(t+t_a)}{s_a}\Delta, \qquad \forall a \in A.$$

where, $u_a(t)$ is the link flow, $v_a(t)$ is the link exit function, and Δ is the length of discretized unit time interval. Figure-2.4 presents the general process of their algorithm.

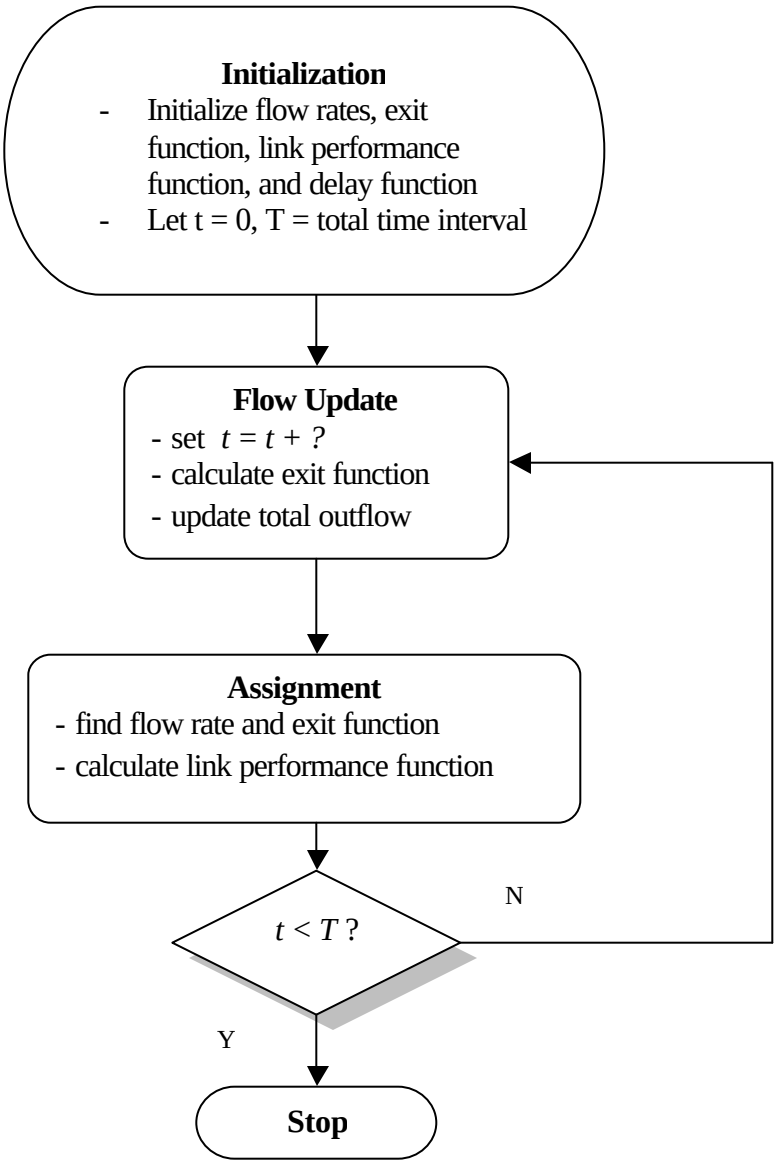


Figure-2.4: General Process of Li et al.'s Algorithm.

2.2.4.2 Heuristic Methods

Heuristic approaches involve a rule of thumb, simplification, approximation, or educated guesses that reduce or limit the search for solutions in domains that would otherwise be difficult to characterize or explore. Unlike exact algorithms, heuristics do not guarantee optimality, or even feasible solutions, and are often used with no theoretical guarantee.

Wie et al. (2002) developed a heuristic algorithm that does not need a path enumeration or storage of path-specific flows and cost information. Their iterative algorithm for solving the link-based non-linear complementarity formulation is a heuristic method in which the convergence is not established by certain regularity conditions. They used a heuristic to avoid having to evaluate the actual travel time too many times as needed by non-linear programming algorithms. Figure-2.5 presents the flow-chart of their solution algorithm. They assumed in their numerical example that link travel time function is:

$$C_a(x_{at}, u_{at}) = g_a + z_a x_{at} + l_a (x_{at})^2 + v_a u_{at} \Delta + x_a (u_{at} \Delta)^2$$

where g_a is the free-flow travel time on arc a, ? is the length of each time period that is assumed to be one-minute long, and z_a, l_a, v_a, and x_a are the parameters.

Wie et al. have shown that a DUE exists by using Brouwer's fixed point theorem and the solution is unique by imposing strict monotonicity conditions on the demand function and link travel cost. The shortcomings of this method are that the convergence of the algorithm is not guaranteed, the method needs to be accelerated for savings in computational time, and it does not find an equilibrium, but only an approximation of it.

Gawron (1998) proposed a queuing model in which each link has an assigned capacity and if the number of cars arriving at the link exceed the capacity, a queue builds up at this link. He used an iterative algorithm to determine a dynamic user equilibrium with respect to link costs. He assumed that the demand and the link capacities are constants from day to day, but the demand varies over the day. Also, his queuing model is simplistic and not realistic.

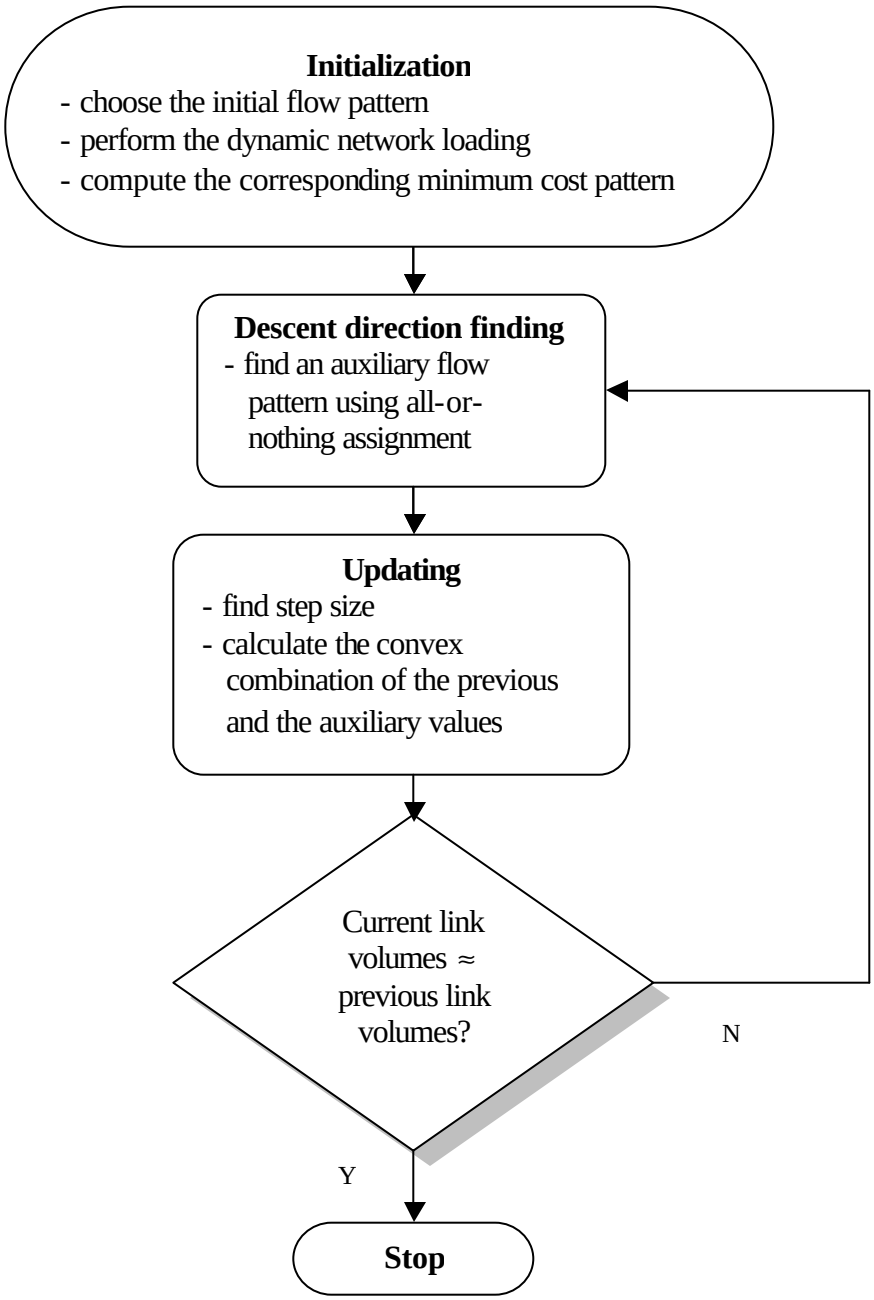


Figure-2.5: Flow-chart of Wie et al. (2002)'s Solution Algorithm.

All of the above methods are based on the Frank and Wolfe (1956) algorithm. This method cannot solve the problems without an explicit objective function. Realistic representations of the objective function leads to complex nonlinear functions for which an explicit analytical form may not be available.

Sheffi and Powell (1982) proposed the method of successive averages (MSA) for static user equilibrium problems and this has been used by other researchers in both UE and SO problems (Peeta and Mahmassani (1995), Contarella and Casceta (1995)). MSA is based on pre-determined step-sizes and converges to an optimum.

Peeta and Mahmassani (1995) presented a heuristic iterative procedure in which a special purpose simulation model, DYNASMART is used, to evaluate system performance and to represent the traffic interaction. Each simulation time interval is six seconds and it is assumed that traffic conditions do not change during this small period. They did not represent a complete mathematical formulation of this problem and hence, did not find an exact equilibrium, but they found an approximation of the equilibrium using simulation and stated that simulation implicitly satisfies the constraints required for the traffic assignment problem. The traffic simulator circumvents the need for link performance functions and link exit functions, ensures FIFO, captures link interactions, and prevents holding of traffic. Figure-2.6 presents their algorithm.

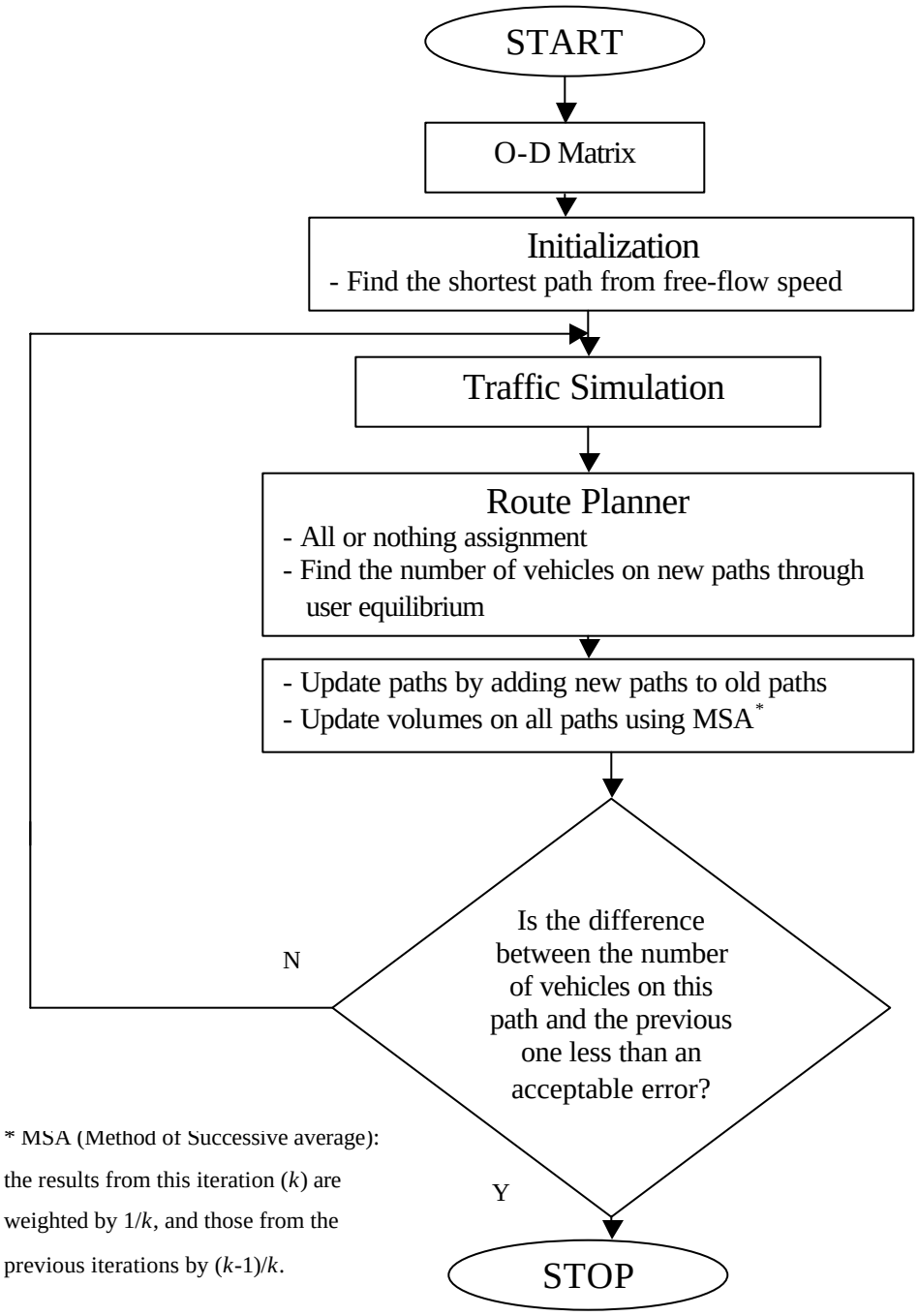


Figure-2.6: Flow-chart of Peeta and Mahmassani's UETDTA Algorithm.

All the above algorithms try to find a user equilibrium. One solution methodology for finding a system optimum that is considered in this study is Wie et al. (1994)'s algorithm. They formulated the dynamic traffic assignment as a discrete time optimal control problem. The algorithm consists of two iterative loops. The inner loop is similar to the F-W algorithm that is used to find the static equilibrium. The outer loop updates the dynamic marginal path costs and solves the two-point boundary value problem with fixed values of the marginal path costs (Lagrange multipliers). The stopping criterion for the outer loop is as follows:

$$\sum_{k \in M} \sum_{i=0}^{I-1} \frac{\left| m_k^{r+1}(i) - m_k^r(i) \right|}{m_k^r(i)} \leq w ,$$

where, $m_k^r(i)$ is the Lagrange multiplier for iteration r, and i is a time period. In their numerical example, the length of the time period (a) is considered to be one minute, the cost function is defined as $C_a[x_a(i)] = a\, t_a x_a(i)$, and the exit function is expressed as $g_a[x_a(i)] = g_a^{\max}(1 - e^{-[x_a(i)]/g_a})$, where $g_a^{\max}$ is the maximum number of vehicles that can exit from arc a in each time period, and g_a is a parameter that is determined by arc characteristics such as speed limit, length, and traffic signal setting.

Wie et al.'s numerical method has some advantages especially for large-scale networks. The algorithm requires neither a shortest path calculation nor a path enumeration. Thus, it reaches an equilibrium very fast. The memory space is also reduced by not using the enumeration of paths. Using time periods reduces the dimensionality of the model to be solved. The dynamic marginal costs are computed for all paths. This model could be extended to the multiple destination case and it could be changed to the user equilibrium approach.

Laborczi and Nowotny (2003) presented a fast heuristic method to solve stochastic dynamic system optimization problem. Their heuristic approach is based on Simulated Allocation, a stochastic method proposed by Pioro (1996). They defined two sets A and B that play an important role in their method. Set A contains the demands, for which no path was found and set B contains the demands, for which a path was found. Each step of the algorithm can be determined by two possible actions, Allocation and Deallocation. The Allocation step is chosen with probability $.5 < p < 1$ and the Deallocation step with

probability 1-p. After each Allocation step, the link travel times are increased and after each Deallocation step, the link travel times are decreased. They used a modified time-dependent Davidson's travel time function introduced by Akcelik (1991), and changed the network to be a time-extended graph, temporarily deleting all links that are loaded after the Allocation step. The algorithm stops if a threshold is overrun, e.g., the number of unsuccessful allocations exceeds a certain constant, TR_{unsucc}. Their algorithm is summarized in Figure-2.7.

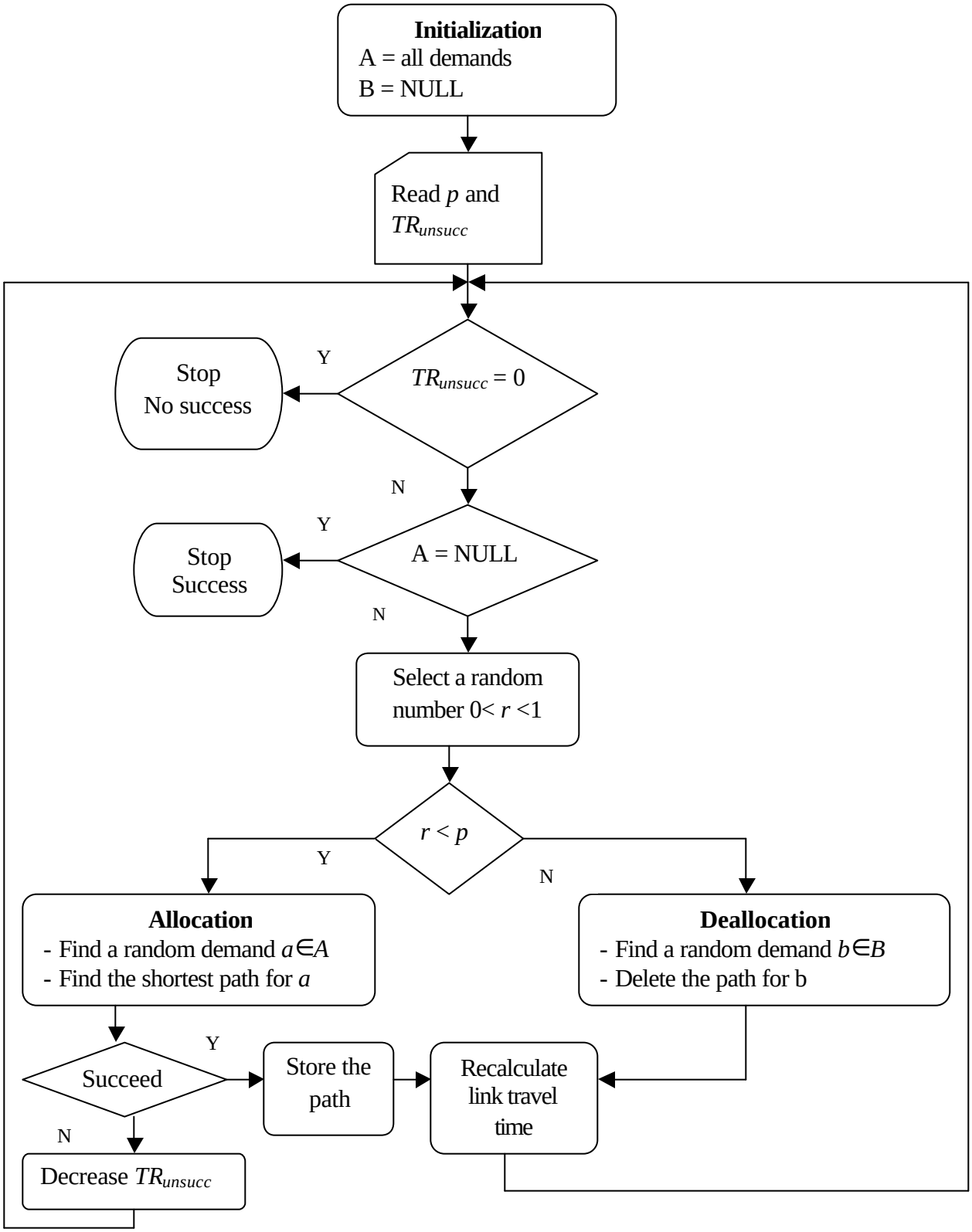


Figure -2.7: Flow-chart for the Laborczi-Nowotny Solution Methodology.

TRANSIMS, the Transportation Analysis and Simulation System, is a microscopic transportation modeling framework that consists of several modules: Population Synthesizer, Activity Generator, Route Planner, Microsimulator, Feedback, and Emissions Estimator. One of the most important objectives of the Feedback module is solving the traffic assignment problem. The simplest version of the traffic assignment is a loop between the Route Planner and the Microsimulator modules. In the first run, the Route Planner uses free-flow speeds on each link to estimate the travel time, which is not accurate because there exist other vehicles on the link and so, the speed is not simply equal to the free-flow speed. Therefore, some paths might not be the shortest path due to congestion. The Microsimulator produces the new travel times based on the accurate vehicle speeds. The new travel times computed by the Microsimulator are fed back to the Route Planner, and the new routes are selected as the shortest paths for selected travelers. In the Feedback module, a user can select a proportion of people to be re-routed in one of the following ways.

Random: A fraction of people are selected randomly.

Tail: People are selected based on the lowest or the highest cost. Cost is a network variable such as travel time.

All: All people in the network are re-routed.

This iteration is repeated until a stopping criterion is met. In the Portland study (2001), the selection of people to be re-routed was made by a uniform random sampling among all households. They stated that a targeted selection of travelers had yielded results that were insignificantly better, and in some cases worse, than using a uniform random selection.

Nagel and Barrett (1997) performed a case study to investigate the traffic assignment process in TRANSIMS. The case study was conducted in the Dallas-Fort Worth area. They routed all the travelers using free-flow speeds, and then used the Microsimulator to update the link travel times. Then they randomly selected 20% of the travelers and re-routed them. They performed ten iterations between the Microsimulator and the Route Planner to derive a reasonable traffic assignment. The fraction of selected travelers to be rerouted in each iteration were taken as 20%, 10%, 10%, 10%, 10%, 5%, 5%, 5%, 5%, and 5%, respectively. The stopping criterion used was the vanishing of deadlocks in the

Microsimulation, which was realized by visualization. To prevent oscillation, they used a 30% individually distorted view of the link travel times rather than using the correct values. Technically, a random number between 0.7 and 1.3 is drawn for each traveler for each link and the average link travel time obtained from the Microsimulator is multiplied to this random number.

Nagel et al. (1998) presented results of simulation studies from the Dallas-Fort Worth area on a relatively large network having 6124 links. They performed a feedback between the Route Planner and the Microsimulator to find a reasonable traffic assignment. They found that the best performing method was similar to the method of successive averaging. They started the re-routing fraction at 30%, which slowly decreased to 5% by the 20th iteration.

Raney and Nagel (2002) implemented the modified TRANSIMS for all of Switzerland. They found the shortest path for each traveler and incorporated memory of past plans. Travelers choose their new plans based on the performance of the routes in their memory. Thus, a fraction of travelers were chosen to be re-planned based on informed decisions rather than by random selection. They introduced a database to give travelers memory of their plans. The database includes three tables: Plan table, travel-time table, and flags table. The plan table includes the travelers' number, the plan number, plans, and the starting time of the plan. Once a new set of plans enters into the plan table, the travel-time table and the flag tables are joined and all are written into a file to be used to choose the next plan. Once the database knows which plans to choose (plans having flag=1), it writes that plan into the Microsimulator's input. The mechanism for choosing a plan is based on the assumption that each traveler has the following utility function of his/her stored plans.

$$P(tt_{a,i}) = \exp(-b\ tt_{a,i}),$$

where $P(tt_{a,i})$ is the probability of choosing a given plan i for traveler a, ß is an empirical constant, and $tt_{a,i}$ is the total travel time of traveler a for his/her plan i. This is similar to a logit model in discrete choice theory. Let P' be the normalized probability:

$$P'(tt_{a,i}) = \frac{P(tt_{a,i})}{\sum_{i=1}^{p} P(tt_{a,i})}.$$

The cumulative probability is then calculated by: $SC_{a,i} = \sum_{j=1}^{i} P'(tt_{a,j})$.

Traveler a next draws a random number $r \in [0,1)$, and chooses plan i such that:

$SC_{a,i} < r \leq SC_{a,i+1}$.

Thus, travelers are most likely to choose the plan having the highest performance, second-most likely to choose the plan having the second highest performance, etc. The value of ß determines how likely it is that a "non-best" plan will be chosen. Raney and Nagel chose the value of ß so that about 90% of the travelers would choose their best possible plan. Thus they allowed 10% of travelers to be re-routed. They set ß to be 1/360.

Besides TRANSIMS, the development of two different real-time simulation packages have been sponsored by Federal Highway Administration: DYNASMART and DynaMIT. DYNASMART is a dynamic network analysis and evaluation tool that was originally developed at the University of Texas at Austin, with participation from researchers at the University of Maryland, Northwestern University, Purdue University, and the University of California at Irvine. DYNASMART is an intelligent transportation network design, planning, evaluation, and traffic simulation tool. It models the evolution of traffic flows that result in a traffic network from the travel decisions of individual travelers. The model is also capable of representing travel decisions of travelers seeking to perform a chain of activities at different locations in a network over a given planning horizon. The dynamic traffic assignment component in DYNASMART is based on the solution methodology proposed by Peeta and Mahmassani (1995). DynaMIT is developed by the Massachusetts Institute of Technology (MIT) team and submitted to Federal Highway Administration in 2002. It is divided into two different sub-models DynaMIT-R and DynaMIT-P. DynaMIT-R is a real-time computer system for traffic estimation, prediction, and generation of traveler information and route guidance. DynaMIT-R supports the operation of Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems

(ATMS) at Traffic Management Centers (TMC). DynaMIT-P is the offline version of the real-time traffic estimation and prediction system DynaMIT. DynaMIT-P is a tool that can be used for the evaluation of Intelligent Transportation Systems (ITS) at the planning level and for the evaluation of short-term planning projects. It models the day-to-day evolution of traffic, traveler behavior, and network performance for special events and situations such as incidents, weather emergencies, sporting events, etc. In DynaMIT-P, a list of travelers and their paths is generated based on their current travel times using a demand simulator that implements demand dissaggregation. Then an aggregated supply simulator is run for the entire planning horizon to provide aggregate link travel times that are experienced by the travelers. The expected travel times (toward computing equilibrium solution) are updated by a convex combination of the travelers' expected travel times and their recent experienced travel times. The above procedure is repeated until convergence criterion based on the expected travel times of travelers matching their experienced travel times is satisfied. This is checked by comparing the equilibrium travel times in the current and the previous iterations using a weighted norm function. The flow-chart for the computation of equilibrium travel times in DynaMIT-P is presented in Figure-2.8.

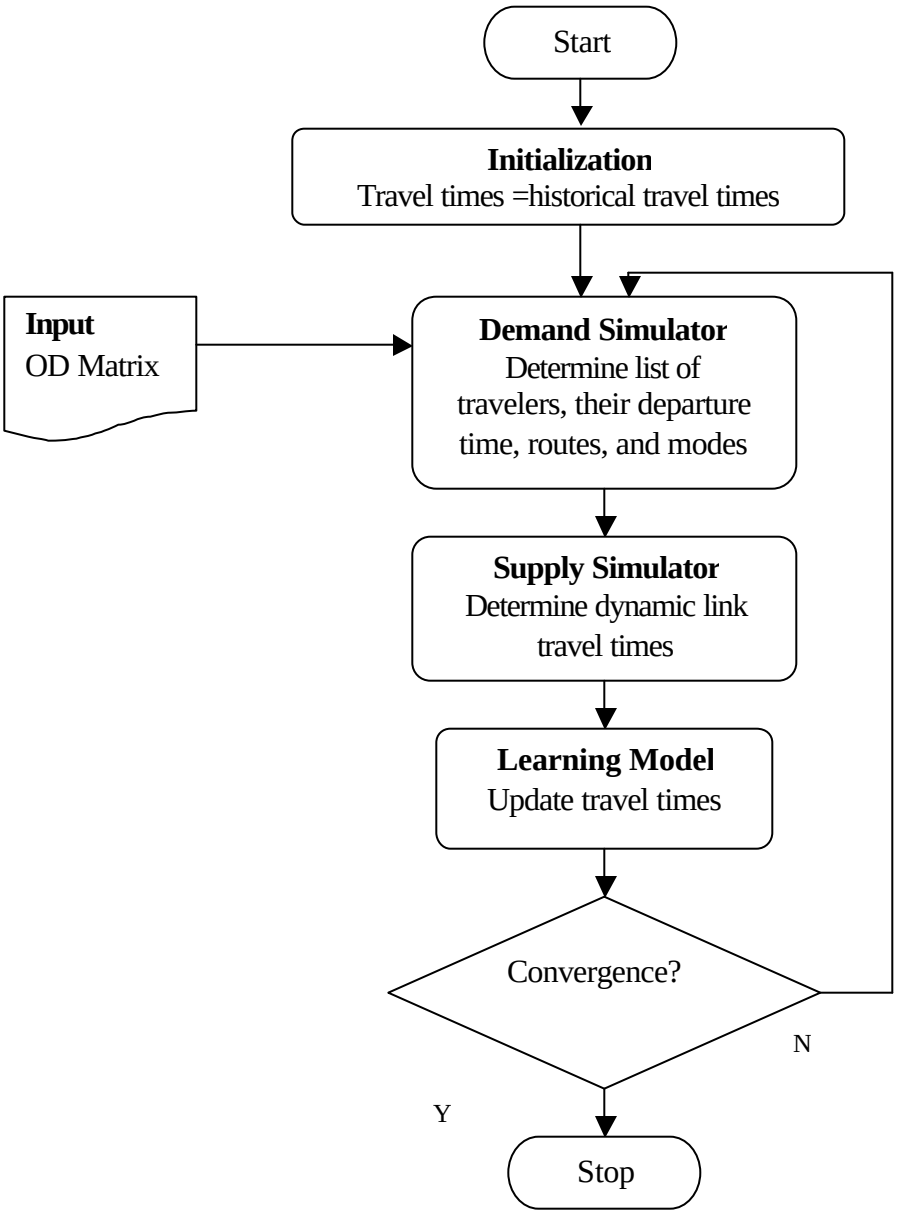


Figure-2.8: Flow-chart for the Equilibrium Travel Time Computation in DynaMIT-P.

2.2.5 Summary and Conclusion

As stated earlier, most of the models do not present a solution algorithm and among the presented methods, most of them are not suitable for large-scale networks.

Dynamic assignment models formulate the dynamic assignment problem using two different assignment philosophies: system optimal and user equilibrium. The formulations are based on mathematical programming, optimal control, or variational inequality. The solution methodologies can be classified into two different categories: heuristic or algorithmic, and analytical or simulation-based. The network can be viewed as a single or a multiple destination and also, as having a fixed or elastic demand. The assignment problem can be viewed as deterministic or stochastic, flow-based or vehicle-based, and also, link-based or path-based. All the above characteristics have been investigated collectively by Merchant and Nemhauser (1978), Friesz et al. (1989), Ran and Boyce (1996), Cantarella and Casceta (1995), Peeta and Mahmassani (1995), Wie et al. (1994), Li et al. (2000), Laborczi and Nowtony (2003), and Raney and Nagel (2002). Table-1 compares the capability of the reviewed models and Table-2 shows some characteristics of these models.

Table-2.1: Comparison of Dynamic Assignment Models.

Model	Formulation (M/O/VI) *	Solution Methodology (A/S) **	De /St***	Single (Si)/ Multiple (Mu) Destination	Algorithmic (Al)/Heuristic (H)
Merchant & Nemhauser (1978)	M	A	De	Si	Al
Friesz at al. (1989)	O	A	De	Si	Al
Ran and Boyce (1993)	VI	A	De	Mu	Al
Peeta & Mahmassani (1995)	M	S	St	Mu	H
Cantarella & Casceta (1995)	O	A	De/St	Mu	Al
Wie et al. (1994)	O	A	De	Si	H
Li et al. (2000)	VI	A	De	Mu	Al
Laborczi & Nowotny (2003)	-	A	St	Mu	H
Raney & Nagel (2002)	-	S	St	Mu	H

* M: Mathematical Program, O: Optimal Control, VI: Variational Inequality

**A: Analytical, S: Simulation- Based

*** De: Deterministic, St: Stochastic

Table-2.2: Comparison of Dynamic Assignment Model Characteristics.

Model	Fixed (F)/ Elastic (E) Demand	SO / UE	Flow (F) / Vehicle (V) Based Model	Link Exit Function	FIFO	Link (L)/ Path based (P)
Merchant & Nemhauser (1978)	F	SO	F	Yes	Yes	L
Friesz at al. (1989)	F	SO/UE	F	Yes	Yes	L
Ran and Boyce (1996)	F	UE	F	No	Yes	P
Peeta & Mahmassani (1995)	F	SO/UE	F	Yes	Yes	P
Cantarella & Casceta (1995)	F	UE	F	No	No	L
Wie et al. (1994)	F	SO	V	Yes	Yes	P
Li et al. (2000)	F	UE	V	Yes	Yes	P
Laborczi & Nowotny (2003)	F	SO	V	No	Yes	L
Raney & Nagel (2002)	F	UE	V	No	Yes	L

2.3 Drivers' Behavior

The lifestyle that we use and the factors that affect our economic and social activities influence directly or indirectly how we travel. Drivers' behavior is the result of a complex process involving human judgment, decision making and learning in a dynamic environment. There is uncertainty in this dynamic environment for two reasons. First, the consequence of a driver's decision depends on the other drivers' decisions. Second, the interactions that determine the outcomes are highly non-linear. The ideal way to study this process is to observe the actual driver decisions in the real-world.

There is a wide research on driver behavior in seven areas: Changes in road capacity, traffic restraint measures, new modal alternatives, information provision, tele-services, mobility management, and land-use policies. This section concentrates on the first three areas.

Three principal dimensions: time-dependent flow patterns, dynamics of commuter decisions, and real-time dynamics, are important in dynamic analysis of commuter decisions. The time-dependent formulation recognizes path selection and departure time choices. The second dimension consists of the day-to-day dynamics of user decisions in response to experienced congestion and exogenous information. The departure time and route choice, which the user may change from one day to another without requiring major changes in activity patterns or expenditures, are main short-run choices of interest. The third dimension is concerned with the flow patterns that result from the real-time decisions of commuters in response to traffic conditions and also supplied information and reliability.

Many researchers state that changes in route and departure times are the most important driver responses to changes in network conditions. Cairns et al. (1998) reviewed evidence from ninety case studies of road capacity reduction and concluded that the two responses, changing routes and changing departure times are the most universal.

The first section explains time-use data and its application in studying drivers' behavior. The second section reports on the variations in travel behavior. The third section classifies the existing theoretical and empirical models. The fourth, fifth, sixth, and seventh sections review the results of existing papers on switching routes, departure

times, schedule delays, and travel modes respectively. Finally, the seventh section provides a summary and conclusion.

2.3.1 Time-use and travel

Since the travel demand is difficult to understand by examining the characteristics of the transportation system, a collection of time-use data has been used by many transport researchers. Some researchers have argued that activity-based analysis is a more fundamental analysis than person trips. The major elements in the activity setting framework are actors, activities, space or location, and time.

Actors: Employment status and gender determine the activity patterns. The presence of young children affects the time allocation of the family.

Activity: Activity definition and identification are difficult due to multidimensionality of activities. It has been suggested that a tradeoff between workability and meaningful detail can be achieved with eight categories of activity, excluding travel.

Time: Time has three relevant dimensions which can affect or be affected by planning and design considerations: position (the point at which actions occur), duration (the period during which actions occur), and sequence.

Space: This is not only a geographical location, but also a temporal location.

Many researchers have used time-use data to explore travel behavior. Time-use data is valuable since activity diaries give comparable or even better results than travel diaries. The earliest application of time-use study is done by Javeau (1972). He examined the commuting behavior between home and workplace. He also explored the time-tabling of the work trips of the Belgian population. Javeau analyzed two separate groups, one consisting of the five largest Belgian agglomerations, and the other of small or medium sized agglomerations. He found that the evening rush hour was 30 minutes later for the smaller places.

2.3.2 Variations in traffic conditions and travel behavior

It is generally acknowledged that traffic conditions change from day to day. Part of this short-run variation is because of temporary changes in network conditions such as road closure. Also a degree of day-to-day variation is part of the drivers' normal lives.

The classic work on this subject is done by Smeed and Jeffcote (1971). They reported on the variability of a trip repeated 253 times between Bray and central London. They found that between 10 to 20% of trips to work would be traveled with an overall door-to-door time more than 20% faster than the average, and a similar proportion would travel more than 20% slower than the average. Bonsall et al. (1983) traced 100 drivers between 8:15a.m. and 8:30a.m. on a given weekday and found that a week later, 30% of the drivers would drive past the same point between 8:15a.m. and 8:30a.m., 15% between 7:15a.m. and 8:15a.m., 7% before 7:15a.m. or after 9:45a.m..; 14% of drivers would drive to the same destination by a different route and 8% would travel by another mode; 5% would travel to a different destination, 5% would stay at home, and 1% would have sold their car. Berg and Bartsch (1995) studied eight Swiss examples of capacity reductions and concluded that the majority of drivers did not change their routes and accepted the additional time delays. A minority changed the route. Atkins (1985) reported surveys for 111 households to describe the effect of a bridge toll in Southampton. He concluded that after a week, 60% did not change their behavior, 14% changed their routes, 8% changed their travel mode, 7% changed their trip time, 6% changed their journey, and 5% changed their destinations. But after four months, 49% remained in the same behavior, 25% changed their journey, 13% changed their destinations. The remaining 13% changed their routes, trip times, or modes. Hs study shows that the long-run effect of changing the network characteristic is totally different from the short-run.

Jan et al. (1999) used GPS data from 216 drivers over a one-week period and reported that the path chosen most often differs from the shortest path across the network. Also, commuters follow the same path for the same trip.

Stephenson and Tepley (1984) investigated the data after the closure of the Kinnaird Bridge in Edmonton. Based on their investigation, drivers who were using the closed bridge were not the only ones to change their routes. The drivers who were not using the bridge changed their routes to avoid the congestion resulting from the closure. They also compared two days from before the closure period, and found that 60% of drivers traveled at the same time over the two days. Daugherty et al. (1999) reported that traffic tends to divert from the priority route if drivers perceive that their journey may be delayed along certain sections of the route. This would not be a problem if traffic diverts

to routes suitable and capable of absorbing the extra demand. Lock and Gelling (1976) reported on the effect of the collapse of the Tasman Bridge in Tasmania, where the morning peak extended from 7-9a.m. to 6:30-9a.m. in 1976.

The MUSIC (Management of Traffic USIng Flow Control) project studied the effect of introducing new signal control policies in three European cities and concluded that a long period after the implementation of the new traffic signals is necessary to allow rerouting and attainment of a new traffic equilibrium.

2.3.3 The Modeling Challenge

This section presents and classifies the driver's behavior modeling in response to traffic information, congestion, and network changes. The models basically concentrate on routes, departure times, and mode choices in the short-run.

2.3.3.1 Work trip modeling

Most researches on trip mode and departure time choices have focused on the work trip. In work trips, people need to arrive at their work places at certain times. Thus, work trips are not as flexible as non-work trips in regard to departure times. Suppose that commuters have a particular work start time. Four major factors may cause a commuter to plan to arrive earlier or later than this start time: (i) Congestion avoidance- the travel time can be reduced by avoiding peak congestion period. (ii) Schedule delay- early or late arrivals at work may be due to the schedule of shared-ride vehicles. (iii) Service reliability- workers may plan to arrive earlier than their work start time to avoid a late arrival when travel time varies from day to day. (iv) Peak/off-peak tolls and parking availability- varying charges for parking, transit fares, or roadway facilities by time-of-day causes changes in planned arrival time. Work trip modeling can be divided into three different categories that are explained below.

2.3.3.1.1. Trip timing for the work commutes under equilibrium conditions

These models are based on a standard microeconomic perspective that views the choice of departure times as a tradeoff between trip timings and schedule delays. The schedule delay is the difference between the desired and the actual arrival times.

Small (1982) modeled theoretically and empirically the scheduling of a discrete activity at the individual level. He focused on the distribution of departure times for trips to work. The commuters were assumed to trade off the schedule delay against travel time. He assumed commuters receive some disutility from schedule delay. The disutility raises linearly in either the late or early direction. Also, there is a discrete penalty for being late. Scheduling disutility is traded off against the possible advantages of shifting a commuter's schedule to take advantage of lower congestion. Therefore, the work trip scheduling model is formulated and empirically estimated using a standard random utility maximization (RUM) framework.

Hendrikson and Plank (1984) treated mode and departure time choices as a simultaneous interactive decision and used a logit model to analyze traveler decision. They chose two representative individuals traveling to a central business discrete (CBD) workplace from two suburbs to estimate the elasticities. Since the logit demand model is based on individual choices, elasticity values represent the choice sensitivity of individuals.

Mannering, Abu-Eisheh, and Arnadottir (1990) introduced a discrete/continuous econometric framework in which commuters' departure times and route choices are modeled. The choice of routes was modeled as being discrete and the choice of departure times was modeled as continuous. They assumed that departure time is continuous and not discrete because commuters have some control over vehicle speed and hence, the trip travel time. Spatial and temporal equilibria were defined in the classic economic sense.

Most research has focused on departure time choices under certainty. Noland and Small (1995) used schedule delay, defined as the difference between the actual arrival time and some ideal time (e.g. official work start time), to model departure time switching. They explained a model in which uncertainty in travel time affects the expected cost of the morning commute.

2.3.3.1.2 Prediction of within-day equilibrium departure patterns

To predict the departure time patterns of peak period congestion, it is necessary to recognize the interrelation between the system attributes in a congested system and the departure decisions of users.

Hendrikson and Kocur (1981) developed a simple approach for modeling schedule delay in a deterministic setting based on user equilibrium and illustrated the importance of schedule delay for departure time decisions. Their work is a direct extension of Wardrop's static conditions, such that no users can improve their utility by switching departure time and route. They experimented with three simple commuting examples with and without possible late arrivals at work, and imposing a toll charge. They also extended the three cases to analyze the situation in which a general distribution of work start time exists.

2.3.3.1.3 Day-to-day dynamics

Surveys of daily choices of actual commuters show day-to-day variations in trip timing decisions for the work commutes. Jou and Mahmassani (1996) conducted surveys of commuter behavior in Dallas and Austin. Their focus was on departure time and route decisions for morning and evening work-related trips. They developed Poisson regression models of the frequency of daily switching of departure time and route, separately and jointly. They prepared the survey in two stages. In the first stage, a short questionnaire was mailed to a large number of households in Dallas and Austin. In the second stage, two types of diaries were designed to record the day-to-day behavior of a smaller sample of commuters over two weeks.

The day-to-day adjustments of trip-timing decisions of commuters in congested systems were first investigated through a series of controlled laboratory experiments. Behavioral process models assume that commuter daily adjustment behavior follows simple heuristic strategies and mental rules. These models depart from the formal utility maximization paradigm and view the behavior of commuters as a boundedly rational search for an acceptable outcome. Ben-Akiva et al. (1986) developed a dynamic model of peak period traffic congestion with a limited number of bottlenecks. In response to changes in the traffic conditions, commuters can switch to a different mode, change their route, or changes their departure times. They modeled the delays at the bottlenecks with a deterministic queuing model at the time of arrival at the bottleneck. The day-to-day adjustment of the distribution of traffic was derived from a dynamic Markovian model. They performed simulation experiments using this model.

Mahmassani (1990) presented an overview of an experimental approach to investigate the day-to-day dynamics of commuter behavior in congested traffic system. He presented models to predict the daily switching of departure times and/or routes by commuters in response to congestion or exogenous information. His focus was on trip timing and routing decisions for the home-to-work morning commute. He conducted interactive laboratory-like experiments in which actual commuters interact in a simulated traffic system. This allows simultaneous measurements of individual user decisions and the associated traffic performance characteristics. Three experiments were performed; the first two involved a single-route corridor used for travel from a residential origin to a single work location, allowing users one choice dimension, namely, the departure time. In the third experiment, two routes were available in the commuting corridor, allowing departure time and route choices. A convergence to the steady-state was attained in the first and second experiments but not in the third one through the 7-week period of the experiment. He gave two reasons for this non-convergence: The availability of an additional choice dimension and the greater level of congestion compared to the first two experiments. He extended a previously developed framework for departure time choice dynamics (Mahmassani and Chang, 1985, 1986; Chang and Mahmassani, 1988, 1989) to include route switching. He assumed a boundedly-rational decision framework in which the user has two indifference bands, one for departure time and the other for route. Each bound has two components, one for early and the other for late arrivals. Actually, he extended his previous model to include the route switching dimension by defining separate dynamically varying random thresholds that govern departure times and route switching.

Mahmassani and Liu (1999) investigated departure time and route switching decisions made by trip makers in response to real-time information, based on data collected using a laboratory interactive dynamic simulator. They presented both a model framework and an empirical analysis based on the data collected from an experiment. The experiment involved actual commuters who interact with each other within a simulated traffic corridor. The trip makers could determine their departure times and routes at the origin and their path en-route along their trip. Forty five randomly selected subjects were used in the experiment for five decision days from full-time faculty and staff members at

the University of Texas at Austin. They used a multinomial probit (MNT) model, which provides a flexible framework to model and calibrate the trip maker's joint departure time and route switching behavior. They postulated that a commuter does not switch his/her next day's departure time as long as the corresponding schedule delay on the current day, which is the difference between the preferred arrival time and the actual arrival time, remains within the commuter's indifference band for departure time switching. They also assumed that a commuter does not switch route or path as long as the trip time saving, which is the trip time difference between the current path and the best path, remains within the commuter's route indifference band.

2.3.3.2 Non-work trip modeling

Most researches on trip mode and departure time choice has focused on the work trip while non-work travel accounts for about three-quarters of the total trips in urban areas. A recent paper on non-work trips is Bhat (1998). Since the home-based shopping (HBS) purpose constitutes a major fraction of total non-work trips, he focused on the HBS. Travel mode and departure time choice of urban trips are determinants of urban travel demand. He examined the joint nature of mode and departure time choices for urban shopping trips. He used five discrete time representations, a.m. peak (6a.m. - 9a.m.), a.m. mid-day (9a.m. -12 noon), p.m. mid-day (12 noon - 3p.m.), p.m. peak (3p.m. – 6 p.m.), and other (6p.m. – 6a.m.), for two reasons. First, commuters have departure time flexibility for shopping trips and are likely to choose among broad time periods rather than a precise time. Second, he mentioned that using this model, he could evaluate the effect of policy measures better. The author formulated a joint model of mode and departure time choice using a nesting structure in which the mode choice is modeled at the higher level of the hierarchy and the departure time choice at the lower level. He found that the alternative nesting structure (departure time choice modeled at the higher level and mode choice at the lower level) is inconsistent with global utility maximization. The nesting structure implies that individuals are more likely to shift departure times than change travel modes in response to congestion. Especially in shopping trips, commuters are more likely to shift departure times than shift modes. He used the multinomial logit (MNL) formulation for the higher-level mode choice decision and the standard ordered

generalized extreme values (OGEV) formulation for the lower level departure time choice decision. The data were drawn from the San Francisco Bay area household travel survey conducted by the Metropolitan Transportation Commission (MTC) in 1990. He estimated three different models of mode-departure time choices: The MNL model, the nested logit (NL) model with departure time alternatives within each mode, and the MNL-OGEV model proposed by him. His latter model has only one additional parameter over the nested logit model and at the same time, allows a more flexible correlation structure among alternatives than does the nested logit model. This is important when the number of alternatives is very large. He concluded that the nested logit model outperforms the MNL model in terms of data fit and the MNL-OGEV model outperforms the NL model using the same criterion. The MNL and NL models underestimate the reduction in p.m. peak period congestion due to implementation of congestion pricing in the p.m. peak. These models also underestimate the displacement of driver alone mode share to the adjacent p.m. off-peak and p.m. evening time periods. He concluded that the MNL and NL models lead to inappropriate policy evaluations of transportation control measures and biased level-of-service estimates.

2.3.4 Evidence on route choice and switching

There are three definitions of route switching:

1) Mode switching (all days);
2) same definition as 1) but only considering those days with no stops;
3) day-to-day switching from the previous day's route.

Jou and Mahmassani (1996) found that route switching is more frequent in Dallas for evening than for morning commutes under all definitions except definition 2). There are more opportunities to switch routes to arrive on time to work in Dallas.

More joint (route and departure time decisions), switching occurs in the evening. Dallas has more morning joint switchings than Austin, but the same amount of evening switching. Jou and Mahmassani conduct surveys of commuter behavior in Dallas and Austin and investigated if the decision to switch routes relates to switching departure times. They found the evolution of percent of users switching both route and departure

times, percent switching only one of them, and percent switching each of them separately. When only one dimension is changed, it is most likely the departure time, and when a user switches the route, it is frequently accompanied by a change of departure time. Mahmassani and Stephan (1998) established that these two decisions are not independent. Mannering, Abu-Eisheh, and Arnadottir (1990) used a survey of 151 morning commuters conducted in State College, Pennsylvania in 1986. The survey concentrated on a single origin-destination pair. They assessed the impact of reducing the capacity of the arterial by 50% when total travel demand remains constant and observed a diversion of traffic to two alternate routes. It was found that the longest-duration queue on the arterial begins at 7:37 a.m. and lasts 15 minutes.

Mahmassani and Liu (1999) stated that commuters tend to keep their routine departure time but change their routes when the real-time information system becomes less reliable. When the system provides an under-estimate trip time information, commuters become more prone to switch routes than when the system provides over-estimated trip times. If commuters have experienced an increase in travel time resulting from a small adjustment in the departure time, they will have a greater schedule delay. When the differences between the predicted arrival time and the preferred arrival time increases, commuters tend to switch their route both pre-trip and en-route. Also, trip makers tend to switch routes when they perceive a late arrival by following the current path than when they perceive an early arrival. Their results confirm the need to incorporate the correlation between departure time and pre-trip route decisions.

2.3.5 Evidence on departure time choice and switching

There are four ways of capturing departure time switching behavior:

1) Median switching, from a commuter's departure time;
2) Median switching, with work start time or end time controlled;
3) Day-to-day switching, from a commuter's previous day's departure time; and
4) Day-today switching, with work start or end time controlled.

In the first and second ways a switch is considered to occur whenever the absolute value of difference between the current commuter's departure time and the median is

greater than or equal to some minimum threshold. In the third and fourth ways, a switch is considered to occur whenever the absolute value of difference between the current commuter's departure time and the previous day's departure time is greater than or equal to some minimum threshold. The day-to-day definitions capture more switches than the median-based definitions. The second and fourth ways do not consider the departure time switching caused by different work start or end times.

Jou and Mahmassani (1996) conducted surveys of commuter behavior in Dallas and Austin. They found that switching is more frequent in the evening than in the morning under all four ways and thresholds. The percentage of workers never switching departure times in the mornings in Dallas is 30% under the 10-minutes median definition, and 8% under the 5-minutes day-to-day definition. These percentages are 40% and 7% for Austin, respectively. They also found that average travel times to and from work for Austin commuters on days are less than for Dallas commuters. About 52% of the Dallas and 42.6% of the Austin commuters reported more than 5 minutes tolerance to lateness at the workplace.

For both cities, departure time switching frequency in the morning is influenced by workplace attributes (lateness tolerance at work, flexible work hours, and so on), traffic conditions, and trip chaining factors.

They finally concluded that commuters tend to change departure times, routes, or both, more frequently in the morning than in the evening because they have a constrained arrival time at work and a flexible arrival time at home. The switching pattern is different in the morning for the two cities but is the same in the evening. They state that route and departure time decisions are interdependent and the percentage of route switching is lower than the departure time switching in both cities. There are more opportunities to switch in order to arrive at the workplace on time for commuters in Dallas. They concluded from the comparative tests that socioeconomics, commuter preference, and workplace condition variables have similar effects on morning departure time switching behavior in both cities, but the effect of trip chaining behavior is different between the two cities. For evening departure time switching behavior, socioeconomics, workplace conditions, and routine stop factor variables have similar effects on morning departure time switching behavior in both cities.

Hendrikson and Plank (1984) examined the flexibility of departure times for the trip to work using a data from Pittsburgh, Pennsylvania. 65% of CBD commuters reported earlier departure times during a transit strike in 1976 in Pittsburgh. Also, based on a survey of 1800 commuters through a construction zone, work trip departure times were 19 minute earlier during the reconstruction of a major roadway in Pittsburgh. They used the Pittsburgh CBD data, which offers a useful laboratory to study time-dependent travel variations using 80,000 workers commuting to a triangle area. The pattern of travel time peaking was shown to be quite regular in each area, with the maximum travel time occurring at about the same time on each day of observation. With experience, commuters can predict the amount of time required to travel to work with a particular departure time on any given day. Thus travel time peaking occurs in a regular pattern on each day. In an area with unscheduled service, the average wait time was shown to be relatively short and constant during the peak period. A random or Poisson arrival process of transit vehicles was suggested since the variance of wait time is equal to the average wait time. Scheduled transit services were shown to have irregular variations in average wait times and the variance of wait times over the peak period. Using a sample of 1800 workers in the CBD, they also reported that most downtown workers arrive early. The mean departure time was 7:00a.m., the mean arrival time was 7:40a.m., and the average work start time was 8:00a.m.. They reported that the probability of choosing to depart in a ten-minute interval, for both individuals was more than 20%. The departure time interval was observed to have a much greater sensitivity to change in transit fare, wait time, travel time, and auto cost. Thus, individuals could alter their departure time and still choose the transit mode. The removal of congestion did not result in significant mode shift but the departure time experienced a significant shift. This shift was towards later departures and periods having lower wait times. The departure time decision occurred to be more elastic than the choice of mode.

Noland and Small (1995) stated that as uncertainty increases, commuters shift their departure times to earlier hours to compensate for the probability of arriving late. Sometimes the commuters overcompensate, and the probability of a late arrival decreases as uncertainty increases.

Mahmassani and Liu (1999) concluded that in the departure time switching decision

model, older trip makers have greater schedule delay than the younger ones. Also female commuters have a wider mean indifference band for departure time and route decisions than male commuters. Mahmassani (2000) also reported that commuters have a greater propensity to change departure times than routes.

2.3.6 Evidence on schedule delay switching

In Hendrickson and Kocur's (1981) experiment, 1500 vehicles were required to arrive at work by 8:30a.m. over a facility with a 1500 vehicles per hour capacity. In the first case, late arrivals were not permitted, and they observed that the first worker arrived at 7:30a.m., experienced no queue, and had a one hour schedule delay. The last worker arrived at the bottleneck at 8:15a.m., waited 15 minutes in the queue, but arrived exactly at the work start time. The maximum delay was 15 minutes and the maximum queue length was 375 cars, which occurred at 8:15a.m.. In the second case, late arrivals were permitted, the first worker arrived at 7:42a.m. and the last one arrives at 8:42a.m.. The maximum queue length was 300 cars, which occurred at 8:18a.m. with a 12 minutes wait. In the third case, late arrivals were permitted but a toll was charged to prevent the formation of a queue, and the first and last arrivals turned out to be the same as for the second case. They increased the capacity in the absence of tolls and concluded that increasing capacity does not eliminate queuing but reduces the average wait time and the duration of the queue.

Small (1982) concluded that on average, urban commuters will shift their schedules by one to two minutes toward the early side or by 1/3 to one minute toward the late side, to save one minute of travel time. Ben-Akiva, De Palm, and Kanaroglou (1986) considered two parallel routes with commuters jointly selecting routes and departure times. The mode and route choices were dependent on travel times and travel costs. The departure time choice was based on the tradeoff between travel time and schedule delay. Route 1 had a capacity of 8000 vehicles and Route 2 has a capacity of 3000 vehicles with a shorter distance. Simulation results showed that the beginning of congestion occurred at 7:04a.m., the maximum delay occurred at 8:00a.m., the end of the congestion period was at 8:36p.m., and the period of on time arrivals began at 7:12a.m. and the period of late arrivals began at 8:04a.m.. During a particular period the travel time on Route 2 was

greater than the travel time on Route 1. The travel time on Route 2 before and after this period was faster. The shifts from Route 1 to Route 2 caused this problem. They found that in the case of inelastic demands an increase in arrival time flexibility always reduced the delays. Increasing arrival time flexibility caused the spreading of the departure times over a longer period. Thus, more drivers chose to travel on Route 2 and the volume on Route 1 decreased. In the case of elastic demand, the total volume increased with arrival time flexibility. In the elastic total demand case, the total volume decreased with increasing toll.

2.3.7 Evidence on travel mode choice

Mahmassani (2002) has reported that in the case of drivers switching to a public transport service, travelers prefer a rail-based to a bus-based system. Hendrickson and Plank (1984) started that the shared ride modes of travel have the greatest amount of travel time peaking, with 77% of all workers sharing rides (carpools, vanpools, etc), arriving between 7:15 to 8:15a.m.. Workers driving alone have the lowest degree of arrival time peaking and transit services are at intermediate level. In the survey data, 55.5% used public transportation, 15.9% drove alone, 24.2% shared rides and others used bicycles or walked. They chose two representative individuals traveling to a CBD workplace from two suburbs to estimate the elasticities. Individual A had a high probability of using transit (80%), but individual B had a 53% probability of using transit. The elasticities were calculated for changes in transit fares, transit wait times, congestion delays on transit, walk to transit, and auto travel costs. For each of these changes, choice sensitivity was inelastic, meaning that a 1% change in the level of service characteristic resulted in a less than 1% change in transit choice probabilities. For individual A, a 1% increase in the peak period fare caused a 2.6% reduction in the probability of transit use during the peak period. A $1 increase in toll on all automobile trips shifted people towards transit ridership but caused little shift in the departure times. Horwitz (1993) estimated a binary response model of travel mode choice, automobile and transit, using fixed and random-coefficient probit specifications, and semiparametric specifications. He concluded that distributional assumptions can be important in applying binary response modeling.

2.3.8 Summary and Conclusion

Transportation related decisions of travelers often depend on what others are doing. For example decisions about mode choice, route choice, activity scheduling, etc., can depend on congestion, caused by the aggregated behavior of others.

In response to the information about network conditions such as congestion, commuters tend to switch their routes, departure times, travel modes, and destinations or origins with respect to modes.

While there is little empirical evidence of the effect of introducing a road-pricing scheme on travel behavior, there exists a consistent picture. The preferred driver response is first to re-route and then to switch departure times. If destination switching is a realistic option, it is preferred to modal switching. If the existing destination is of a high quality and access type, then mode switching becomes attractive. If alternative modes are poor, then significant trip suppression might occur.

The response to changes in network conditions is different in the short, medium, and long runs. For example in capacity reduction, some studies show that in the first few days, there are longer queues and worse congestion than usual. After a while, there is a more settled period as traffic adjusts to the new conditions. After several years, the reduction in traffic due to the capacity reduction is offset by growing traffic levels for other reasons.

Thus, for fixing unrealistic transportation movements or congestion, a set of hierarchical changes based on the above order can be examined. Since TRANSIMS is very flexible, any model and framework can be used to begin the Feedback module. The Feedback module adjusts travel activities in response to problems or new opportunities. Such adjustments change travel routes, travel modes, activity locations, activity schedules, and the number of activities. In the existing Feedback module, it is necessary to define who, what, and how to change. A framework based on the above driver behavior studies can be defined to choose and implement the changes. Table-3 compares the characteristics of the different discussed models and Table-4 exhibits the different studies regarding driver behavior.

Table-2.3: Comparison of Driver Model Characteristics.

Study	Model Type *	Statistical model	Characteristic of concern	Conclusion
Small (1982)	Em/ Th	RUM	Scheduling/ departure time	Urban commuters shift their schedule delay to save on travel times.
Hendrikson & Plank (1984)	Em/ Th /Ex	Logit	Departure time / mode	Departure time decisions are more elastic than mode choice. Share ride, transit, and cars have the greatest, medium, and lowest degree of arrival times.
Mahmassani & Jou (1996)	Em/ Th	Poisson Regression	Departure time / Route	Changes in departure times, routes, or both are more frequent in the morning than in the evening.
Mahmassani & Lui (1999)	Em/ Th / Ex	MNT	Departure time / Route	Drivers tend to keep their departure times same but change their routes when information becomes less reliable.
Bhat (1998)	Em/ Th	MNL/ NL/ MNL-OGEV	Departure time / Mode	Individuals are more likely to shift departure times than change travel modes in response to congestion.

* Em: Empirical, Th: Theoretical, Ex: Experimental modeling

Table-2.4: Comparison of Different Studies on Drivers' Behavior.

Study	Data	Change in network conditions	Conclusion
Bonsall et al. (1983)	Tracing 100 drivers at 8:15-8:30a.m. on a given weekday	No Change (Investigating a week later)	Only 30% did not change, 36% changed their times, 14% changed their routes, 8% changed their modes, 5% changed their destinations.
Stephenson & Tepley (1984)	Commuting Data of Edmonton	Closure of Kinnaird Bridge	60% of drivers traveled at the same time after closure. The drivers who were using the bridge were not the only ones who changed their routes.
Hendrikson & Plank (1984)	1800 commuters through a construction zone in Pittsburgh, Pennsylvania	Reconstruction of a major roadway	Work trip departure times were 19 minutes earlier during the reconstruction period. Departure times are more sensitive to changes in transit fares, wait times, travel times, and auto costs than to changes in trip modes.
Atkins (1985)	111 household survey in Southampton	Bridge toll	60% did not change their behavior, 14% changed their routes, 8% changed their travel modes, 7% changed their trip times, 6% changed their journey, and 5% changed their destinations.

Chapter 3

The Feedback Process in TRANSIMS

3.1 Introduction

The urban travel demand models, developed in the mid 1950s, provided accurate and precise answers to the planning and policy issues being addressed at that time, which mainly revolved around expansion of the highway system to meet the rapidly growing travel demand. However, the urban transportation planning and analysis have undergone changes over the years, while the structure of the travel demand models has remained largely unchanged except for the introduction of disaggregate choice models beginning in the mid-1970s. Legislative and analytical requirements that exceed the capabilities of these models and methodologies have driven new technical approaches such as TRANSIMS.

TRANSIMS consists of a series of modules that produce synthesize households, activities for individuals within each household, the choice of routes for movements among these activities, and the microsimulation of these movements to create traffic dynamics on the network, and consequently produced emission estimation as shown in Figure-3.1. TRANSIMS framework allows each module to be executed in any desired order by a set of scripts specified by the user in the Feedback Controller. TRANSIMS starts with creating the identity of individual synthetic travelers and maintaining them throughout the entire simulation. All synthetic travelers are generated by the Population Synthesizer module using census data, land-use data, and network data.

After the Population Synthesizer module estimates the number of synthetic households, and the demographics, characteristics of each individual of these households, and the locations of these households on the network, the Activity Generator create activity list for each synthetic traveler. These activities include work, shopping, school, etc. These activity estimations are based on the activity survey demographic characteristics of individuals, and from the survey data. In addition, activity times and activity locations are determined for each individual.

Then the Route Planner module computes the combined route and mode trip plans to accomplish the desired activities of each individual, such as work, shopping, etc. the Traffic Microsimulation module uses the intermodal paths developed in the Route Planner module to perform a regional microsimulation of vehicle interactions. The

microsimulation continuously computes the operating status, including locations, speeds, and acceleration or deceleration of all vehicles throughout the simulation period. The output can provide a detailed, second by second history of every traveler in the system over a 24-hour period. Every motor vehicle in the study area is monitored in this manner to identify traffic congestion and emission concentrations, which is done by the Emissions Estimator module. The Emissions Estimator module provides the data for the air quality analysis. Using the vehicle information generated in the microsimulation module, the emission module forecasts the nature, amount, and location of motor vehicle emissions. The emission information is then used in urban air shed models to predict urban air quality. Data from the emission module will be compatible with the EPA's Models-3 air shed model.

Finally, the Feedback Controller module manages the feedback of information among the Activity Generator, the Route Planner, and the Traffic Microsimulator modules of TRANSIMS as shown in Figure 3.1. This feedback controller module makes decisions such as what percentage of the regional trips should be fed back between modules, which trips should be fed back, how far back the trips should go for replanning, and when to stop iterating to reach stability in the results. As in the traditional assignment methodologies, the Route Planner may place more vehicles on links than the capacity of the link may allow, this may cause congestion to spill back onto other links. Results of the microsimulation in these cases can be fed back to reroute selected travelers to stabilize this situation.

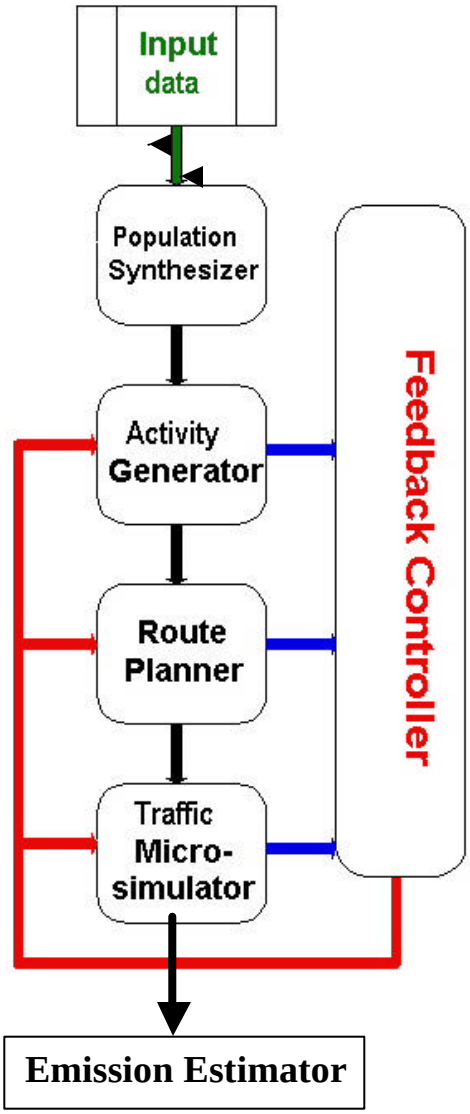


Figure -3.1: The Framework of TRANSIMS.

3.2 Overview of the Feedback Module

A key distinguishing feature of TRANSIMS is the process known as iterative feedback. Feedback provides a natural way to tailor models to specific, possibly overlapping, subpopulations. The Feedback module uses an iteration script provided by the user to control the overall framework of TRANSIMS. A typical study using TRANSIMS involves repeated iterations between its modules. There is no single, "standard" iteration script because different study designs involve different iterative schemes.

One important example of feedback is in solving the traffic assignment problem. The simplest version of this uses a loop between the Route Planner and the Traffic Microsimulator modules to determine the actual travel times from the Microsimulator and consequently to feed this back to the Route Planner to compute the shortest routes for the travelers in the network. In the first iteration of the Route Planner, routes are chosen under the hypothesis that travel times are represented by the free-flow speeds on the network. Of course, there are many more information-flows in TRANSIMS than just the travel time table. Every TRANSIMS module can be used to update only a selected subpopulation using information provided by the feedback process. In effect, this is like providing a separate model for every conceivable subdivision of the population without the need for fitting each model separately. For example, work location is chosen using a single simple model for the entire population. If people who commute by bus across a river are assigned work locations poorly, selecting that subpopulation and running the work location assignment model with slightly different input information can change the poorly selected locations for that subpopulation with no change in the model itself.

The structure of the Feedback Module is shown in Figure-3.2. It has three modeling tools, Collator, Stratifier, and Selector.

The Collator gathers data from some input files of TRANSIMS, i.e., the Network file and the Transit Route file. It also extracts data from the output files of the Population Synthesizer, the Activity Generation, the Route Planner, and the Traffic MicroSimulator. All of data collected are then put into the Collator Iteration Database.

The Stratifier divides trips into Binnings and Stratifications based on the criteria the user has defined. The numeric identifier of each Binning and Stratification is added to the iteration database and the database is developed into the Stratifier Iteration Database.

The Selector model's responsibility is to pick up a subset of travelers from the Stratifier Iteration Database. This is done in two steps. In step1, the targeted cells are picked up from the stratification, where a subset of travelers or trips is selected within the cell. In Step 2, the output of the Selector module for each selected subset will consist of one Activity Feedback file and one Route Planner Feedback file. The Activity Feedback file is then sent to the Activity Regenerator, and the Route Planner feedback file is sent back to the Route Planner. The user is free to use either or both of them according to his/her intention.

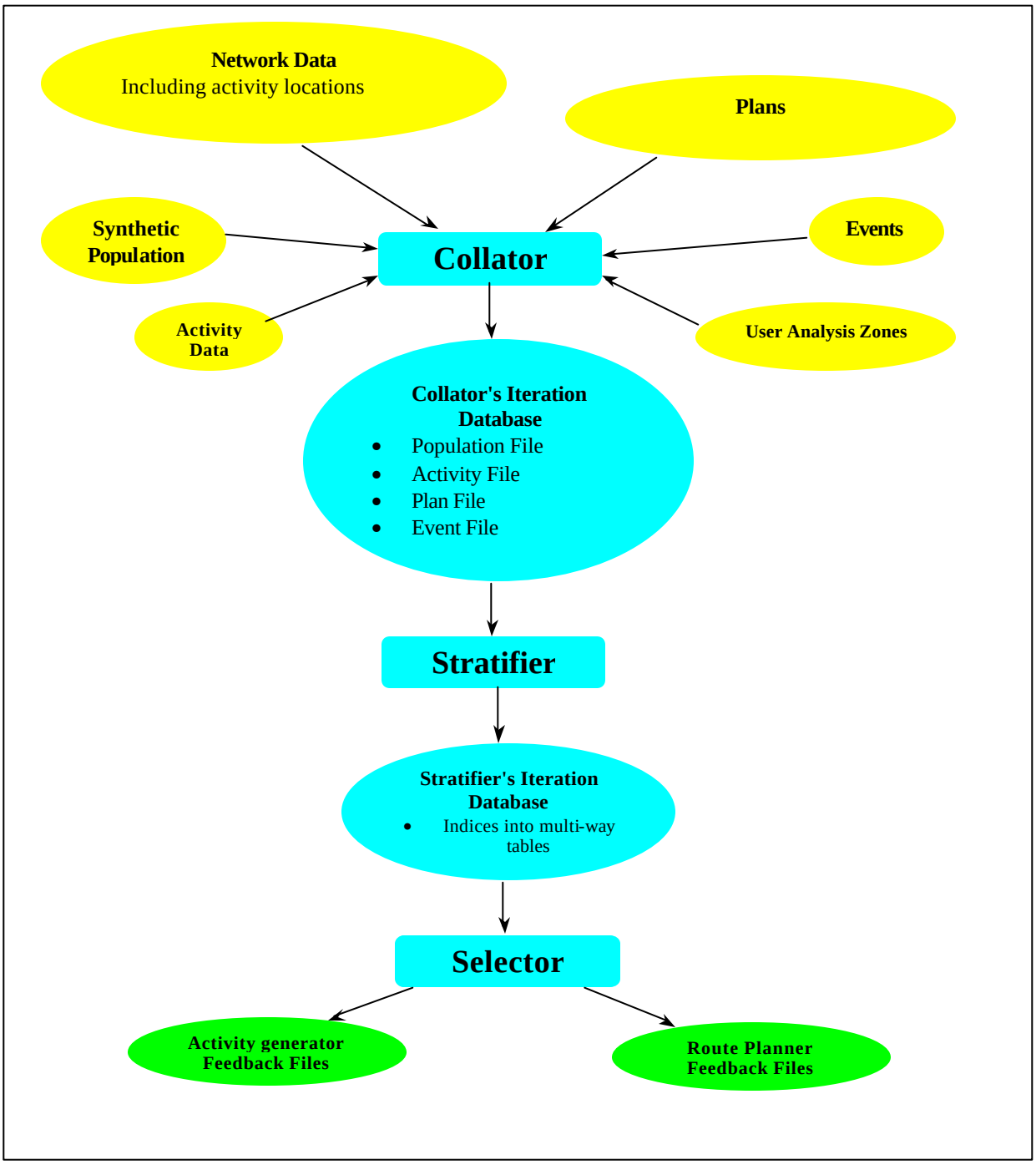


Figure-3.2: Data Flow Diagram of the TRANSIMS Feedback Module.

3.3 Terminology

- **Iteration:** Execution of one TRANSIMS program (e.g., Activity Generator, Route Planner, Traffic Microsimulator).

- **Iteration Database:** It is the archive of information about travelers across iterations. It is used by selector to make its selection decisions. It enables user to filter results and run repeated iterations.

- **Iteration Script:** It is edited to control the overall TRANSIMS Framework with Iteration Database by the user.

- **Collator:** This is one of the tools providing Selector/Iteration Database functionality. Its main function is to gather data from disparate sources (e.g., activity files, plan files, event files) into a single table keyed by traveler ID and trip number. It accumulates data over an entire trip and provides some commonly used processing algorithms. It can be run after each module and will fill in all of the fields in Iteration Database that depend on that module with the most recent data available.

- **Stratifier:** This is one of the tools providing Selector/Iteration Database functionality. It uses a combination of built-in algorithms on the information contained in ITDB to stratify or classify trips.

- **Selector:** This is one of the tools providing Selector/Iteration Database functionality. Its goal is to either reroute the traveler or to make use of one of the feedback pathways defined by the Activity Generator. It uses the Iteration Database to select a set of travelers. It has a set of algorithms and each algorithm may require its own set of parameters and is associated with a name, goal and a cost function.

- **Tour:** This concept is only used in Selector Module for the user's convenience. Each trip starting from the home location is defined as the start of a new tour.

- **Subtour:** This concept is only used in Selector Module for the user's convenience. Each trip from an anchor activity that returns to the same anchor location before returning to the home location is defined as the start of a new subtour.

- **Binning:** Each discretized variable is known as a "binning". Discretization is implemented in Stratifier and binnings are added to the Iteration Database. Each binning is given a numeric identifier.

- **Bin:** During discretization, each item in a binning is called as a “bin”. Each bin is given a numeric identifier. Discretization is implemented by Stratifier.

- **Stratification:** It is a *k*-way table created by stratifier from the binnings. Each stratification is given a numeric identifier. Each cell in the *k*-way table is given an index and represents a group of travelers with similar preferences, demographics and/or experiences.

- **Selector Choices:** Files that list the travelers who will be reassigned activities, replanned, resimulated, etc.

- **Selector Statistics:** Statistics that provide a basic summary of the choices the Selector makes.

- **Feedback Controller:** It has a control over the tools-*collator*, *stratifier* and *selector*. It reroutes selected travelers, updates travel times, regenerates activities and microsimulates all travelers.

3.4 Key Concepts

The Iterative Process

The most important function of the Feedback Module is the iterative process. During each iteration, the user invokes the Collator, Stratifier, and Selector to do the following:

- Read information about the travelers from the Iteration Database.
- Examine each traveler and decide whether to
  - regenerate his or her activities using the Activity Generator,
  - select a new route between his or her existing activities using the Route Planner, or
  - retain his or her existing activities and the planned route between them.
- Write the selections made for each traveler into data files that can be read by the Activity Generator and the Route Planner when they are executed.

Figure-3.3 illustrates a step in the iterative process conducted by the Selector module.

After the Selector completes the selection process for all travelers, the Activity Generator, Route Planner, or Traffic Microsimulator runs to calculate the updated activity set, plan set, or microsimulation output files, respectively (according to the decisions made by the Feedback).

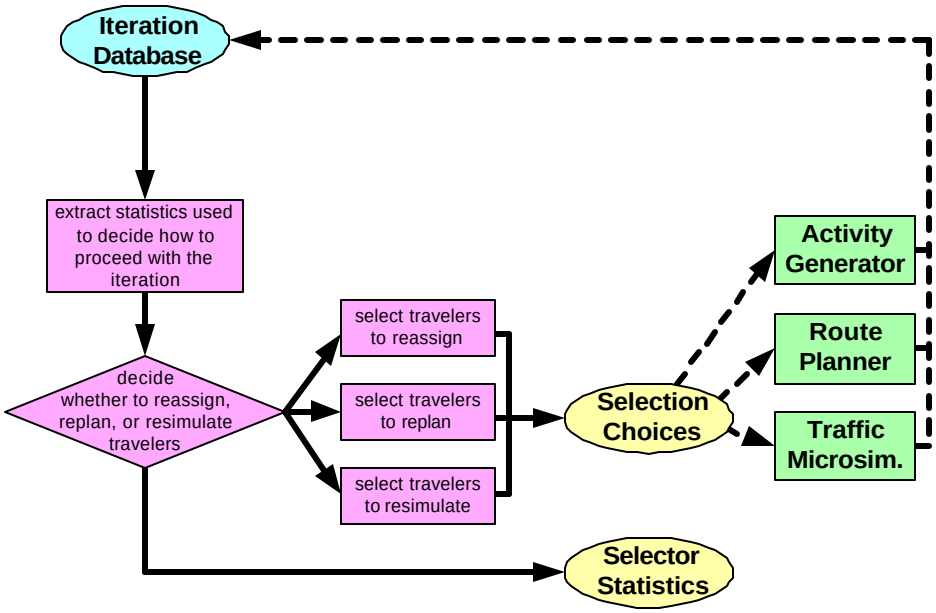


Figure-3.3: Typical Selector/Iteration Database Logic.

The iteration script will reinvoke a Selector again at the start of the next iteration in the study. Figure-3.4**Figure-3.4:** shows an example of one possible progression, as determined by the Selector.

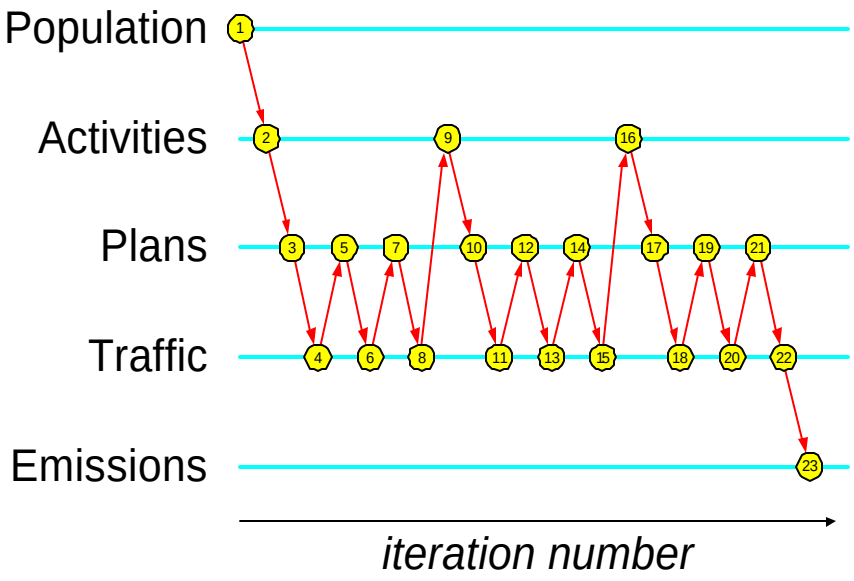


Figure-3.4: An Example of Iteration Progressions.

The Iteration Database is the archive of information about travelers across iterations. The Selector uses this information to make its selection decisions. The data contained in the database are chosen by the user from:

- The fields of the population, activity, and plan files—for example, income, mode preference, or the expected duration of a trip.
- Information extracted from detailed Traffic Microsimulator event output—for example, the actual duration of a trip.
- Information deduced from combinations of the previous two—for example the duration of a trip relative to its expected duration.

The left side of Figure-3.5 shows this data flow into the Selector/Iteration Database.

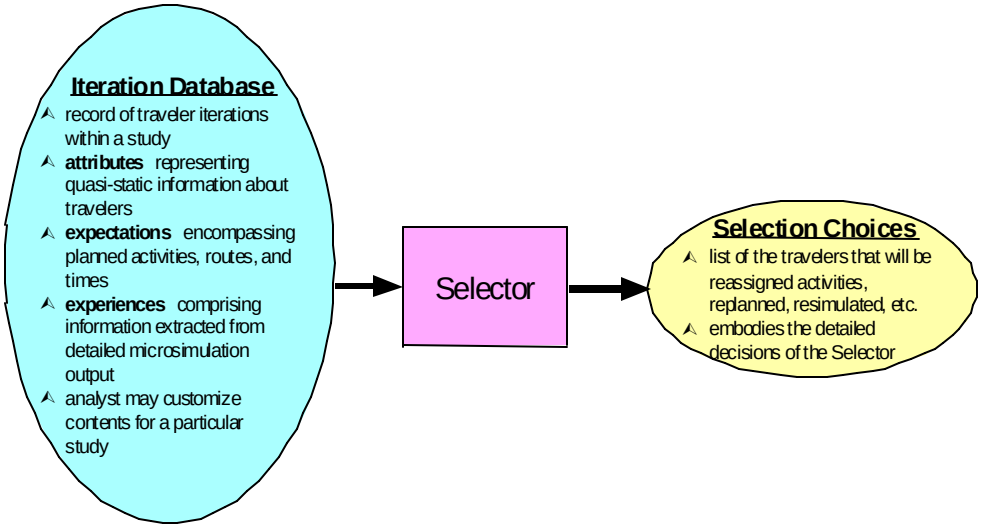


Figure-3.5: Typical Selector/Iteration Database Data Flow.

Flexibility of TRANSIMS' Framework

The Feedback Module is very flexible and mainly controlled by the user. It only provides information pathways and tools for manipulating the information. Users have to design the order in which the modules are called and how the feedback process is executed.

The Framework's flexibility allows for countless variations in the iterative process. For example, in some studies, the Selector may run after the Activity Generator or the Route Planner completes its execution. Thus, the Selector can decide which of the generated activities or plans will be accepted for travelers. Those not accepted are discarded and new activities or plans are produced.

The iteration script has the potential to make additional choices, such as the following:

- which version of the Activity Generator, Route Planner, or Traffic Microsimulator will run during the present iteration;
- if transit schedules will be adjusted or vehicles added or removed from the transit fleet;
- if network characteristics (such as traffic signal timing, congestion pricing, or roadway information signs) will be altered;
- which travelers receive data from traffic information systems; or
- whether to complete the study (i.e., end the iteration) because the iterations have converged sufficiently (or diverged).

Several implementations have been written that were used in the demonstration project for a typical transportation planning study. For example, Figure-3.3 shows a typical iteration scheme that is set up by the Selector/Iteration Database script. In this scheme, activities, plans, and microsimulations are iterated until traffic behavior on the network stabilizes. It is not difficult for analysts to write additional iteration scripts for their own specialized studies.

3.5 Major Data Inputs

Feedback Module can extract data from any input files according to the user's intention. Usually, Network file, Transit Route file and Transit Schedule file might be used in the feedback processes.

3.6 Major Data Outputs

There are no direct outputs from Feedback Module to users. All of the outputs from Feedback Module need to be sent back to other Modules for feedback purposes.

3.7 Module Interfaces

The information about travelers available to TRANSIMS consists of the traveler-specific data contained in population, activity, plan, vehicle, and simulation output files. For feedback purpose, Feedback Module can extract data from the output files of all other TRANSIMS modules, except Emission Estimator. After some kinds of analysis, selected data will be sent back to Activity Generator and Route Planner module.

An outline for the module interfaces involving the Feedback Module is displayed in Figure-3.6.

3.7.1 Inputs received from the Population Synthesizer Module

All of the output data of Population Synthesizer Module can be used as inputs to the Selector Module.

The major output of the Population Synthesizer module is a synthetic population of households containing a set of information associated with each household and each person within the household. The Population Synthesizer also generates and assigns private vehicles to households.

Household demographic data include tract ID, block group ID, household ID, number of persons living in the household, number of autos in the household, the home location ID, the PUMS household ID, the presence of persons under the age of 18 in the household, number of workers in the family, and the household income.

Person demographic information includes household ID, person ID, age of the person, relationship of that person to the householder, sex of the person, whether that person worked in the year 1989.

3.7.2 Inputs received from the Activity Generator Module

All of the output data of Activity Generator Module can be used as inputs to the Selector Module.

The primary output of the Activity Generator Module is the list of activities for each member of a synthetic household. Each activity list is comprised of the following:

- An activity type (e.g., home or school, etc.) and its priority;
- Starting, ending, and duration time preferences;
- A preferred travel mode;
- A vehicle preference (if appropriate);
- A list of possible locations for an activity, and
- A list of other participating HH members (if the activity is shared).

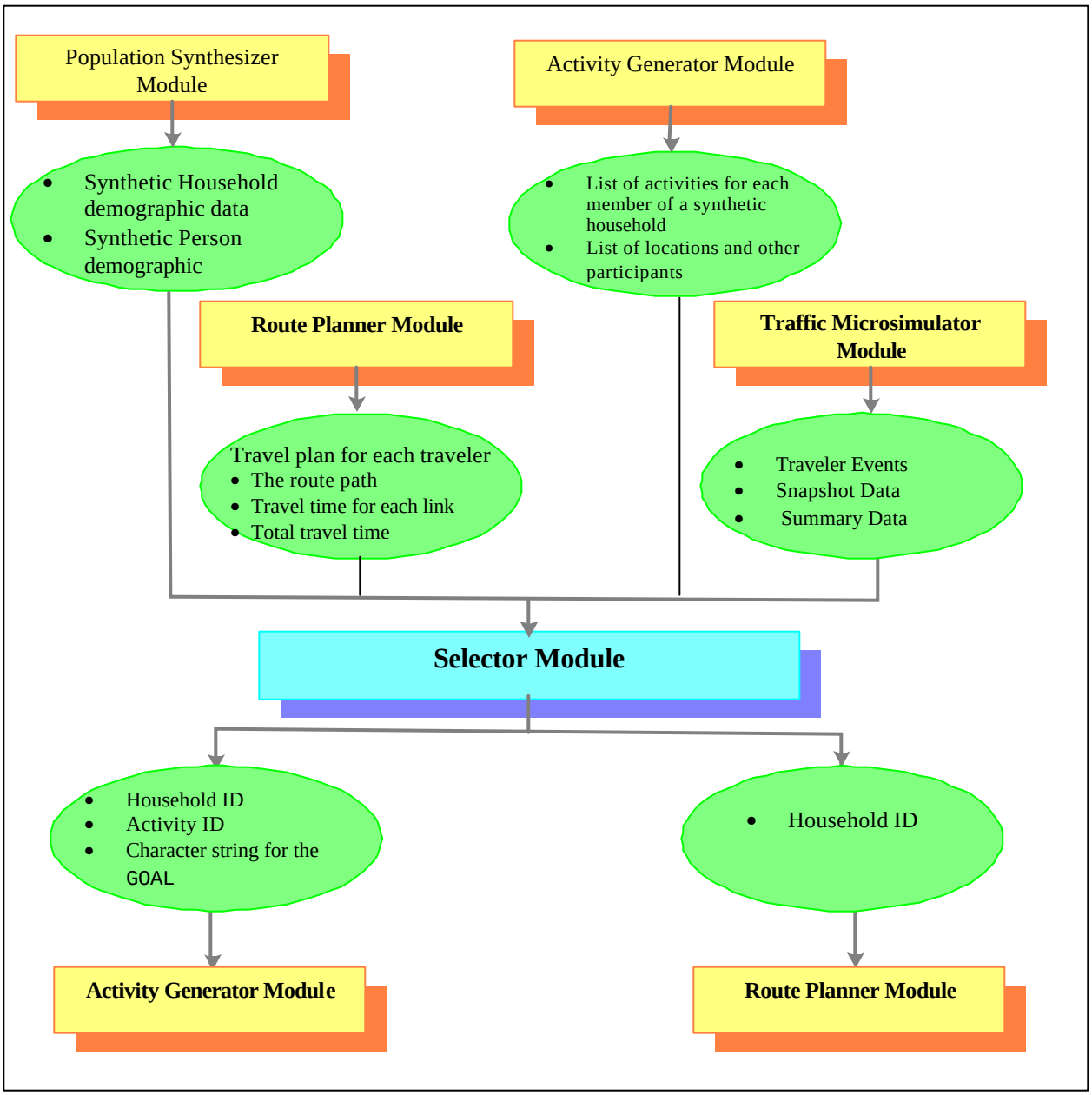


Figure-3.6: Flow-chart of the Module Interfaces for the Feedback Module.

3.7.3 Inputs received from the Route Planner Module

All of the output data of the Route Planner Module can be used as inputs to the Selector Module.

The Route Planner Module develops the route plans based on the demand represented in the Activities data file. The major outputs of the Route Planner for each traveler are the information about the travel plans, which comprise the route path (nodes, links, and travel modes), the travel time for each link, and the total travel time. In addition, TRANSIMS displays the non-transportation activities in the output as well, in order to maintain a record for every activity for each member over the 24-hour horizon.

Generally, the outputs are formatted to show the relevant information for *each leg* on a trip. A leg must start and end at an activity location, a parking location, or a transit stop (notice that there is a special leg for the non-transportation activity that starts and ends at the same location).

3.7.4 Inputs received from the Traffic Microsimulator Module

All of the output data of the Traffic Microsimulator Module can be also used as inputs to the Selector Module.

There are three major categories of output from Traffic Microsimulator. The *Traveler Events* provide information whenever an event occurs for a traveler, such as trip ID, leg ID, time, location, or anomalies. The *Vehicle Snapshot data* gives the positions of vehicles on links, at intersections etc., recorded by every time step or less frequently as desired. *Summary data includes* (*Spatial* – data collected over sections of roadway such as flow, density etc. and *Temporal* – data such as travel time over links). Summary data is sampled, accumulated and reported periodically throughout the simulation. These three broad data types capture most kinds of output a user might find necessary for analysis.

3.7.5 Outputs sent to the Route Planner Module

The households identified to be re-routed need to be sent to the Route Planner Module through Route Planner feedback files. In addition, it is likely that each household for which an activity has been changed should be replanned for a new route. There is only household ID recorded in the Route Planner feedback files. The Route Planner feedback files should be concatenated, sorted, and duplicate lines removed so that each household appears only once.

3.7.6 Outputs sent to the Activity Generator Module

The households identified that their activities need to be re-generated need to be sent to the Activity Generator Module through Activity Generator feedback files. Household ID, activity ID, and a character string are recorded in Activity Generator feedback files.

3.8 Model Tools

Feedback processes are important in the framework of TRANSIMS. Models are developed from a series of feedback loops between TRANSIMS modules that changes the behavior of selected individuals of the synthetic population. Besides, feedback is necessary to stabilize the traffic.

Feedback is used to solve the problems with the current travel plans, to consider new travel options, and to stabilize the traffic. To fulfill the above objectives, Feedback Module adjusts travel activities by changing:

- Travel path;
- Mode of travel;
- Activity location;
- Activity schedule, and
- Number of activities.

Users determine who to change, what to change, and how to implement the change using the scripts. Feedback rules and decision weights may change based on the type of

study. As stated earlier, Feedback is a very flexible module and all the rules should be determined by the user. Users should also define the convergence criteria and decide when the Feedback module should stop iterating. Feedback is usually stopped when no significant changes occur after several iterations.

TRANSIMS provides three tools for choosing the sets to be acted on:

- Collator
- Stratifier
- Selector

Basically, the Collator collects input and output data the user is interested in for each trip and creates a Collator Iteration Database. In the process, all data are translated from diverse TRANSIMS-formats into a simple, but possibly large, ASCII, comma-separated format. The Stratifier allows the user to group trips in the Collator Iteration Database based on criteria the user sets, called Discretization. The criteria are added to the database and thus a Stratifier Iteration Database is generated. The Selector chooses all or part of trips of some groups for feedback from Stratifier Iteration Database, according to users' intention. Rudimentary database queries and reporting system are implemented in Stratifier and Selector. Figure-3.7 shows the relationship between the three model tools.

3.8.1 Collator

The Collator can gather data from some input files of TRANSIMS, i.e., Network file and Transit Route file. It can also extract data from output files of Population Synthesizer, Activity Generation, Route Planner, and Traffic MicroSimulator. It can calculate values based on data from several different sources: for example, it uses the activity location table together with an activity file to calculate Euclidean distance between activities.

The Collator creates a database with one record for each trip made by each traveler from every household specified in its household file, or in the population file, if no household file is specified.

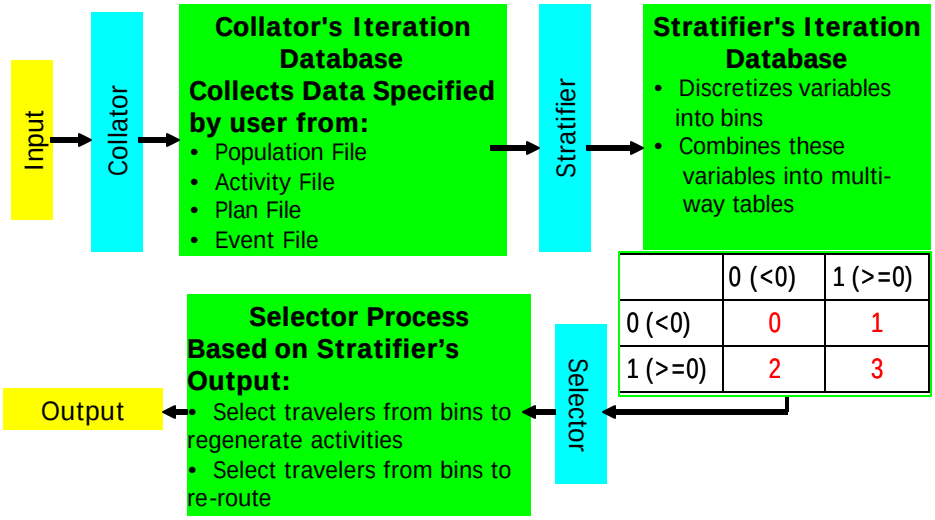


Figure-3.7: Relationship between the Three Model Tools.

Each record contains a few pre-determined fields and as many other fields as the user has requested. Fields that are present in every record are: `household id`, `traveler id`, `trip id` (as it would appear in the traveler's plan), `starting` and `ending activity ids` (as they appear in the activity file), and `tour` and `subtour id`.

The user requests fields by turning on any of the configuration file keys shown in the Configuration Files part in this chapter.

If the data for a given field is not available—for example, if there is no plan corresponding to a particular trip—that column will be blank in the database. If the data cannot be calculated—for example, if a denominator is zero—an "NA" will appear in that column. Records with blank or "NA" values are not used in Stratifier or Selector calculations that rely on that field.

After the run of Collater, if the user wishes to analyze the data by tour rather than by trip, he/she could use text processing tools to aggregate the data and prepare a new file

for later use by the Stratifier and Selector. The only requirements imposed by the software are:

- Two lines of header information, the second of which lists all the field names
- Fields are comma-separated ASCII text;
- The fields `HH`, `TRAV`, `TRIP`, `START_ACT_ID`, and `END_ACT_ID` are required.

The Collator output can be indexed if desired. If the user requests an index, Collator Iteration Database from previous runs will be kept for the current invocation. If the underlying data files have not changed since the most recent Collator run or if data is missing from the data files, the most recent value from a previous database will be used. If an index is present, certain fields can be added to the database, such as a number indicating the last iteration on which a variable changed. Indexing can be time-consuming, and is not often necessary.

Collator Example

In this example, we have two tasks. One is to find all trips that start in region 1 and end in region 2 or 3, with a duration of 20 minutes or less. The other is to find all the trips made by people older than 18 years old.

To implement these two tasks, we need to collect the information about whether the trip crosses a river, the duration of the trip, the age of the traveler and the ratio of expected travel times in modes ‘w' and 'c' with Collator.

First, we create a set of polygons that distinguish the two sides of a river. These are stored in the file `$TRANSIMS_ROOT/network/rivers.polygons`, called a "User Analysis Zone" or UAZ file by the Collator. In our case, there are two rivers that merge and thus three polygons as shown if Figure-3.8.

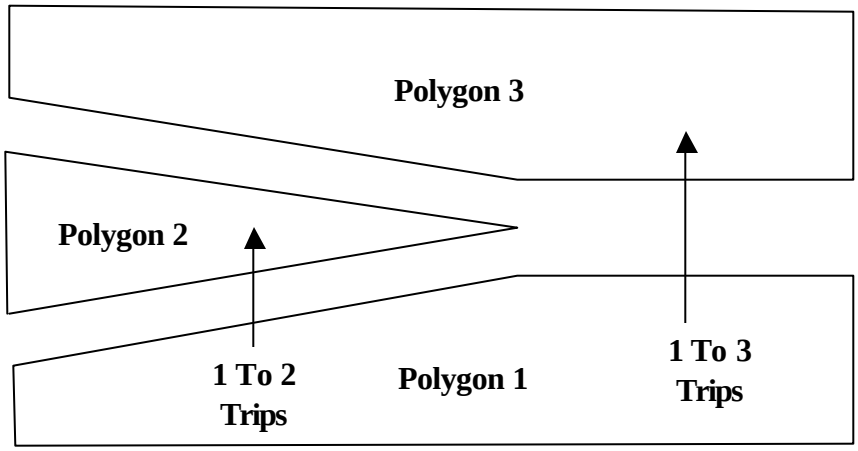


Figure-3.8: Three Polygons and Two Kinds of Trips Mentioned in the Collator Example.

To find out whether a trip's origin and destination are in different polygons, we add the following keys to a configuration file:

```
    SEL_UAZ_FILE_1
$TRANSIMS_ROOT/network/rivers.polygons
    SEL_USE_CROSS_BOUNDARY   1
```

If we wanted to be more specific about which polygonal region the trip started or ended in, we could also add:

```
    SEL_USE_START_REGION     1
    SEL_USE_END_REGION       1
```

Finally, if we are only interested in trips that start and end in specific polygons, say starting in polygon 1 of UAZ file 1 and ending in polygon 2 or 3 of UAZ file 1, we could use the following configuration file keys:

```
  SEL_USE_START_IN_REGION    1, 1
  SEL_USE_END_IN_REGION      1, 2; 1, 3
  SEL_USE_AND                START_IN_REGION_1_1,
END_IN_REGION_1_2; START_IN_REGION_1_1, END_IN_REGION_1_3
```

The SEL_USE_AND configuration file key creates two fields, each of which is the logical AND of its arguments. In this case, the first field is true if, and only if, the trip starts in polygon 1 of UAZ file 1 and ends in polygon 2 of UAZ file 1.

In this example, we have included all of these different REGION and BOUNDARY configuration file keys.

Only the START_IN_REGION, END_IN_REGION, and AND configuration file keys are required for what follows.

To find the expected travel time (from the Plan file), we add the following configuration file key:

```
SEL_USE_DURATION   1.
```

For the age of the traveler and the ratio of expected travel times in modes ‘w' and 'c', we use the following configuration file keys:

```
SEL_USE_AGE       1
SEL_USE_T_MODE    w; c
SEL_USE_RATIO     T_MODE_c, T_MODE_w
```

Running the Collator with these configuration file keys on 10,265 households out of a full Portland Activity and Plan set takes about 2.5 hours with a 400 MHz Sun SPARC processor. The Collator can be run in a distributed fashion, with each processor handling a different set of households. The resulting 20 Megabyte database contains 258,000 records with 20 fields per record. Figure-3.9 shows a sample of the Collator output database for these configuration file keys.

HH	TRAV	TOUR	SUBTOUR	TRIP	START_ACT_ID	END_ACT_ID	1 AGE	2 START_IN_REGION_1_1	3 END_IN_REGION_1_2	4 END_IN_REGION_1_3
2	4	0	0	1	1	1	59	TRUE	FALSE	FALSE
2	4	1	0	2	1	2	59	TRUE	TRUE	FALSE
2	4	1	0	3	2	2	59	FALSE	TRUE	FALSE
2	4	1	0	4	2	3	59	FALSE	FALSE	FALSE
2	4	1	0	5	3	3	59	TRUE	FALSE	FALSE
2	4	2	0	6	3	4	59	TRUE	FALSE	FALSE
2	4	2	0	7	4	4	59	TRUE	FALSE	FALSE
2	4	2	0	8	4	5	59	TRUE	FALSE	FALSE
2	4	2	0	9	5	5	59	TRUE	FALSE	FALSE
2	5	0	0	1	6	6	56	TRUE	FALSE	FALSE
64	198	0	0	1	1	1	63	TRUE	FALSE	FALSE
64	198	1	0	2	1	2	63	TRUE	FALSE	FALSE
64	198	1	0	3	2	2	63	TRUE	FALSE	FALSE
64	198	1	0	4	2	3	63	TRUE	FALSE	FALSE

5 START_REGION_1	6 END_REGION_1	7 CROSS_BOUND_1	8 DURATION	9 T_MODE_w	10 T_MODE_c	11 *RATIO_T	12 AND_START_IN*(1)	13 AND_START_IN*(2)
1	1	FALSE	28688	0	0	NA	FALSE	FALSE
1	2	TRUE	1012	46	966	21	TRUE	FALSE
2	2	FALSE	31385	0	0	NA	FALSE	FALSE
2	1	TRUE	1018	46	972	21.13043	FALSE	FALSE
1	1	FALSE	4672	0	0	NA	FALSE	FALSE
1	1	FALSE	581	46	535	11.63044	FALSE	FALSE
1	1	FALSE	2479	0	0	NA	FALSE	FALSE
1	1	FALSE	583	46	537	11.67391	FALSE	FALSE
1	1	FALSE	57900	0	0	NA	FALSE	FALSE
1	1	FALSE	99242	0	0	NA	FALSE	FALSE
1	1	FALSE	27197	0	0	NA	FALSE	FALSE
1	1	FALSE	410	46	364	7.913044	FALSE	FALSE
1	1	FALSE	36222	0	0	NA	FALSE	FALSE
1	1	FALSE	408	46	362	7.869565	FALSE	FALSE

*RATIO_T_MODE_c_T_MODE_w
*(1) AND_START_IN_REGION_1_1_END_IN_REGION_1_2
*(2) AND_START_IN_REGION_1_1_END_IN_REGION_1_3

Figure-3.9: Example Iteration Database.

Notice that "NA" appears for the value of a field when it requires division by 0. In general, inappropriate fields are either left blank or given an "NA".

3.8.2 Stratifier

The Stratifier groups trips in two steps:

1. Discretize variables, i.e., group trips according to a specific criterion. For example, one criterion could be age of 18. Therefore, the trips made by people older than 18 years old fall into one group, whereas trips made be people younger or equal to 18 years old fall into another group. Another criterion could be to group trips based on whether trip duration is shorter than 1200 sec or not. Each discretized variable is

known as a "binning", and each binning is given a numeric identifier. 'People older than 18' and 'trip duration shorter than 1200' are two binnings.

2. Combine binnings into multi-way tables, i.e., group trips based on the combination of multiple criteria. In the previous example, we will group trips according to both binnings and get a 2-way table. Each binning can be used in any number of different tables. Each table is called a "stratification", and each stratification is given a numeric identifier. Furthermore, each cell in the k-way table is given an index.

Figure-3.10 shows an example of the two steps employed by the Stratifier.

…	TRIP	…	age bin (0/1)	duration bin (0/1)
	1		1	1
	2		0	1
	3		1	0

STRATIFICATION

Age

		0	1
Duration	0	0	1
	1	2	3

			0	1	
…	TRIP	…	age bin	duration bin	STRAT-0
	1		1	1	3
	2		0	1	2
	3		1	0	1

Figure-3.10: An example Showing the Two Steps in STRATIFICATION.

Discretization is accomplished in any of the following three ways;

- Automatically which is the best for fields that already contain discrete data with only a few different values. Each different value is assigned a numeric bin id.
- Manually, specifying the number of bins. The data in a field is sorted and placed into the user-specified number of quantiles. If the data is discrete and heavily concentrated on a few values, the algorithm may reduce the number of quantiles.
- Manually, specifying the bin boundaries.

The user chooses one of these methods for each binning using configuration file keys. The boundaries for each binning are listed in the header of the Stratifier's output database.

The Stratifier's output database contains one record for each record in the Collator's database. Each record contains the pre-determined fields `household id`, `traveler id`, `trip`, and starting and ending activity for the trip. It also contains one field for each binning specified and one field for each stratification. The stratification field's value is the index of the cell into which this record falls.

Stratifier Example (cont'd on the Collator Example)

In the first step, we want to group trips based on whether they start in region 1 and end in region 2 or 3 or not, and whether the trips have a time duration of 20 minutes or less or not, and whether these trips are made by people older than 18 years old or not respectively. We will add binnings that address each of these categories to the database.

1. Specify a binning for the `DURATION` field into two bins with a boundary of 20 minutes (= 1200 seconds):

```
SEL_BIN_NAME_0       duration_bin
SEL_BIN_FIELD_0      DURATION
SEL_BIN_BOUNDS_0     1200
```

2. Create separate binnings for trips that start in region 1 and end in region 2 or 3. Since these are Boolean fields, we need not specify bin boundaries.

```
SEL_BIN_NAME_1       1_to_2
SEL_BIN_FIELD_1      AND_START_IN_REGION_1_1_END_IN_REGION_1_2

SEL_BIN_NAME_2       1_to_3
SEL_BIN_FIELD_2      AND_START_IN_REGION_1_1_END_IN_REGION_1_3
```

3. Bin trips by travelers' ages:

```
SEL_BIN_NAME_3       age_bin
SEL_BIN_FIELD_3      AGE
SEL_BIN_BOUNDS_3     18
```

In the second step, we create three k-way tables, or stratifications, which we will use in the selection process. The first two stratifications are about whether a trip starts in region 1 and ends in region 2 or 3, with a duration of 20 minutes or less. The third one is about whether the traveler is older than 18 years old.

The first two stratifications are a pair of two-way tables: one from binnings 0 and 1; and one from binnings 0 and 2.

We will use the age binning (number 3) by itself to generate the third stratification. The configuration file key `SEL_STRAT_BINS` specifies this arrangement:

```
SEL_STRAT_BINS      0, 1; 0, 2; 3
```

The Stratifier takes less than two minutes to run on the 258,000 record database created by the Collator above. Figure-3.11 shows a sample of the Stratifier output database for the same trips shown in the Collator database sample above.

					1	2	3	4	5	6	7
HOUSE	TRAVELER	TRIP	STARTACT	ENDACT	duration bin	1 to 2	1 to 3	age bin	STRAT 0	STRAT 1	STRAT 2
2	4	1	1	1	1	0	0	1	1	1	1
2	4	2	1	2	0	1	0	1	2	0	1
2	4	3	2	2	1	0	0	1	1	1	1
2	4	4	2	3	0	0	0	1	0	0	1
2	4	5	3	3	1	0	0	1	1	1	1
2	4	6	3	4	0	0	0	1	0	0	1
2	4	7	4	4	1	0	0	1	1	1	1
2	4	8	4	5	0	0	0	1	0	0	1
2	4	9	5	5	1	0	0	1	1	1	1
2	5	1	6	6	1	0	0	1	1	1	1
64	198	1	1	1	1	0	0	1	1	1	1
64	198	2	1	2	0	0	0	1	0	0	1
64	198	3	2	2	1	0	0	1	1	1	1

STRAT_0

1 to 2 \ Duration	0	1
0	0	1
1	2	3

STRAT_1

1 to 3 \ Duration	0	1
0	0	1
1	2	3

STRAT_2

Age <= 18	0
> 18	1

Figure-3.11: Example Stratifier Database.

The first line of the file (not shown here) describes each of the binnings. There are three different types: rational, categorical, and ordinal. In a "rational" binning the elements are assumed to be floating-point numbers, and bin bounds are as indicated; in a "categorical" binning, the elements take on one of the few indicated discrete values, each of which is one bin; in an "ordinal" binning, they take on all integer values between the indicated upper and lower bounds.

The Stratifier adds the same `household`, `traveler`, `trip`, and `start and end activity` fields as the Collator. It does not add `tour` or `subtour` information.

3.8.3 Selector

Basically, there are two steps for Selector to pick up a subset of travelers in the Stratifier Iteration Database.

1. Choose cells from stratifications. The cells can be picked by index (using the PICK_CELL selection algorithm). The index of a desired cell can be determined as follows:

$$\sum_{j=1}^{k}\{i(j)\prod_{m-0}^{j-1} n(m)\}$$

where k — the total number of binnings, i.e., the number of dimensions of stratification.

j — the index of the binning

$i(j)$ — the index of the bin for the desired cell in binning j

$n(m)$ — total number of bins in binning m, $n(0) = 1$.

Figure-3.12 shows an example of choosing a cell.

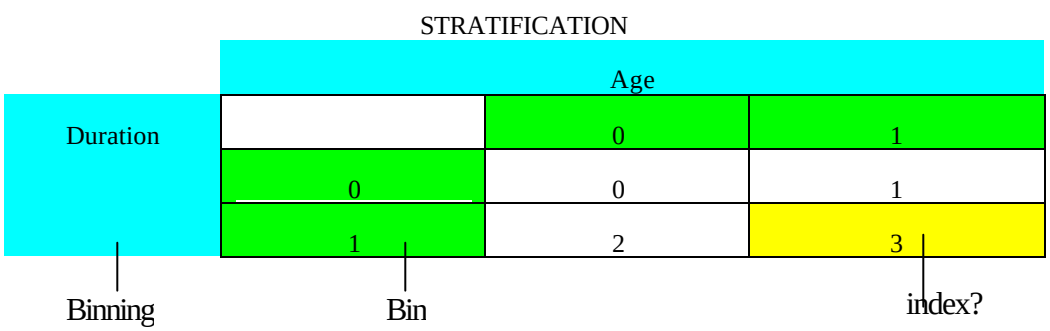


Figure-3.12: Pick Up a Cell by Index from a Stratification.

Since there are two binnings, Age and Duration, k is 2. And because there are two bins in each binning, $n(1) = 2$, $n(2) = 2$. $n(0)$ always equals to 1. As the numerical identifier for the cell in Age binning is 1, $i(1) = 1$. $i(2) = 1$, based on similar reason. Therefore, we can get the index of the cell according to the formula, which is coincident with what we got before.

$$\text{Index} = i(1) + i(2) * (n(0) * n(1) = 1 + 1 * 2 = 3$$

It is probably worthwhile for the user to verify that the cell index used in fact corresponds to the desired one by looking at the values of variables of interest for a traveler assigned to that cell.

2. Select a subset of travelers or trips within the cell. There are three ways to do that:

- The user can specify that all elements of the cell be chosen.
- Elements can be chosen uniformly at random.
- Elements can be chosen based on the cost function. Usually, each entry in a stratification can be associated with a cost function, such as travel time. So the user can pick the cell that has the highest or lowest mean travel time; or the cell that has the largest or smallest standard deviation or range in travel times.

A single run of the Selector can be used to select many different subsets of trips or travelers—each stratification in the Stratifier's output database can be associated with a cost function one or more times.

There are two possible goals for selection. One is to identify activities to be changed using the Activity Regenerator. The other is to identify households to be re-routed.

The output of Selection on each subset will consist of one Activity Feedback file and one Route Planner Feedback file. The Activity Feedback file is then sent to the Activity Regenerator, and the Route Planner feedback file is sent back to the Route Planner. The user is free to use either or both of them according to his/her intention.

In the Activity Feedback file, a traveler id and an activity id are recorded. For each selection, the user specifies a "goal". The corresponding character string from the `GOAL` configuration file key is also included in the Activity Feedback file. In the Route Planner

Feedback file, there is only a household id reported, as we always need to re-plan the whole household.

Each selection creates a different pair of Activity Regenerator and Route Planner feedback files. If we make several selections at the same time, we will get multiple Activity Regenerator feedback files and Route Planner feedback files. The Activity Regenerator files should be concatenated so that all activities for a given household can be updated simultaneously. The order in which commands are read by the Activity Regenerator is significant. The user should be careful to concatenate Activity Regenerator files in the order s/he wishes the Regenerator commands to be applied. The Route Planner feedback files should also be concatenated, sorted, and duplicate lines removed so that each household appears only once.

Selector Example (cont'd on Stratifier Example)

The following example illustrates one way of performing the desired selections. It is not unique.

As we mentioned before, we have two tasks. One is to find all trips that start in region 1 and end in region 2 or 3, with a duration of 20 minutes or less. The other is to find all the trips made by people older than 18 years old. By now, we have already divide the trips according to these criteria. Next, we will pick up trips from the Stratifier Iteration Database based on these criteria.

We can find the cell corresponding to bin 1 of the first binning (which has a total of 2 bins) and bin 1 of the second binning satisfying our criteria of the first task. This cell, according to the formula above, gives a cell index of:

$$1 + 1 * 2 = 3$$

We would like to pick up all the elements of the cell. Even though no cost function is required for the selection, the current implementation of the Selector requires one to be provided. It is best if the cost function not have any "NA" values in it, because that will

unnecessarily remove some records from consideration. Hence we choose `HH`, which is guaranteed to be defined for every record.

In practice, we would create a similar selection from the second stratification. We don't use this selection in the following example because it does not illustrate anything different from the first selection.

For the second task, we will again pick a specific cell, but not all of the elements of that cell are needed. In this case, we are using bin 1 of the only binning, so the cell index is also 1. We use the ratio of travel times in walk and car mode as the cost function:

```
SEL_USE_STRATIFICATION   0; 2
SEL_ALGORITHM            PICK_CELL 3; PICK_CELL 1
SEL_COST                 HH; RATIO_T_MODE_c_T_MODE_w
SEL_BIN_SEL_ALGO         ALL; TAIL, 1, , 1.5
```

For 1st selection *For 2nd selection*

The parameters to the "`TAIL`" within-cell selection algorithm specify that the elements with costs above 1.5 are to be chosen.

Finally, we want to assign a goal for each selection. The goal of the first task is to relocate the ending activity for each selected trip. The goal of the second task is to reroute.

```
SEL_GOAL      LS 0.01; REROUTE
```

The output of a Selector run with this configuration file will consist of four files. There will be one Activity Feedback file and one Router Feedback file for each of the two selections. The Activity Feedback file for selection 2 should be ignored because we intend only to re-route these households.

If several different Activity Regenerator commands were generated, the user should concatenate the files in the order he/she wishes the commands to be applied. For

example, it makes little sense to change an activity location after a household's activities have been regenerated from the survey, so files containing a "regenerate activities from the survey" command should be appended after all other files. Similarly, the Router Feedback household files should all be concatenated and sorted and duplicate lines should be removed.

The base filenames for Selector output are specified by two configuration file keys:

```
ACT_FEEDBACK_FILE        /home/eubank/test/feedback.act
ROUTER_HOUSEHOLD_FILE    /home/eubank/test/feedback.router
```

The actual filenames will have "_<n>" appended to them, where "<n>" is an integer indicating which selection they represent.

The Selector takes less than a minute to run on the 258,000 record database created by the Stratifier above. Here is the first line of the first Activity Regenerator Feedback file:

```
2 2 LS 0.01
```

And here is the first line of the second Router Feedback file:

```
2
```

3.9 User Equilibrium in TRANSIMS

One of the most important objectives of the Feedback module is to solve the traffic assignment problem. The simplest version of the traffic assignment process is a loop between the Route Planner and the Microsimulator modules. In the first run, the Route Planner uses free-flow speeds on each link to estimate the travel times, which is not accurate because there exist other vehicles on the link, and so, the speed is not simply equal to the free-flow speed. Therefore, some paths used might not be the shortest paths due to congestion. The Microsimulator produces the new travel times based on accurate vehicle speeds, and feeds these back to the Route Planner, which in turn determines the new routes as shortest paths for selected travelers. The Route Planner finds a time-

dependent shortest path for each activity. (TRANSIMS uses activities of each individual traveler rather than generic O-D matrices.)

The current dynamic user equilibrium process in TRANSIMS has some problems. TRANSIMS routes one person at a time but does not update link travel times. Therefore, each traveler is routed regardless of the other travelers on the network. Changing the procedure to update link travel times after routing each person is an improvement of the current dynamic traffic assignment in TRANSIMS. The current stopping criterion is based only on visualization and the existing procedure might oscillate. Also, the current traffic assignment process expends a huge amount of time by iterating frequently between the Route Planner and the Microsimulator. For example in the Portland study, 21 iterations between the Route Planner and the Microsimulator was performed that took 33:29 hours using three 500-MHZ CPUs (parallel processing). A faster mechanism is needed to assign the traffic.

The current Route Planner in TRANSIMS loads the network and changes it into an internal network, loads the link delay functions (if they exist), which is an output of the Microsimulator, and then routes each person, one at a time, using a time-dependent shortest path algorithm. In the first run of the Route Planner (without any Microsimulator output), the link delay function is based on the free-flow speed on each link. A time-dependent shortest path algorithm utilizes the travel time on each link during the desired time period to find the shortest path given the starting time for each activity. Since the link travel times are not updated after routing each person, some links might get very congested and this might entail several additional iterations to redistribute the traffic to dissipate this congestion.

3.10 Summary and Conclusion

The Feedback module in TRANSIMS, its function, the inputs and outputs of this module, and the module tools have been explained in detail in this chapter. The feedback procedure to find dynamic user equilibria in TRANSIMS and the drawbacks of this procedure has been presented as well.

Since the current procedure to compute dynamic user equilibria for large-scale

transportation networks is not convergent in nature and a near-equilibrium is not guaranteed to be obtained, a new procedure to solve dynamic user equilibrium problem is needed. Such a procedure is proposed in Chapter 5.

Chapter 4

The TRANSIMS Dynamic Traffic Assignment Model

4.1 Introduction

Dynamic traffic assignment (DTA) models find time-dependent link flows in a transportation network. Many researchers have developed related models for more than three decades. Nevertheless, these models, which are an extension to the static traffic assignment models, are still relatively unexplored and lack a proper formulation due to their complexity as compared with the static models. Therefore, developing dynamic traffic assignment models that are applicable to the large-scale transportation networks are still of interest to researchers. A comprehensive review of the literature in this area is outside the scope of this paper, but is presented in Peeta and Ziliaskopoulos (2001).

There are several computer packages that address the time-varying (dynamic) traffic assignment problem by using a simulator to model complex traffic interactions. Some well-known packages existing in the market are DYNASMART, DynaMIT, CONTRAM, and TRANSIMS. In contrast to the other packages that are developed mainly for traffic operations and real-time applications but they can be used for planning process as well, TRANSIMS is created for planning purposes only. Among these packages, TRANSIMS has not received a good exposure and its capabilities are unknown to many researchers in the transportation field.

The approach adopted by TRANSIMS is different from that in other packages in the way it handles the demand and supply aspects of transportation. TRANSIMS introduces new concepts and models to deal with the demand and supply components of transportation networks. It develops and traces the movements of each individual person, and simulates these movements microscopically on a fully described network. TRANSIMS is not just a model, it is a modeling framework. Transportation researchers and modelers could implement their own models within the framework of TRANSIMS by exploiting its flexibility, and then employ it for different transportation purposes. TRANSIMS is developed on a Linux platform using parallel processing. IBM is currently developing the commercially viable version of TRANSIMS. Training practitioners on the use of TRANSIMS as well as refining the theories and methods utilized in this package have been conducted by Virginia Tech, which has played a leading role in the education and enhancement of the current version of TRANSIMS.

The objective of this chapter is to introduce TRANSIMS, its capabilities and its shortcomings; to describe how TRANSIMS deals with the dynamic traffic assignment problem in the planning context; to state the benefits of TRANSIMS over the planning version of other well-known existing packages; and to introduce some of the improvements conducted by researchers at Virginia Tech. While TRANSIMS is more than a traffic assignment tool, we focus in this chapter on its capabilities in the context of dynamic traffic assignment for planning purposes.

4.2 Fundamental Characteristics of TRANSIMS

The purpose of urban travel demand modeling is to provide tools to forecast urban travel patterns under various conditions. The predicted travel patterns then provide useful information in planning the transportation system. The classical or traditional urban travel demand modeling that has been used over forty years is called the "four-step" model. These models have been used widely in urban transportation planning. Although they are good practical models and have been used for decades, they have several shortcomings (see Oppenheim, 1995). The main shortcoming has been in the traffic assignment model that uses macroscopic equations to predict the performance of the network under various demands. This traffic assignment does not consider vehicular spill-over of queues from one link to another, and it statically assigns the volumes on all the links without considering time aspects. Hence, transportation researchers have shifted their attention to dynamic traffic assignment instead.

TRANSIMS (TRansportation ANalysis and SIMulation System) has been developed at the Los Alamos National Laboratory for transportation planning (See TRANSIMS 3.1 Documentation, 2003) as part of the Travel Model Improvement Program (TMIP), a multiyear, multi-agency program. It is designed to improve both the analytical tools and the integration of these tools into the planning process. In essence, TRANSIMS is a microscopic large-scale transportation system framework. It is an integrated system of travel forecasting models designed to give transportation planners accurate and complete information on traffic impacts, congestion, and pollution. TRANSIMS consists of the

following modules, all of which are integrated with respect to their input and output, as presented in Figure-4.1.

Population Synthesizer: Estimates the number of synthetic households, the demographics and characteristics of each individual within these households, and the locations of these households on the network.

Activity Generator: Creates an activity list for each synthetic traveler. These activities include home, work, shopping, school, etc. These activity estimations are based on the activity demographic characteristics of individuals, and available survey data. Furthermore, activity times and activity locations are determined for each individual.

Route Planner: Computes combined route and mode trip plans to accomplish the desired activities of each individual, such as work, shopping, etc.

Microsimulator: Uses the intermodal paths developed in the Route Planner module to perform a regional microsimulation of vehicle interactions. The microsimulation continuously computes at every second the operating status, including locations and speeds, of all vehicles throughout the simulation period. The output can provide a detailed, second-by-second history of every traveler in the system over a 24-hour period.

Feedback Controller: Manages the feedback of information among the Activity Generator, the Route Planner, and the Traffic Microsimulator modules of TRANSIMS.

Emissions Estimator: Using the vehicle information generated in the Microsimulation module, the emission module forecasts the nature, amount, and location of motor vehicle emissions.

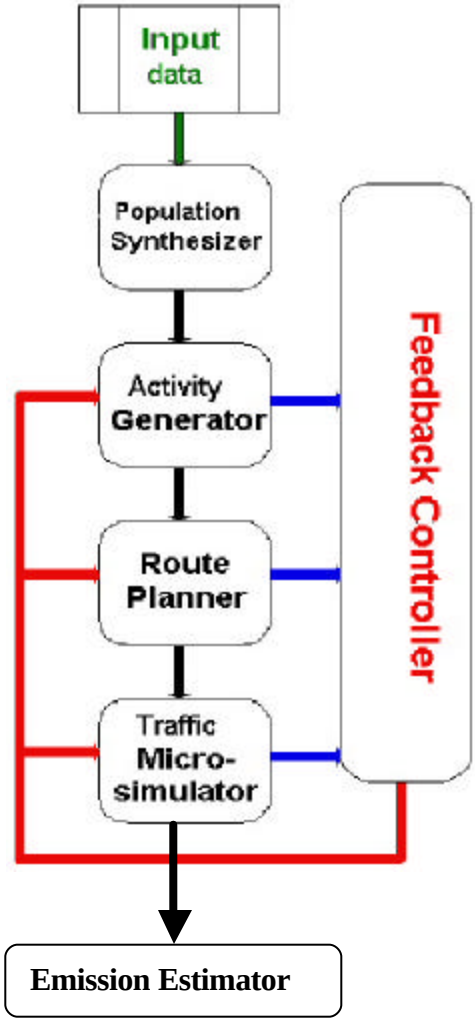


Figure-4.1: Framework of TRANSIMS

TRANSIMS contains many significant advantages beyond the four-step models. It is disaggregated and tracks individuals, households, and vehicles. Each traveler's, household's, and vehicle's identity is maintained throughout the demand and supply analyses. Therefore, it is easy to trace individuals through the entire process. Hence, TRANSIMS is referred to as a "people mobility" modeling system. TRANSIMS uses a microsimulator to simulate vehicle interactions on a second-by-second basis to produce vehicle operating conditions, operating speeds, and intersection operations. It also estimates the tailpipe emissions based on the operation of individual vehicles as they interact on the transportation network.

CONTRAM, DYNASMART, and DynaMIT employ mesoscopic simulation models in contrast with TRANSIMS' microscopic approach. Due to computational restrictions, early simulators were macroscopic in nature. Macroscopic models move vehicles based on aggregate equations and therefore provide only gross approximations. Although macroscopic simulators partially address the traffic modeling needs of freeways, a greater level of detail may be required for surface street networks, especially at intersections. Therefore, mesoscopic and microscopic simulators were introduced. Mesoscopic models relate speed and concentration on a section using macroscopic equations, but move vehicles individually or in packets. On the other hand, Microscopic models keep track of vehicles second by second and track details of individual drivers and vehicles and the interactions between them. They are more accurate than the mesoscopic models, but they entail large computational requirements. There is a tradeoff between precision and computational speed in utilizing mesoscopic or microscopic models. While other packages employ mesoscopic models, TRANSIMS takes advantage of both precision and speed by employing parallel processing in a microscopic framework. Furthermore, using a cellular automata (CA) approach (Nagel et al., 1998c; Simon and Nagel, 1998), simulating large numbers of vehicles and maintaining a fast execution speed is feasible. Using this approach, the traffic Microsimulator provides the computational speed necessary to simulate an entire region at the individual level.

4.2.1 Demand Estimation and Prediction in TRANSIMS

In planning applications, the most challenging work is to estimate and predict the time-dependent origin-destination (O-D) demand matrices. These matrices, which are used as an input in traffic assignment models, are computed within a demand estimation modeling framework. The existing demand models do not provide satisfactory methods for estimating time-dependent O-D demands. While there are a few studies related to time-dependent demand estimations (Peeta and Ziliaskopoulos, 2001), there is no framework in the literature to produce time-dependent demand data. The existing models usually extend the static demand estimation models. The most commonly used static demand estimation models are trip generation and trip distribution in the four-step

models. The four-step models determine the 24-hour O-D matrix and then discretize it by the time of day into peak and off-peak periods.

The planning version of DYNASMART (DYNASMART-P) and DynaMIT (DynaMIT-P), and also CONTRAM obtain the time-dependent O-D demand as an input from actual period counts or from a time-sliced demand model. Since there is no realistic time-dependent O-D demand to be fed into these packages, the reliability of their results is questionable. The approach adopted by TRANSIMS for demand estimation and prediction is different from that of other models. It addresses the existing problems in the four-step demand estimation models and introduces a framework to produce time-dependent demand data for individual travelers for a 24-hour period using the census data, land-use data, and an activity survey of travelers. The trip generation and trip distribution aspects within the four-step models and the demand estimation implemented in TRANSIMS are briefly explained next.

Trip generation determines the number of trip productions and attractions associated with a given set of activities in a zone. There are three types of trips or trip purposes: home-based-work (HBW), home-based-other, and non-home-based (NHB). Trip generation does not consider trip chaining among trip types, which causes a misrepresentation of some types of trips. The complex nature of travelers' behavior is not considered in trip generation. Using the number of trip productions and trip attractions by trip types, trip distribution determines the number of trips between each origin-destination pair by trip types. In short, the four-step models provide a 24-hour O-D matrix, which is then distributed into three O-D matrices, morning peak period, evening peak period, and off-peak, using time of day splitting factors. This kind of O-D cannot be considered as dynamic because there are different O-D rates during each of the above periods.

To capture household travel behavior accurately, a time-dependent activity-based demand modeling is proposed in TRANSIMS. Since TRANSIMS is based on the movement of individual travelers between activities at different locations, it creates a synthetic population that represents every individual on the network. TRANSIMS uses an algorithm developed by Beckman et al. (1996) to generate synthetic populations from

available census data for a given planning year, using the Population Synthesizer module. Land-use data is then used to place individual households at activity locations along the transportation network, where, an activity location represents a place that a household member would travel to and from. TRANSIMS uses activity location rather than zones to represent travels, which is more precise and more detailed. A zone could contain several different activity locations. These activity locations could be residential areas, business places, schools, and so on. Non-home activity locations reflect land-use and employment data obtained from the metropolitan area.

The Activity Generator module generates a list of activities for each member of a synthetic population. To create the list of activities, the Activity Generator performs the following: (a) it creates skeletal activity patterns from the survey households; (b) uses the CART (Classification and Regression Tree) algorithm to build a classification tree based on household demographic data; (c) matches each synthetic household with a survey household, (d) generates activity times and durations, and (e) generates non-home activity locations.

TRANSIMS uses the CART algorithm to produce a classification of household demographic characteristics, which are designated as independent variables, based on household travel behavior indices serving as dependent variables. Each survey household is effectively classified by the CART algorithm into one of the tree's terminal nodes representing the household demographic characteristic. This tree is sensitive to the household travel behavior characteristics but is parsimonious with respect to household characteristics that do not affect travel behavior.

Then, a discrete choice model is used to select an appropriate zone for each activity. Using a variety of activities such as home, work, school, college, shop, visit, serving passenger, and other; a trip chain between activities is suggested in TRANSIMS. Each individual in the household has a chain of activities; and each activity consists of activity type and its priority, starting and ending time preferences, a preferred mode of transportation, a vehicle preference (if appropriate), a list of possible locations for an activity, and a list of other participants (if the activity is "shared-ride"). The set of activities for each household is based on a household's demographics. Consequently,

TRANSIMS develops a time-dependent activity-based demand estimation and forecast which is unique in this arena.

Activity-Based Modeling

Activity-based modeling of urban travel behavior is derived from the principle that the travel demand is based on the need to participate in non-home activities. This realization has lead to a shift in the study of urban travel behavior from trips to activities assuming that travel is not an end in itself but is a bridge between activities that are separated in time and space. The activity-based modeling requires information about individual travelers and households rather than aggregated data.

Activity-based models analyze households as the decision making units, inspect detailed timings and durations of activities, and focus on a sequence of patterns of behavior rather than on discrete trips. Early reviews of the state of activity-based research points out to the lack of data required to implement activity-based models (Pas, 1985). However, the progress in standing activity/travel behavior has increased this data availability.

The choice of an activity-travel pattern that has space-time components is a complex cognitive task. Hence, cognitive theory has been used by some researchers to determine activity travel patterns. This cognitive theory represents forms of imperfect choice behavior that are not covered by current utility-maximization models in economic theory. There are several operational activity-based models that utilize computational process models (CPM). Examples of these models are PCATS (Kitamura and Fuji, 1998) and SMASH (Ettema, 1993). These models use logit models or other forms of algebraic equations to predict single-facet choices. Nevertheless, fully operational rule-based models have not been as yet developed due to the lack of a method for empirically deriving rules of a production system.

Decision trees provide an alternative to address the existing problems in the CPM. Methods to construct decision trees from data are available from statistics and artificial intelligence. In decision tree models, a tree is developed by recursively splitting a sample of observations into relatively more homogenous groups. Arentze et al. (2004) proposed

an operational computational process model based on decision tree methods. TRANSIMS also uses a decision tree method, namely, the CART algorithm. This multivariate regression tree program can help construct a binary tree for matching synthetic households to survey households.

4.2.2 Transportation Supply in TRANSIMS

Transportation supply refers to the transportation network and its associated traffic control system. Since most existing models and packages provide aggregated O-D demand information based on interchanges between different zone centroids, their network structure utilizes zonal centroids along with virtual links that connect these centroids within the network. These virtual links are usually called zone connectors. The network road types are limited to three-to-five categories: freeway, ramp, and urban road in DynaMIT; and freeway, highway, on-ramp, off-ramp, and arterial in DYNASMART. TRANSIMS uses activity locations within the street network instead of zones. Hence, all roads in the network from small lanes to freeways are represented. Road types are therefore broader as compared with other packages, and include local streets, collectors, ramps, freeways, expressways, primary arterials, secondary arterials, light rails, heavy rails, ferries, and walkways. The TRANSIMS Network representation provides detailed information about streets, intersections, signals, and transit modes within the road network. Furthermore, detailed network characteristics that exist in the real-world such as pocket lanes, merge lanes, parkings, bus stops, link grades, and lane use restrictions are all incorporated in TRANSIMS, unlike other packages.

Signal cycles and random variations in traffic cannot be modeled explicitly in CONTRAM. Although DynaMIT-P and DYNASMART-P account for signal control, intersection movement delay, ramp metering, and so on, they might not be very precise due to the mesoscopic modeling paradigm. This capability is accommodated more accurately in TRANSIMS using microscopic modeling.

4.2.3 Outputs from TRANSIMS

TRANSIMS provides a complete picture of traffic in a fully described network in which all results are derived from the microscopic model. It produces traveler events, spatial and temporal summary data, and snapshot data. Analysts can specify the simulation period, the portion of the network to be simulated, the sampling rate, and the reporting frequency.

The event output reports each time pertaining to an event of interest to the analyst whenever if occurs for each traveler. The simulation time intervals are specified by the analyst. These events include begin and end of waiting at a location such as a bus stop, a pass through an intersection, entering a vehicle, and so on. Each specific traveler can be traced in the network because the location of each traveler in the network is reported at each time epoch in the event output. Filtering capabilities are provided in order to reduce the size of this output. The analyst can select which of the many potentially interesting events should be recorded.

Spatial summaries include data aggregated over user-defined sections of the roadways. Temporal summaries report data about travel times of links at different times of the day. TRANSIMS has the ability of producing traffic animations from snapshot files, which contain time, position, and velocity information for each vehicle on the network.

4.3 Traffic Assignment and Equilibrium

Traffic assignment is the assignment of transportation demand to transportation supply entities, based on factors that affect route choice. There are two different approaches adopted for solving the traffic assignment problem. The system optimal (SO) approach that minimizes the total system travel time over the planning horizon, and the user equilibrium approach (UE) that seeks user path assignments that satisfy Wardropian user equilibrium conditions (Sheffi, 1985) in which no user can improve his/her experienced travel time by unilaterally switching routes. Traffic assignment is classified

into two categories, static and dynamic. In dynamic traffic assignment models, the demand is allowed to be time varying so that the number of vehicles passing through a link and the corresponding link travel times become time-dependent. Dynamic traffic assignment models determine time-dependent link flows given time-dependent origin-destination (O-D) trip rates, the network, and the link performance functions. The O-D trip rates represent the transportation demand, which represent the number of trips between each origin and each destination. The network constitutes the transportation supply consisting of nodes and links. The link performance functions are positive, increasing, and convex curves that give the link travel times as functions of the respective volumes on the links. These functions could be substituted by simulation modules in order to derive more realistic link travel times. In the planning context, the user equilibrium problem is usually solved. The dynamic traffic assignment procedure employed by TRANSIMS and by other packages is described in the following two sub-sections.

4.3.1 TRANSIMS' Traffic Assignment Model

TRANSIMS performs the Dynamic Traffic Assignment using the Route Planner, the Microsimulator, and the Feedback modules. To assign the demand on the network, shortest path algorithms are used. The shortest path could be the shortest time path or the shortest cost path based on any suitably defined cost function. The shortest path algorithm utilized in the Route Planner module of TRANSIMS solves a time-dependent label-constrained shortest path problem. Here the labels refer to travel modes such as car, walk, and rail transit, and a string of admissible labels is also specified as a constraint where computing the time-dependent shortest paths. The solution method used is an extension of the classical Dijkstra's algorithm developed by Barrett et al. (1998), which uses a non-deterministic finite automata (NFA) to construct a composite graph that consists of all possible combinations of paths that satisfy the specified label constraints for the trip from an origin node to a destination node.

Each link has a delay associated with it, which can be constant as for example for walking, or might be dependent on the time of day, as for street links, railway links, etc.

The default delay for a link is the free-flow delay. More realistic delays are calculated by the Microsimulator that represent the average delay experienced by vehicles traversing a link over 15-minute intervals. Therefore, in the traffic assignment process, TRANSIMS divides the time horizon into 15-mintue time-slots. The label-constrained time-dependent shortest path algorithm finds the shortest path for each traveler based on the arrival time at each node and the link delay values for the relevant time-slot.

Besides the time-dependency, as mentioned above, the problem is multi-modal and requires the determination of a shortest path based on the permissible strings of travel modes as specified by the label constraint. Computing shortest paths on time-dependent multi-modal networks is challenging, particularly in the absence of FIFO characteristics, and very little effort has been devoted to this problem in the literature. Most existing models assign paths to vehicles rather than to travelers, and so mode-split is usually used. TRANSIMS accommodates the modeling of 12 modes: Walk, Bike, Car, Bus, Light Rail, Regional Rail, Rapid Rail, Trolley, Street Car, Transit, Magic Move-School Bus, and Magic Move-Other. The Magic mode is an un-routed mode that is used to enable the use of travel modes that are not explicitly supported by the Route Planner or the Microsimulator such as school bus mode. To support finding multimodal shortest path in networks, the Route Planner views the network as a set of interconnected, unimodal layers. Special links, called process links, are used to connect one unimodal layer to another. Hence, intermodal transitions are allowed using process links. Figure-4.2 depicts the concept of the various layers and the associated process links.

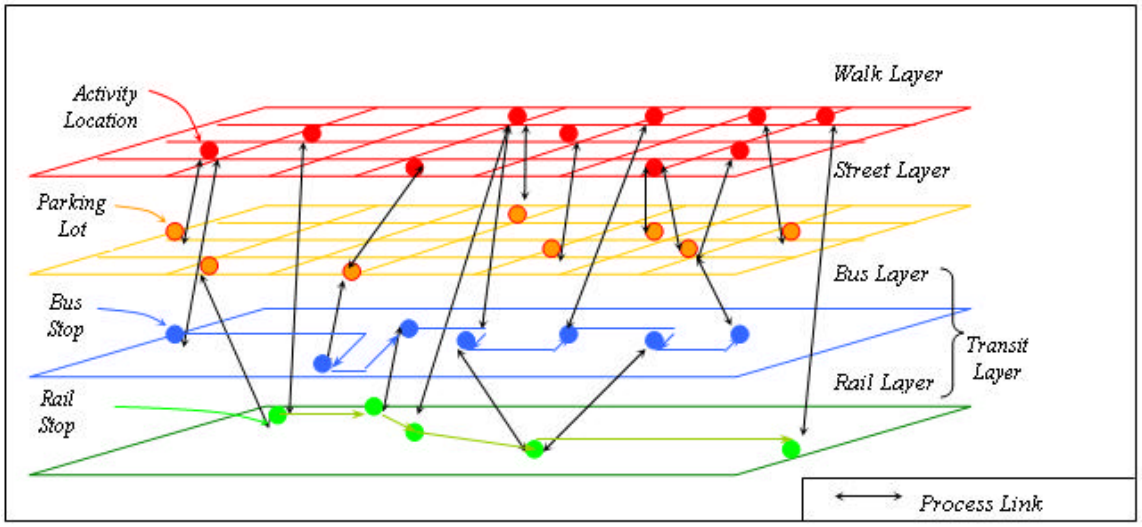


Figure-4.2: A High-level Depiction of the Various Layers Used by TRANSIMS.

To maintain computational efficiency, a heap implementation is used by the algorithm. Also, the network is converted to an internal route network that represents a weighted directed graph. To enhance performance, TRANSIMS defines an overdo parameter (between 0 and 1) that allows for a tradeoff between the running time and optimality of the paths found. The Sedgewick-Vitter heuristic (RLT) is used for overdo values greater than zero. In this method, which is applicable to Euclidean graphs, the Route Planner examines distance-based delays to focus the determination of paths that more pointedly lead toward the destination. The internal network is not strictly Euclidian because the graph is not complete. Nevertheless, it has been found that the paths produced with small amounts of overdo, e.g. 0.25, are realistic and the running time is extremely reduced.

Some modeling as well as algorithmic enhancements of the above shortest path algorithms has been done by researchers at Virginia Tech. Sherali et al. (2003) developed an effective method for implicitly working with the composite graph rather than constructing the full composite graph apriori. This model is based on the partitioned shortest path algorithmic developed by Glover et al. (1985) and its dynamic programming (DP) interpretation afforded by Sherali (1991). Furthermore, to reduce the search effort,

Sherali et al. (2003) also proposed several heuristic curtailing schemes by focusing the search to systematically proceed toward the destination from the origin while avoiding the searching of paths that are beyond a defined boundary. They developed an *Ellipsoidal Region Technique* (ERT) that examines only those paths that lie within the union of some three defined ellipsoidal regions that envelope the origin and destination, including any freeway sections along with related entrance and exit ramps. This ERT technique effectively curtailed the search and was demonstrated to find solutions within 7% of optimality while saving 33.57 % CPU time as compared with the exact method on some test cases related to the Portland, Oregon test network.

Sherali et al. (2004) also developed an approach-dependent, time-dependent, label-constraint shortest path model to accommodate turn-penalties in transportation networks, where the time spent at an intersection before entering the next link depends on whether the driver travels straight through the intersection, makes a right turn, or makes a left turn. Further heap-based implementations and refined ERT-based heuristics were shown to determine solutions within 0.78% of optimality while curtailing effort required by the exact method by 56.77%.

The traffic assignment problem in TRANSIMS is solved by the interfacing the Activity Generator, the Route Planner, and the Microsimulator modules using the Feedback module, which can be designed by the analyst by writing some scripts. Feedback loops can be performed on any two modules to clean up the demand, to calibrate the models, to perform the mode choice, and to stabilize the traffic. The traffic assignment in TRANSIMS can be executed by involving any combination of departure time choice, mode choice, and route choice models. This ability is unique, to the best of the authors, knowledge. The simplest version of the traffic assignment process, which is only a route choice model and is applied to the Portland study (2001), is a loop between the Route Planner and the Microsimulator modules. In the first run, the Route Planner uses free-flow speeds on each link to estimate the travel time, which is not accurate because there exist other vehicles on the link and so, the speed is not simply equal to the free-flow speed. Therefore, some paths might not be the shortest paths due to congestion. The Microsimulator produces the new travel times based on accurate vehicle speeds. The

new travel times computed by the Microsimulator are fed back to the Route Planner, and the new routes are determined as the shortest paths for selected travelers. The Route Planner finds a time-dependent shortest path between different activities for each individual.

This iteration between the Route Planner and the Microsimulator is repeated until a stopping criterion is met. In the Portland study (2001), the selection of people to be re-routed was made by a uniform random sampling among all households. It is stated that a targeted selection of travelers yielded results that were insignificantly better, and in some cases worse, than using a uniform random selection.

Nagel et al. (1998b) presented results of simulation studies from the Dallas-Fort Worth area on a relatively large network having 6124 links. They performed a feedback between the Route Planner and the Microsimulator to find a reasonable traffic assignment. They found that the best performing method was similar to the method of successive averaging (Sheffi, 1985). They started the re-routing fraction at 30%, and slowly decreased this to 5% by the 20th iteration. The stopping criterion used was the vanishing of deadlocks in the Microsimulation, which was realized by visualization.

Using free-flow speeds at the first iteration between the Route Planner and the Microsimulator causes a heavy congestion on some links and needs several subsequent iterations to clear this congestion. For the Portland study, (2001), this congestion turns out to be too excessive to simulate. Hence, travelers were loaded incrementally onto the network. The iteration between the Microsimulator and the Route Planner progress until a stopping criterion is met. In deterministic steady-state assignment, the process is terminated when changes are smaller than some pre-defined value and the method tends to converge to a near-equilibrium solution. Since microsimulations are neither deterministic nor steady-state, it is unclear if any described property holds true for the resulting solution. Another difficulty with the current method is that the travel time is minimized with respect to the previous iteration's solution, which could induce excessive oscillations. Therefore, the current method needs to be improved.

To address some of the above problems, Raney and Nagel (2002) implemented a modified version of TRANSIMS for all of Switzerland. They found the shortest path for each traveler and incorporated a memory of the past plans. Travelers selected their new plans based on the performance of the routes in their memory. Thus, a fraction of travelers were chosen to be re-planned based on informed decisions rather than by random selection. The selection methodology adopted is similar to that of logit models used in discrete choice theory. Their test network was comprised of 10572 nodes and 28622 links, and 7.5 million travelers and 20 million trips were studied over a 24 hour horizon. The method terminated after 49 iterations, which is very time consuming since the Microsimulator and the Route Planner are both quite computationally intensive.

In all of the above implementations, there are usually a large number of iterations performed between the Route Planner and the Microsimulator, which is extremely time consuming. These excessive iterations are necessary in order to clear the heavy congestion on some links due to the use of the free-flow speed to route travelers onto the network in the first iteration. TRANSIMS routes one traveler at a time but does not update link volumes and link travel times after each routing. Therefore, each person is routed regardless of the existence of other travelers on the network. After routing all the travelers via the Route Planner module, the Microsimulator simulates these travelers and their interactions on the network, and finds link volumes and link travel times. Then, the Route Planner reroutes a proportion of travelers using the updated link travel times. This sequence of iterations between the Route Planner and the Microsimulator continues until a stopping criterion is satisfied. Another problem is that this stopping is itself based simply on visualization, which is not precise.

We proposed an approach for determining dynamic user equilibria to improve the current DTA model in TRANSIMS, which will be explained next. The method is a two-stage process that employs a combined use of link performance functions and a microsimulator in order to design a framework suitable for application to real transportation systems. In the first stage, they used a new version of the Route Planner in which they route one traveler at a time and update the link travel times after each routing, using a link performance function. Therefore, the next traveler is routed based on the

most updated travel time, thereby precluding heavy congestion on key links. This stage is executed only once to distribute travelers well on the network from the beginning to obviate an excessive number of iterations between the Route Planner and the Microsimulator in order to re-distribute them. This also mitigates the oscillations that frequently occur in the present version of TRANSIMS. In the second stage, the modified process to find a near-equilibrium by alternating between the current Route Planner and the Microsimulator by iteratively redistributing a subset of travelers that have the highest ratio of the actual travel time (experienced in the Microsimulator) to the travel time that was calculated in the Route Planner. This method was applied to a large-scale network, Bignet, which is part of the transportation city network of Portland, Oregon. The results exhibited an improved distribution of travelers obtained while consuming less than 20-33% of the effort required by the current version of TRANSIMS.

4.3.2 Traffic Assignment Model in Other Packages

This sub-section briefly describes the traffic assignment model in CONTRAM, DynaMIT-P, and DYNASMART-P. The dynamic user equilibrium procedure adopted in CONTRAM divides the given time-sliced demand into small packets, routes them sequentially along minimum cost routes in the order of the trip start times, using the time-dependent Dijsktra algorithm and then updates the link delays using the COBA speed/flow relationship (see Taylor, 2003). Queuing is modeled on the motorways using an explicit bottleneck for congested networks. After routing all the packets, CONTRAM re-assigns them over several subsequent iterations, because each packet can influence the travel times of other packets. To re-assign each packet, it is deduced from the network, network states are updated, and then the new route is found for the packet. The packet flow is added temporarily to the link flow to calculate the expected travel time along each link. This means that a weaker definition of a user equilibrium is used: switching a path is allowed only if this switch decreases the actual resulting travel time. The procedure is terminated after a user-specified maximum number of iterations, or until the algorithm converges to an equilibrium solution, or until certain stability criteria are satisfied. Leonard et al. (1978) state that sensitivity tests revealed poor convergence properties for

this process. To remedy this, they proposed a 'fixed route' option, in which particular traffic demands are forced to choose certain specific routes, however unattractive they might be.

The CONTRAM model is dynamic and time-sensitive to each routing; therefore, packets are adequately distributed onto the network using the most updated network state. Nevertheless, the method is not convergent and the solution method to make it convergent is not realistic and deviates from user equilibrium principles. Most operations in CONTRAM adopt macroscopic traffic models which are not precise. Using the analytical COBA speed/flow relationship to find link delays is unsuitable for modeling congestion, which is corrected using a queuing model. The FIFO property is also not satisfied due to the averaging of flows over each time slice. The packet modeling approach in CONTRAM deviates from theoretical principles and is not compatible with the minimization of any objective function. The model also supports only the car mode in the network. Among the three models: route choice, mode choice, and departure time choice; only route choice is used.

DynaMIT-P is an off-line version of the real-time traffic estimation and prediction that has been developed for planning purposes. It consists of a supply simulator, a demand simulator, and an algorithm to conduct the interaction between demand and supply. The demand simulator is a microscopic simulator of traveler behavior in terms of route choice, departure time choice, and response to information. It estimates and predicts O-D demand using a Kalman Filtering method based on historical information and the drivers' response to information. Since the demand simulator is microscopic, the historical O-D matrices are disaggregated into a historical population of travelers. A vector of socioeconomic characteristics is generated by Monte-Carlo simulation based on their distribution within the actual population and is assigned to each traveler. The supply simulator is a mesoscopic simulator that captures traffic dynamics such as queues, spillbacks, and congestion. The choice of the level of aggregation of vehicles into homogeneous packets, and the choice of time steps determines the level of detail in the supply simulation. It is assumed that the only mode is car and that the departure time interval is the interval corresponding to each O-D matrix. A travel choice model is used

to route the travelers using socioeconomic characteristics such as the value of time and information source, and trip characteristics such as trip purpose. A description of the demand is found such that it matches with the link flows reflecting the supply. Therefore, this is a fixed-point problem whose solution is reached through an iterative process, and a new assignment matrix is computed at each iteration.

In DynaMIT-P, a list of travelers and their paths is generated based on their current travel times using a demand simulator that implements demand dissaggregation. Then the aggregated supply simulator is run for the entire planning horizon to provide aggregate link travel times that are experienced by the travelers. The expected travel times (toward computing equilibrium solutions) are updated by a convex combination of the travelers' expected travel times and their recent experienced travel times. The above procedure is repeated until a convergence criterion based on the expected travel times of travelers matching their experienced travel times is satisfied. This is checked by comparing the equilibrium travel times in the current and the previous iterations using a weighted norm function.

DynaMIT-P modifies the time-sliced O-D demand to reduce the effect of the unrealistic O-D demand input on the traffic assignment results. The disaggregation procedure in the demand simulation is computationally intensive because it needs to assign characteristics and initial routes to all the drivers. The drivers need to be sorted based on departure time intervals. DynaMIT-P is a tool that can be used for the evaluation of Intelligent Transportation Systems (ITS) at the planning level and for the evaluation of short-term planning projects, rather than for long-term planning purposes. It models the day-to-day evolution of traffic, traveler behavior, and network performance for special events and situations such as incidents, weather emergencies, and sporting events.

DYNASMART-P solves the dynamic user equilibrium problem and the dynamic system optimal problem for O-D demands having fixed departure times, using heuristic simulation-based iterative procedure. For the user equilibrium problem, DYNASMART-P initially finds the shortest path for the given O-D matrices using free-flow travel times to assign them onto the network. Then it iteratively simulates vehicles and finds the new

travel times and assigns the travelers based on the new travel times using the successive averages (MSA). The shortest path algorithm applied in DYNASMART is a multiple user class K-shortest path algorithm with movement penalties. Calculating K different paths for every O-D pair is computationally intensive, especially when the time-slots are small, whereby the number of O-D matrices and so the number of routes are considerably high. To enhance the model's performance, the K paths are not re-calculated at every simulation time-step, but are recomputed only at pre-specified intervals. In the Knoxville, TN, test network, consisting of 1870 nodes and 3304 links, the computational time using a Pentium-4 1.4 GHz processor with 2.0 GB memory, is reported to be 3 hours and 15 minutes when computing K=2-shortest paths.

DYNASMART-P recognizes four vehicle types, namely, passenger cars (PC), trucks, high occupancy vehicles (HOV), and buses for their effect on traffic conditions (such as link capacity, speed, density, and volume) and consequently, on path assignments. Note that PCs, trucks, and HOVs are specified as fractions of the overall vehicle fleet. In this case, the specified OD demand matrix should reflect vehicular trips. Alternatively, the user may specify a separate OD demand matrix to account for trucks in the network. In this case, there is no need to specify the fraction of trucks in the overall vehicle fleet.

4.4 Summary and Conclusion

Dynamic traffic assignment has been of interest to many researchers in the past three decades in both real-time traffic operations and in long-term planning. To capture the complex traffic flow dynamics, simulation-based models are extensively used. Most of the simulation packages in the market have been developed for real-time purposes. Some of the existing packages that can also be applied for planning purposes, and the methodologies used to compute user equilibria are reviewed in this chapter, and are contrasted with TRANSIMS, which is most well-suited for conducting dynamic traffic assignments for planning purposes.

TRANSIMS is the only microscopic large-scale transportation system simulation framework that has been designed for planning contexts. It tracks individual travelers and

estimates second-by-second movements during a 24 hour horizon. It might be argued that modeling the real-world of traffic in detail would be computationally intensive. However, TRANSIMS tackles this problem by applying parallel processing and the high-speed cellular automata approach in microsimulation. Therefore, it simulates every event in a reasonable amount of time. Also, the computational power of computers has doubled almost every two years in recent times, which makes using such a microscopic model possible.

Unlike other packages that require the O-D demand as input, TRANSIMS estimates and predicts the time-dependent demand. This makes the assignment process more reliable. Besides the time-dependency of the demand estimation and prediction process, TRANSIMS also considers the complex nature of travelers' behavior. It is the only package that treats individuals in a disaggregated fashion and is multi-modal. The traffic assignment in TRANSIMS supports route choice, mode choice, and departure time choice. It also does not need any user-class definition since it is disaggregated and investigates each person individually. TRANSIMS also represents the impact of a traffic signal timing plan more precisely than other planning packages.

Some of the existing problems in deploying dynamic traffic assignment in the planning context are addressed by TRANSIMS. Nevertheless, the current dynamic traffic assignment routines used within TRANSIMS have stability and computational problems, some of which have been addressed by researchers at Virginia Tech as discussed in this chapter. Yet, several improvements are still needed, and it is hoped that this study will serve to stimulate researchers to examine TRANSIMS more closely than has been forthcoming these far.

Chapter 5

An Algorithm to Compute Dynamic User Equilibria for Large-Scale Transportation Networks

5.1 Introduction

Traffic assignment is the allocation of transportation demand to the transportation supply infrastructure, based on factors that affect route choice. Traffic assignment models are classified into two major categories: static and dynamic. Static assignment models assume that traffic is in a steady-state, the link volumes are time-invariant, the time to traverse a link depends only on the number of vehicles on that link, and that the vehicle queues are stacked vertically and do not traverse to the upstream links in the network. In dynamic traffic assignment models, the demand is allowed to be time-varying so that the number of vehicles passing through a link and the corresponding link travel times become time-dependent.

There are two different approaches for determining steady-state flows. The system optimal (SO) approach, which minimizes the total system travel time over the planning horizon, and the user equilibrium (UE) approach, which seeks user path assignments that satisfy the Wardropian user equilibrium condition: No user can improve his/her experienced travel time by unilaterally switching routes. While the user equilibrium concept is descriptive in nature, the system optimal objective is normative. The dynamic user equilibrium problem (DUE) seeks time-dependent user path assignments that satisfy an extension of the Wardropian user equilibrium conditions in which no traveler can improve his/her experienced travel time by unilaterally switching routes (at any point in time) for a given departure time.

In contrast with the static case, the dynamic traffic assignment problem is still relatively unexplored. Most models in the literature are not accompanied by a solution algorithm, and whenever they are, the prescribed procedures are not suitable for large-scale networks. Among the suggested solution methodologies that claim to be applicable to large-scale networks, very few techniques have been actually tested in practice. We refer the reader to Peeta and Ziliaskopoulos (2001) for a comprehensive review of the literature in this area.

Static traffic assignment models use link performance functions to compute link travel times. A link performance function gives the link travel time as a function of

volume on the link. This function is a positive, increasing, and convex curve. One assumption implied by link performance functions, which is especially restrictive for congested networks, is that the travel time on a link is independent of the flows on the other links in the network. In reality, traffic queues spill backward and block junctions, resulting in queues forming on different streets. This phenomenon cannot be represented by models using link performance functions.

Nevertheless, in the development of dynamic traffic assignment models, some researchers have continued to use link performance functions to relate travel times to link volumes. These models are usually referred to analytical models. However, some other researchers have proposed to use traffic simulators rather than link performance functions to assess the performance of a particular loading demand. In this way, the complex dynamic interactions among vehicles in the network are captured. These models, which are called simulation-based models, ensure the first-in-first-out (FIFO) property of traffic flow and prevent the unintended holding back of vehicles in the network.

Traffic simulation models can be classified into three groups: macroscopic, microscopic, and mesoscopic. Macroscopic models move vehicles based on aggregate equations and are not very accurate. Microscopic models track the vehicles second-by-second and record details of individual drivers and vehicles and their interactions. Mesoscopic models relate speed and concentration on a section using macroscopic equations, but move vehicles individually or in packets. DYNASMART (*www.dynamictrafficassignment.org*), DynaMIT (*www.dynamictrafficassignment.org*), and CONTRAM (Leonard et al., 1978) are examples of mesoscopic models, while the Transportation Analysis and Simulation System, TRANSIMS, is a microscopic modele.

The approach adopted by TRANSIMS for computing dynamic user equilibria is different from that used in other packages in the way that the demand and supply are handled. The traffic assignment process in TRANSIMS is time-dependent, activity-based, and multi-modal, and the package supports route choice, mode choice, and departure time choice. Although the traffic assignment process in TRANSIMS incorporates many intricate details, it has some fundamental drawbacks. TRANSIMS routes one person at a time within each iteration loop, without considering how other travelers are being routed

within this loop. Also, the stopping criterion in TRANSIMS is based only on visualization, while the procedure might oscillate because of the foregoing routing process. This also entails a huge amount of computational effort to determine a near-user equilibrium, if at all one is found.

The objective of this chapter is to improve the dynamic traffic assignment procedure in TRANSIMS by developing a new heuristic approach that is based on a modification of the convex-simplex method. The method is applicable to large-scale transportation networks and tends to rapidly converge to a near-equilibrium solution. Although this technique can be incorporated in several computer packages such as DYNASMART and CONTRAM, we specifically develop a two-stage variant of this method for implementation within TRANSIMS. This variant adopts a combination of a link performance function for determining a good initial distribution along with a microsimulator for further refining this distribution. As an illustration, we apply the proposed approach to some large-scale networks and present comparative analyses. Our results exhibit that the proposed method achieves an improved distribution of travelers in fewer iterative loops than that required by the current version of TRANSIMS, while reducing computational effort by 20-33%. The proposed methodology is therefore both analytical and simulation-based, and it exploits the trade-off that exists between speed and precision in calculating link travel times using either link performance functions or a simulator by applying these techniques, stage-wise to take advantage of their individual features.

5.2 The User Equilibrium Model

Consider a network $G = (N, A)$ where N is set of nodes and A is set of links. Also, let the link performance function (delay function) on link l, given by $T_l(V_l)$, where V_l is the volume of traffic on link l, be a positive, increasing, convex, and continuous function. We assume that the link performance function depends only on the traffic volume on that link itself. We also assume that the FIFO condition holds true. The user equilibrium problem is to solve the following mathematical program.

$$\text{Minimize} \quad Z(V) = \sum_{l} \int_{0}^{V_l} T_l(w)\,dw \tag{1}$$

subject to

$$\sum_{p} f_p^{rs} = \sum_{a} q_{rsa}, \qquad \forall r, s,$$

$$f_p^{rs} \geq 0, \qquad \forall p, r, s,$$

$$f_p^{rs} = \sum_{a} f_{ap}^{rs}, \qquad \forall r, s,$$

$$V_l = \sum_{r} \sum_{s} \sum_{p} f_p^{rs} d_{l,p}^{rs}, \quad \forall l,$$

where the notation is explained in Table-5.1. The first two constraints are flow conservation constraints and non-negativity conditions, respectively, and the remaining restrictions are definitional equations. This formulation is similar to that in Sheffi (1985), but it is in the individual traveler context rather than aggregated flows. This convex mathematical program is equivalent to the equilibrium assignment problem in that its necessary and sufficient first-order conditions are identical to the equilibrium conditions. To see this, and to introduce some notation, observe that the Lagrangian of this problem with respect to the flow conservation constraints is given by:

Table-5.1: Network Notation

---

f_p^{rs} : flow on path p connecting origin r to destination s; $f^{rs} = (..., f_p^{rs}, ...)$; $f = (..., f^{rs}, ...)$

$d_{l,p}^{rs}$: indicator variable, $d_{l,p}^{rs} = \begin{cases} 1 & \text{if link } l \text{ is on path } p \text{ between } r \text{ and } s \\ 0 & \text{otherwise} \end{cases}$

$q_{rsa} = \begin{cases} 1 & \text{if traveler } a \text{ has origin } r \text{ and destination } s \\ 0 & \text{otherwise} \end{cases}$

$f_{ap}^{rs} = \begin{cases} 1 & \text{if traveler } a \text{ that has origin } r \text{ and destination } s, \text{ chooses path } p \\ 0 & \text{otherwise} \end{cases}$

V_l : flow on link l; $V = (..., V_l, ...)$

$T_l(w)$: travel time on link l, as a function of the flow w

c_p^{rs} : travel time on path p connecting origin r to destination s

$Z(.)$: the objective function, defined in (1).

$$L(f,u) = Z[V(f)] + \sum_r \sum_s u_{rs} (\sum_a q_{rsa} - \sum_p f_p^{rs})$$

where, $u_{rs}, \forall r, s$, denote the dual variables associated with the flow conservation constraints. To derive the first-order optimality conditions of program (1) note that (see Table-5.1):

$$\frac{\partial}{\partial f_k^{mn}} L(f,u) = \frac{\partial}{\partial f_k^{mn}} [Z(V(f)] + \frac{\partial}{\partial f_k^{mn}} \sum_r \sum_s u_{rs} (\sum_a q_{rsa} - \sum_p f_p^{rs}) = c_k^{rs} - u_{mn} \qquad (2)$$

because,

$$\frac{\partial Z(V(f))}{\partial f_k^{mn}} = \sum_{b \in A} \frac{\partial Z(V)}{\partial V_b} \frac{\partial V_b}{\partial f_k^{mn}} = \sum_{b \in A} T_b(V_b) d_{b,k}^{mn} = c_k^{mn}, \qquad (3)$$

where, $\frac{\partial Z(V)}{\partial V_b} = \frac{\partial}{\partial V_b} \sum_l \int_0^{V_l} T_l(w) dw = T_b(V_b) \equiv T_b$, and $\frac{\partial V_b}{\partial f_k^{mn}} = d_{b,k}^{mn}$. Furthermore,

$$\frac{\partial}{\partial f_k^{mn}} \sum_r \sum_s u_{rs} (\sum_a q_{rsa} - \sum_p f_p^{rs}) = -u_{mn}.$$

Therefore, using (2), the first-order Karush-Kuhn-Tucker (KKT) optimality conditions for (1) are (see Bazaraa et al., 1993):

$$c_p^{rs} - u_{rs} \geq 0, \qquad \forall r, s, p,$$
$$f_p^{rs}(c_p^{rs} - u_{rs}) = 0, \qquad \forall r, s, p,$$
$$\sum_p f_p^{rs} = \sum_a q_{rsa}, \qquad \forall r, s,$$
$$f_p^{rs} \geq 0, \qquad \forall r,s,p.$$

Observe that these conditions are equivalent to the user-equilibrium principle, where, for each (r, s), u_{rs} represents the least cost path and $f_p^{rs} > 0$ only if $c_p^{rs} = u_{rs}$, i.e., flow is permitted only on least cost paths.

Now, to motivate our approach, consider the application of the Frank-Wolfe approach to the minimization problem (1) (see Bazaraa et. al., 1993). In this method, we find a solution to the following linear program at each iteration n, which minimizes the first-order approximation at the current point V^n.

Minimize $Z(V^n) + (V - V^n)^t \nabla Z(V^n)$,

or equivalently, Minimize $(V)^t \nabla Z(V^n)$, subject to the constraints of (1). Taking the gradient of the objective function with respect to path flows as in (3), rather than link flows, this problem decomposes by OD-pairs, and for each such OD-pair (r, s), can be written as follows.

$$\text{Minimize} \quad \sum_p c_p^{rs} g_p^{rs} \tag{4}$$

subject to

$$\sum_p g_p^{rs} = \sum_a q_{rsa},$$
$$g_p^{rs} \geq 0, \qquad \forall p,$$

where g_p^{rs} is the auxiliary flow on path p between the OD-pair (r, s). This auxiliary problem is minimized by finding the path m having the smallest travel time among all paths connecting r and s and then assigning all the flow to this path. Therefore, $g_m^{rs} = 1$, and $g_p^{rs} = 0$, $\forall p \neq m$. Having solved this for each OD-pair (r, s), we can compute the

corresponding auxiliary link flows, denoted V_l^*, that result from the solution $(g_p^{rs})^*$, say, $\forall p, r, s,$ to (4) as follows:

$$V_l^* = \sum_r \sum_s \sum_p d_{l,p}^{rs} (g_p^{rs})^*, \qquad \forall l. \qquad (5)$$

This solution defines the direction $d^n = V^* - V^n$, which is a descent direction provided that $\nabla Z(V^n)^t (V^* - V^n) < 0,$ i.e., for at least one OD-pair (r, s), the objective value in (4) is lesser than that for the current flow solution. In this case, we perform a line-search from V^n in the direction d^n, update the link flows to V^{n+1}, and repeat. Convergence of this process is established in Bazaraa et al. (1993), for example. A step-wise statement of this algorithm is given below in terms of individual travelers for the sake of motivating our approach.

Step 1. Initialization: Let POP be the set of all travelers. Set the iteration number n =1. Load all travelers to the network based on free-flow travel times using an all-or-nothing assignment. Let V^1 denote the resulting link flow vector.

Step 2. For each traveler $a \in POP$, between each OD-pair (r, s), find the shortest path m for this traveler a ($g_{am}^{rs} = 1$ and $g_{ap}^{rs} = 0$, $\forall p \neq m$). Denote $(g_p^{rs})^* = \sum_a g_{ap}^{rs}, \forall p, r, s,$ as produced from this solution, and compute the resulting link flows V_l^*, $\forall l$, as in (5). If $\nabla Z(V^n)^t (V^* - V^n) = 0,$ then stop; V^n is an equilibrium solution. Else, proceed to Step 3.

Step 3. Line-Search: Compute the step length $l = l^*$ that solves:

$$\underset{0 \le l \le 1}{\text{minimize}} \quad \sum_l \int_0^{V_l^n + l(V_l^* - V_l^n)} T_l(w)\,dw. \qquad (6)$$

Step 4. Set $V_l^{n+1} = V_l^n + l^*(V_l^* - V_l^n)$, $\forall l$, update the corresponding link travel times using the link performance functions T_l, $\forall l$, increment n by one, and go to Step 2.

Note that in a convex-simplex variant of this procedure (see Bazaraa et. al., 1993), instead of basing the search direction on (4) for all OD pairs, we would examine each individual traveler a between each OD-pair (r, s) and examine if the corresponding shortest path for this traveler based on the current link flows is strictly better than the cost $\sum_p c_p^{rs} f_{ap}^{rs}$ for the existing flows f_{ap}^{rs}. If so, then a line-search can be performed as above based on modifying this user's flow distribution. If no user wishes to change flows at any iteration, then we will have found a user equilibrium and we can terminate. Convergence is again guaranteed for this procedure (see Bazaraa et al., 1993).

However, suppose that instead of performing a line-search, we simply let the step length l^*=1 in the foregoing convex-simplex routine, i.e., we perform an all-or-nothing assignment to find a new solution. Using such an all-or-nothing assignment policy cannot guarantee an equilibrium because it skips the line-search step that is necessary to induce convergence and, moreover, it assumes discrete flow distributions. Therefore, we need to add another stopping criterion to avoid oscillations, as we do below. Also, in the sequel, we use a weaker definition of a user equilibrium similar to Kaufman et al. (1998): Switching a path is allowed only if this switch decreases the actual resulting travel time. This weak-form of equilibrium prevents oscillations of the iterates. Figure-5.1 provides a flow-chart for this modified approach. A more effective algorithm based on this concept that avoids cycling is proposed in the next section.

Note that CONTRAM – CONtinuous TRaffic Assignment Model – (see Leonard et al., 1978) uses an iterative procedure to find dynamic user equilibria based on a related approach in which time is divided into small intervals, vehicles are grouped into packets, and each packet is treated as a single vehicle. At the initial iteration, each packet is assigned to its estimated shortest path. The list of links along the packets' routes and the time intervals in which the packets reach the downstream end of the links are stored. New delays are calculated based on the updated flows and the next packet is routed based on the updated link delays. In the subsequent iterations, each packet is assigned to new shortest routes, taking the initial flows as obtained from the end of the previous iteration. Also, before assigning any packet to its new route, the number of vehicles in the packet is

subtracted from the flow estimate on the links based on the packet's previous route in the relative time interval.

Leonard et al. (1978) state that sensitivity tests revealed poor convergence properties for this process. To remedy this, they proposed a 'fixed route' option, in which particular traffic demands are forced to choose certain specific routes, however unattractive they might be.

Our proposed method is similar to CONTRAM in that link delays are updated after routing each traveler (similar to Figure-5.1). Also, besides routing one traveler at a time rather than packets of travelers, we address the non-convergence issue of the algorithm differently and we also adopt a more effective update criterion. The next section explains this method in detail.

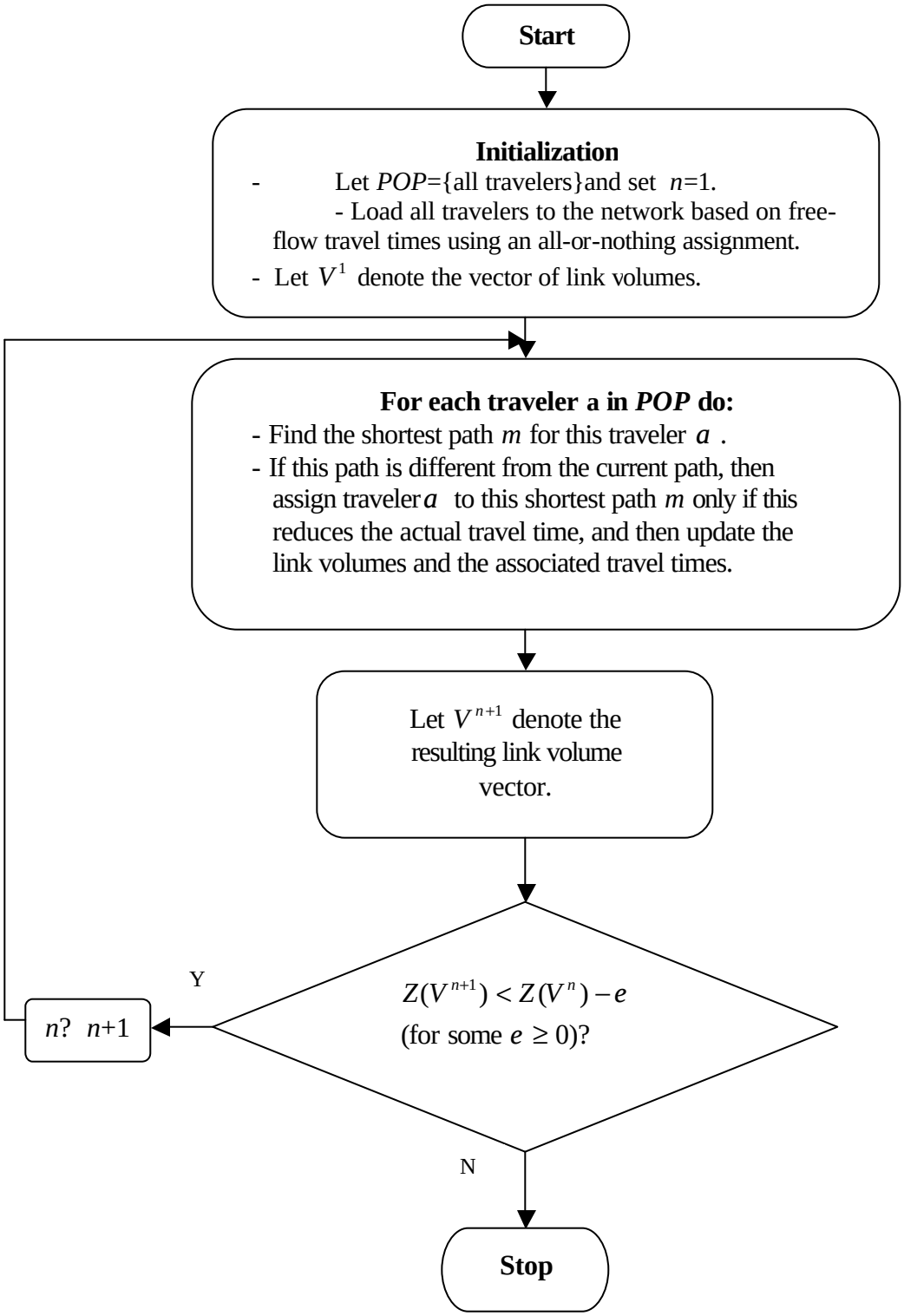


Figure-5.1: A Modified Convex-Simplex Approach for Finding User Equilibria.

5.3 The Propose Method: Dynamic Individual Routing

The method proposed in this section is a disaggregated procedure in which travelers are routed one at a time and the volumes are updated after routing each traveler. The iterative process can be described as follows.

1. Initialization: Let POP be the set of all travelers and let $TRAV$ be the set of travelers that should be rerouted, which is initialized as POP. Set the Experienced Travel Time at infinity for each traveler, where the *Experienced Travel Time*, ET_a, for each traveler a is the travel time based on the most recently updated link travel times. Set the smallest system-based objective value, Y^*, to be infinity and set $Count$ =0.
2. Select a traveler $a \in TRAV$ and do the following:
    a. Find the shortest path for a using the current link travel times.
    b. Tentatively update the volume on each involved link and find the actual new travel time of traveler a on the new path. Call this travel time the *Routed Travel Time* (RT_a). If $RT_a < ET_a$, use the new path for the traveler a, adopt the updated traffic volumes, and increment $Count$ by one; otherwise, use the old path.
    c. Delete traveler a from $TRAV$.
    d. If $TRAV \neq \emptyset$, repeat Step 2; otherwise, go to Step 3.
3. If $Count$=0, stop. (Note that alternatively, in practice, we could stop if $Count$ is less than some specified percentage of POP.) Otherwise, calculate the new Experienced Travel Time ET_a for each traveler. For each traveler a, if $RT_a < ET_a$, add a to $TRAV$.
4. If $\sum_a ET_a < Y^*$ and $TRAV \neq \emptyset$ go to Step 5; otherwise, stop.
5. Set $Y^* = \sum_a ET_a$ and $Count$ =0; go to Step 2.

At the beginning of this method, the network is empty and the link travel times are the free-flow travel times. In the first outer loop (steps 2-5), we load travelers onto the network one at a time while updating the link travel times. In subsequent iterations, we

attempt to re-route targeted travelers, for whom the Experienced Travel Time is greater than their Routed Travel Time. (This constitutes the set TRAV.) After re-routing any traveler during this overall process, the link travel times are updated using the link performance functions. However, before switching the path for any traveler a, we compare the travel time on the new path (the Routed Travel Time, RT_a) and the Experienced Travel Time (ET_a) that was computed in the previous iteration. Only if $RT_a < ET_a$, does the method then switch the path, which means that the weak-form of equilibrium is being used as explained in Section 2. Observe also that this method performs only an all-or-nothing assignment rather than perform a line-search. Hence, aside from the (weak) equilibrium-based stopping criterion, $Count$ =0, which means that no traveler changes his/her route, in order to avoid cycling, we have added another termination criterion: if the summation of the Expected Travel Times over all travelers does not strictly decrease, we stop.

Note that in this method, for any traveler a, whenever RT_a is set, this is the actual routing time for the then-current overall network loading. Hence, at the top of each loop, given some present loading, and the corresponding actual travel times (the Experienced Travel Times) for all the travelers, if $ET_a \le RT_a$, for any traveler a, then we do not consider a in this loop since the current travel time is at least as good as that at the previous loading when RT_a was set. Otherwise, if $ET_a > RT_a$, we reconsider revising the route for a, and change this route if the modified travel time can be made less than the current ET_a value. This motivates our designation of the set $TRAV$ at Step 3. In either case, as above, the RT_a value will be set at the actual travel time for a corresponding to the then-current loading solution.

Remark 1. For computational efficiency, it might be preferable to implement the above type of method by dispatching blocks of travelers at a time rather than only one traveler at a time. This would imitate the above disaggregated routing method, while using batches of travelers in lieu of routing each individual user one at a time. Therefore, this model can be used in computer packages such as CONTRAM and DYNASMART that treat packets of travelers as well as in TRANSIMS that treat individual travelers.

Next, consider the dynamic user equilibrium problem by incorporating an additional time dimension into this process. Assume that we know each traveler's departure time besides the origin and destination information. Since the demand is now allowed to be time varying, the number of vehicles passing through a link and the corresponding link travel times become time-dependent. Suppose further that the time horizon has been divided into small time-slots (t) such that the link volumes, and the consequent link travel times, remain fixed during each such time period. The link performance functions are assumed to be the same as before and depend only on the link volumes.

The related dynamic user equilibrium is computed in a manner similar to that in the previous algorithm, except that we now also have a time dimension. This algorithm is presented in Figure-5.2. Each traveler is identified by the departure time as well: a_t is traveler a with departure time t.

Lin and Lo (2000) demonstrated that the solution obtained by expanding dynamic traffic assignment models from the static user equilibrium models may not necessarily converge to the Wardropian user equilibrium condition. The reason is that these models divide the time horizon into time slots of duration t and find the equilibrium for each time slot. Therefore, when queues are formed in the junction area, the impact of delay in one time interval would spread to later time intervals. The dynamic traffic assignment procedure in our model is not to compute the equilibria for each time slot. We find the time-dependent shortest paths for each traveler based on the arrival time at each node and the link delay in the relevant time slot.

In this study, it is assumed that we know each traveler's origin, destination, and departure time. If the problem data is based on O-D flows during each time period, e.g. 15 minutes, then these O-D flows could be arbitrarily distributed among individual travelers to make the situation compatible with our algorithm. The algorithm would then attempt to find the volumes on each link in each time period such that no traveler can improve his/her experienced travel time by unilaterally switching routes for a given departure time.

Since this method is disaggregated and it is proposed in the individual context in which, one traveler or a packet of travelers are routed at a time, it is appropriate for the

microscopic or mesoscopic simulation-based modeling. Therefore, this method cannot be compared to the aggregated models that use the Frank-Wolfe algorithm (see Sheffi, 1985) and route all travelers simultaneously. Our proposed method might look much slower because of performing the inner iteration (Step 2), which is equal to the number of traveler packets on the network, while, this step is executed only once in the aggregated models. However, most of the dynamic user equilibrium models are mesoscopic or microscopic simulation-based in order to capture the complexity of time-dependent modeling. Our model is comparable with the mesoscopic or microscopic models in which packets of travelers are investigated.

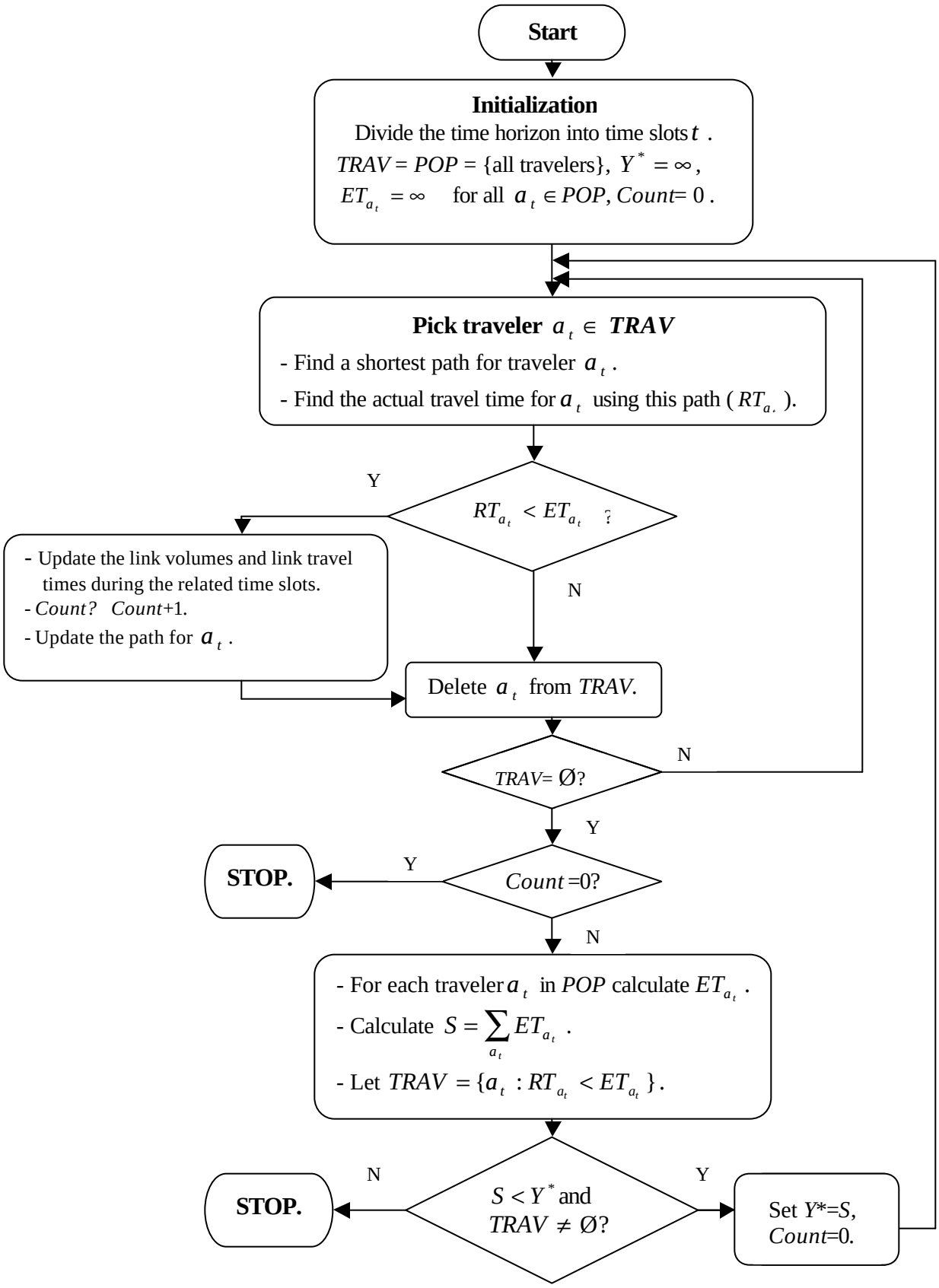


Figure-5.2: A Flow-chart for the Time-Dependent Dynamic Individual Routing Method.

5.4 Numerical Example

We present two simple static examples in this section to illustrate our methodology. (A larger practical dynamic user equilibrium example is considered in the following section.) The first example consists of a network having one origin, one destination, and three simple routes, with each route including only one link. The link performance functions are given as:

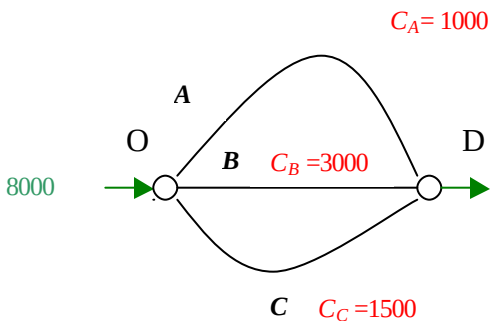
$$T_i = t_i(1 + .15(\frac{V_i}{C_i})^4), \qquad \text{for links } i = A, B, C,$$

where, T_i is the estimated travel time function on link i, V_i is the volume on link i, C_i is the capacity of link i, and t_i is the travel time on link i using a free-flow speed. The total volume on the network (required flow from O to D) is 8000; the capacities of links A, B, and C are 1000, 3000, and 1500, respectively, and all travelers commence at time zero (the problem is essentially static). Figure-5.3(a) depicts this network.

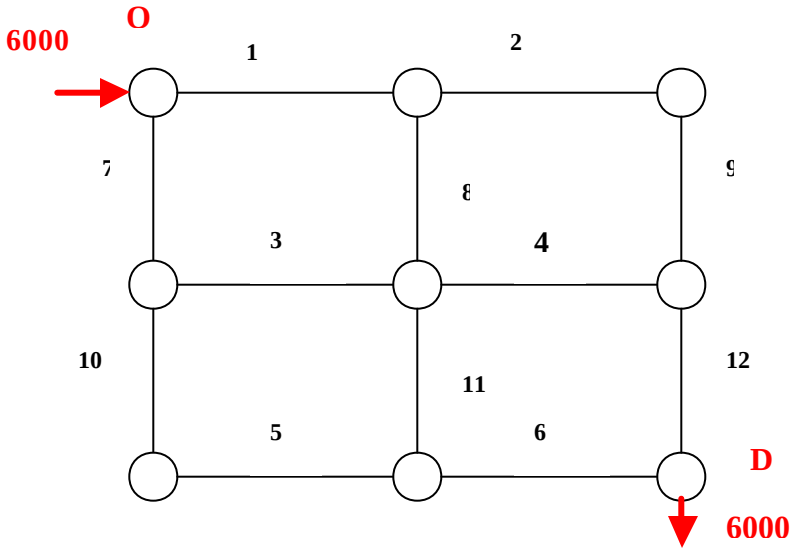
Tables 5.2 and 5.3 display the results of applying the algorithm of Figure-5.2 to the test network. The algorithm needs up to 8000 iterations in each inner loop, which is equal to the total number of travelers in the example. The first outer iteration performs an incremental loading, because the Experienced Travel Time is initially set to be a large number and thus, all travelers choose the current updated shortest path. Then the Experienced Travel Time is changed to 32.3 for all travelers. Since the Routed Travel Time is less than the Experienced Travel Time for all travelers except for the last one, *TRAV* would include 7999 travelers (*TRAV*={1, …, 7999}). Now, no traveler changes the existing path because the consequent actual current Routed Travel Time would then become greater than the corresponding Experienced Travel Time. Therefore, *TRAV* becomes empty and the algorithm terminates. The resulting solution is indeed a user equilibrium.

As stated earlier, our proposed method is not comparable with the aggregated models due to dealing with individual travelers. However, to validate the results of our model, the F-W algorithm in the aggregated context is performed and the results are presented in Table-54. Beside the inner loop (Step 2) in our model which is executed by the number of

packets of travelers, the outer iteration (steps 2-5) is executed only once and gives a more precise results compare to that in the F-W algorithm after 5 iterations.



(a) Three Route Example



(b) Grid Network Having Six Routes

Figure-5.3: Two Test Networks.

Table-5.2: Assignment Solution (First Iteration) for Example 1.

Outer Iteration (1)
POP = *TRAV*={1, 2, …, 8000}, *Count* = 0.

Iteration# (= person #)	Experie-nced Travel Time	Link A		Link B		Link C		TRAV	Routed Travel Time
		Flow	Time	Flow	Time	Flow	Time		
1	100000	1	15	0	20	0	21	{2,…,8000}	15
…	100000								
1221	100000	1221	20	0	20	0	21	{1222,…,8000}	20
1222	100000	1221	20	1	20	0	21	{1223,…,8000}	20
…	100000								
3501	100000	1221	20	2280	21	0	21	{3502,…,8000}	21
…	100000								
3559	100000	1280	21	2280	21	0	21	{3560,…,8000}	21
…	100000								
4627	100000	1280	21	2280	21	1126	22	{4628,…,8000}	21
…	100000								
5058	100000	1280	21	2711	22	1126	22	{5059,…,8000}	21
…	100000								
5106	100000	1328	22	2711	22	1126	22	{5107,…,8000}	22
…	100000								
5151	100000	1373	23	2711	22	1126	22	{5152,…,8000}	22
…	100000								
5440	100000	1373	23	3000	23	1126	22	{5441,…,8000}	22
…	100000								
5653	100000	1373	23	3000	23	1339	23	{5654,…,8000}	23
…	100000								
5796	100000	1373	23	3000	23	1482	24	{5797,…,8000}	23
…	100000								
6020	100000	1373	23	3224	24	1482	24	{6021,…,8000}	23
….	100000	1414	24	3224	24	1482	24	{6062,…,8000}	24
…	100000								
…	100000								
7892	100000	1658	32	4243	32	2050	32	{7893,…,8000}	32
…	100000								
8000	100000	1666	32.3	4269	32.3	2065	32.3	{}	32.3

Count = 7999, Experienced Travel Time for all travelers will be changed to 32.3.

Table-5.3: Assignment Solution (Second Iteration) for Example 1.

Outer Iteration (2)
POP = {1, 2, …, 8000}, *TRAV*={1, 2,…, 7999}, *Count* = 0.

Iteration	Experie-nced **Travel Time**	**Link A**		**Link B**		**Link C**		**TRAV**	**Routed Travel Time**
		Flow	**Time**	**Flow**	**Time**	**Flow**	**Time**		
0	32.3	1666	32.3	4272	32.3	2067	32.3	{2,…,7999}	≥32.3
…	32.3								
1221	32.3	1666	32.3	4272	32.3	2067	32.3	{1222,…,7999}	≥32.3
1222	32.3	1666	32.3	4272	32.3	2067	32.3	{1223,…,7999}	≥32.3
…	32.3								
3501	32.3	1666	32.3	4272	32.3	2067	32.3	{3502,…,7999}	≥32.3
…	32.3								
3559	32.3	1666	32.3	4272	32.3	2067	32.3	{3560,…,7999}	≥32.3
…	32.3								
7999	32.3	1666	32.3	4269	32.3	2065	32.3	{}	≥32.3

TRAV= Ø, *Count* = 0; Stop.

Table-5.4: F-W Algorithm for Example 1.

Iteration	**Link A**		**Link B**		**Link C**		**Objective Function**	**?**
	Flow	**Time**	**Flow**	**Time**	**Flow**	**Time**		
0	8000	9231	0	20	0	21	14864000	
1	0		8000		0			0.731
	2153	63.3	5847	63.3	0	21	220674	
2	0		0					0.258
	1598	29.7	4341				174807	
3	8000		0					0.011
	1666	32.3	4296				174697	
4	0		0					.004
	1659	32.3	4277				174687	
5	8000		0		0			.001
	1666	32.3	4272	32.3	2062	32.2	174686	

The second example, depicted in Figure-5.3(b), consists of a network having 9 nodes, 12 links, one origin, one destination, and six routes (1-2-9-12, 1-8-4-12, 7-3-4-12, 1-8-11-6, 7-3-11-6, and 7-10-5-6). The link performance functions are given as:

$$T_i = 5(1 + .15(\frac{V_i}{1000})^4), \qquad \text{for links} \quad i = 1,2,3,4,$$

$$T_i = 8(1 + .15(\frac{V_i}{3000})^4), \qquad \text{for links} \quad i = 5,6,$$

$$T_i = 7(1 + .15(\frac{V_i}{2000})^4), \qquad \text{for links} \quad i = 7,8,...,12.$$

The total volume on the network (required flow from O to D) is 6000. Unlike the previous example, observe that links are shared between different routes. Therefore, re-routing any traveler would affect travel times on all routes for which some link flows have changed. Table-5.5 presents the results obtained, where again the procedure converges in two iterations to a user equilibrium solution.

Table-5.5: Assignment Solution (First and Second Iteration) for Example 2.

Outer Iteration (1)
POP = *TRAV*={1, 2, …, 6000}, *Count* = 0.

a	Experienced Travel Time	Path chosen	Routed Travel Time
1	100000	1	24
2	100000	1	24
…	…	…	…
885	100000	3	24.04
…	…	…	…
1770	100000	2	25.56
…	…	…	…
2000	100000	1	26.71
…	…	…	…
2100	100000	3	26.93
2200	100000	5	27.77
…	…	…	…
6000	100000	3	47.30

Outer Iteration (2)
POP = {1, 2, …, 6000}, *TRAV*={1, 2,…, 5999}, *Count* = 0

a	Experienced Travel Time	Path chosen	Routed Travel Time
1	47.31	1	47.33
2	47.31	1	47.33
…	…	…	…
885	47.30	3	47.35
…	…	…	…
1770	47.30	2	47.33
…	…	…	…
2000	47.31	1	47.33
…	…	…	…
2100	47.30	3	47.35
2200	47.29	5	47.29
…	…	…	…
6000	46.30	3	47.35

$RT_a \geq ET_a$, $\forall a$, $TRAV = \emptyset$; stop.

5.5 An Implementation of the Proposed Algorithm within TRANSIMS

In this section, we present an implementation of the proposed approach within the context of TRANSIMS in order to design a framework for application to large-scale networks. TRANSIMS, Transportation Analysis and Simulation System, is a microscopic large-scale transportation system that has been developed at the Los Alamos National Laboratory for transportation planning purposes (see TRANSIMS 3.1 Documentation, 2003). It is an integrated system of travel forecasting models designed to give transportation planners accurate and complete information on traffic impacts, congestion, and pollution.

As stated before, this method is flexible and is applicable to analytical as well as simulation-based models. Among the simulation-based packages, this method is applicable to the microscopic models such as TRANSIMS; and mesoscopic models that treat packets of travelers to compute dynamic user equilibria. However, we implemented it within TRANSIMS due to availability and also, because improving dynamic traffic assignment procedure in TRANSIMS was one of our objectives.

The planning version of DynaMIT consists of a macroscopic supply simulator, a microscopic demand simulator, and an algorithm to perform the interaction between demand and supply. The solution is reached through an iterative process between the demand simulator and the supply simulator, and a new assignment matrix is computed at each iteration. The initial link travel times used in this method comes from historical link travel times. The planning version of DYNASMART uses free-flow link travel times to assign the O-D matrices onto the network. Then it iteratively simulates vehicles and finds the new travel times and assigns the travelers based on the new travel times using the successive averages (MSA). As explained before, CONTRAM routes packets of travelers one at a time and updates link travel times after each routing and the re-assigns travelers over several subsequent iterations, to reach a near-equilibrium.

TRANSIMS consists of six modules: Population Synthesizer, Activity Generator, Route Planner, Microsimulator, Feedback Controller, and Emissions Estimator. The Population Synthesizer estimates the number of synthetic households, along with their

demographics, the characteristics of each individual in these households, and the locations of these households on the network. The Activity Generator creates an activity list for each synthetic traveler. These activities include work, shopping, school, etc., and are based on matching the activity demographic characteristics of individuals with available survey data. In addition, activity times and activity locations are determined for each individual. The Route Planner computes the combined route and mode trip plans to accomplish the desired activities of each individual, such as work, shopping, etc. The Microsimulator uses the intermodal paths developed in the Route Planner module to perform a regional microsimulation of vehicle interactions. The Microsimulation continuously computes the operating status, including locations, speeds, and acceleration or deceleration of all vehicles throughout the simulation period. The output can provide a detailed, second-by-second history of every traveler in the system over a 24-hour period. The Feedback Controller manages the feedback of information among the Activity Generator, the Route Planner, and the Traffic Microsimulator modules of TRANSIMS. The Emissions Estimator uses the vehicle information generated in the Microsimulation module to forecast the nature, amount, and location of motor vehicle emissions.

The dynamic traffic assignment process in TRANSIMS is performed by the Feedback module. The simplest version to compute dynamic traffic assignment is a loop between the Route Planner and the Microsimulator modules. The Route Planner assigns transportation demand (travelers) into transportation supply (the network) for a 24-hour period using a time-dependent label-constraint shortest path algorithm developed by Barrett et al. (1998). Label is the travel mode such as car, walk, and bus. Each link has a delay associated with it. Delays are either constant (walking delay) or depend on time of day (street links and railways). The default value for each link delay is free-flow delay. Then, the Microsimulator simulates second-by-second travelers' movements and calculates the more realistic link delays over 15-minutes interval. The Route Planner, then is executed using the link delays calculated in the Microsimulator. This procedure continues until a near-equilibrium is reached.

The dynamic traffic assignment process in TRANSIMS has many capabilities (see Jeihani et al., 2004). Besides the time-dependency, it computes shortest paths on multi-

modal networks, which is unique among all computer packages. The dynamic traffic assignment model in TRANSIMS is microscopic simulation-based. It simulates second-by-second movements of individual travelers during the 24 hours. It uses the cellular automata approach and parallel processing to increase the computational speed. The traffic assignment in TRANSIMS supports route choice, mode choice, and departure time choice. TRANSIMS estimates and predicts the time-dependent transportation demand, which is unique among other computer packages. Furthermore, TRANSIMS is activity-based and estimates activities of each individual traveler rather than generic O-D matrices. Nevertheless, current dynamic traffic assignment in TRANSIMS has some drawbacks as explained next.

The Route Planner uses free-flow speeds on each link to estimate the travel times, which is not accurate because there exist other vehicles on the link, and so, the speed is not simply equal to the free-flow speed. Therefore, some paths used might not be the shortest paths due to congestion. The Microsimulator produces the new travel times based on accurate vehicle speeds, and feeds these back to the Route Planner, which in turn determines the new routes as shortest paths for selected travelers. In other words, TRANSIMS routes one person at a time but does not immediately update the link travel times. Therefore, each traveler is routed regardless of the other travelers on the network. Changing the procedure to update the link travel times after routing each person is an improvement of the current dynamic traffic assignment in TRANSIMS. Also, the current stopping criterion in TRANSIMS is based only on visualization and the existing procedure might oscillate. In addition, the current traffic assignment process expends a huge amount of time by iterating frequently between the Route Planner and the Microsimulator. For example in the Portland study, 21 iterations between the Route Planner and the Microsimulator were performed, which took 33:29 hours using three 500-MHZ CPUs (parallel processing). It is evident that a faster mechanism is needed to assign the traffic.

Applying our proposed method to TRANSIMS alleviates all of the above problems. The method we implemented is slightly different from that described in Figure-5.2 as explained next. As stated earlier, TRANSIMS has a Microsimulator that simulates every

move of the travelers to compute the actual link travel times rather than employ link performance functions. Using a traffic simulator, the complex dynamic interactions among vehicles in the network can be captured. The use of simulation ensures the first-in-first-out (FIFO) property of traffic flow and prevents forming vertical queues on links.

However, using the Microsimulator to compute link travel times for use in a dynamic user equilibrium algorithm is very time consuming. Therefore, there is a tradeoff between using a Microsimulator versus using link performance functions. We addressed this problem by implementing the concept of our approach over two stages. The first stage, which corresponds to the first outer iteration in the method of Figure-5.2, is called the *Incremental Individual Loader*. This procedure is a revised version of the current Route Planner that routes one traveler at a time and updates the link volumes and the link travel times using the BPR link performance function (see Oppenheim, 1995) after each routing. Note that any link performance function which is convex, increasing, continuous, and positive, can be used in this model and the BPR link performance function is only a representative of such kind of link performance functions. Since the arrival time of each person at each node is computed by the time-dependent shortest path algorithm, the volume on each link along with the associated travel time during each specific time-slot can be updated. To route the next person, the selected shortest path is calculated based on the updated travel times. In this manner, we avoid links from getting congested by virtue of falsely appearing to be faster because of the lack of updating travel times based on the occupancy of other travelers. Figure 5.4 presents the Incremental Individual Loader algorithm. This algorithm distributes travelers in a better manner that tends to obviate an excessive number of iterations between the Route Planner and the Microsimulator. Since the link travel times are functions of the volume to capacity ratios of the links (V/C), and the capacities of the links are of the order of thousands of vehicles per hour, the elasticity of the link travel times to link volumes is low. As evident from the examples in Section 5.4 (see Tables 5.3 and 5.5), after performing the Incremental Individual Loader, all paths having the same origin and destination have almost the same travel time and a near-equilibrium is obtained in this step. However, the link travel times might be very different from the realistic ones due to the assumed independence of travel time on a link from

flows on other links in the link performance function, and therefore, the solution in this step might not actually be a near-equilibrium.

Accordingly, the second stage of the algorithm calculates the link travel times as well as that of individual travelers using the Microsimulator to find more realistic travel times than those calculated by the link performance function. It also tries to find a near-equilibrium by alternating between the current Route Planner and the Microsimulator by iteratively redistributing a subset of travelers that have the highest ratio of the actual Experienced Travel Time to the Routed Travel Time that was calculated using the link performance function. Figure-5.5 presents a flow-chart for the overall algorithm that is implemented within TRANSIMS.

Note that in our method, the link performance function predicts the link travel times that increase monotonically with number of vehicles (time-dependent) on the link, rather than flows. Therefore, the problem of associating flow with two possible travel times is addressed. Also, the first stage of the algorithm is executed only once to distribute travelers regard to other travelers' occupancy on the network. Consequently, the travel times calculated by the link performance functions are used only once. Then, the second stage, which is similar to the current DTA process in TRANSIMS, is simply an iteration between the current Route Planner and the Microsimulator to redistribute the travelers that are miss-distributed on the network due to un-preciseness of the link performance functions in calculating link delays, specially in the congested networks. Therefore, the more realistic link travel times are computed by the Microsimulator and are used in the later iterations. The main difference between the second stage of our method and the current DTA process in TRANSIMS is that we select travelers that experiencing highest ratio of their travel time computed in the Microsimulator and the travel time calculated in the Route Planner. Furthermore, we use a descent-based stopping criteria rather than using visualization.

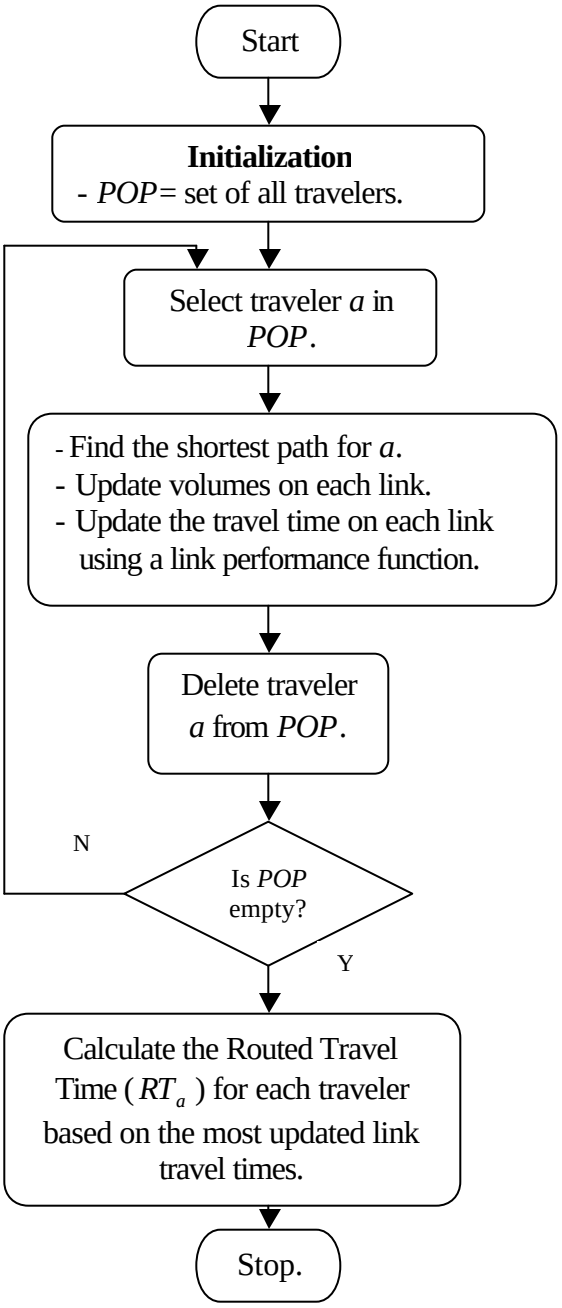


Figure-5.4: A Flow-chart for the Incremental Individual Loader.

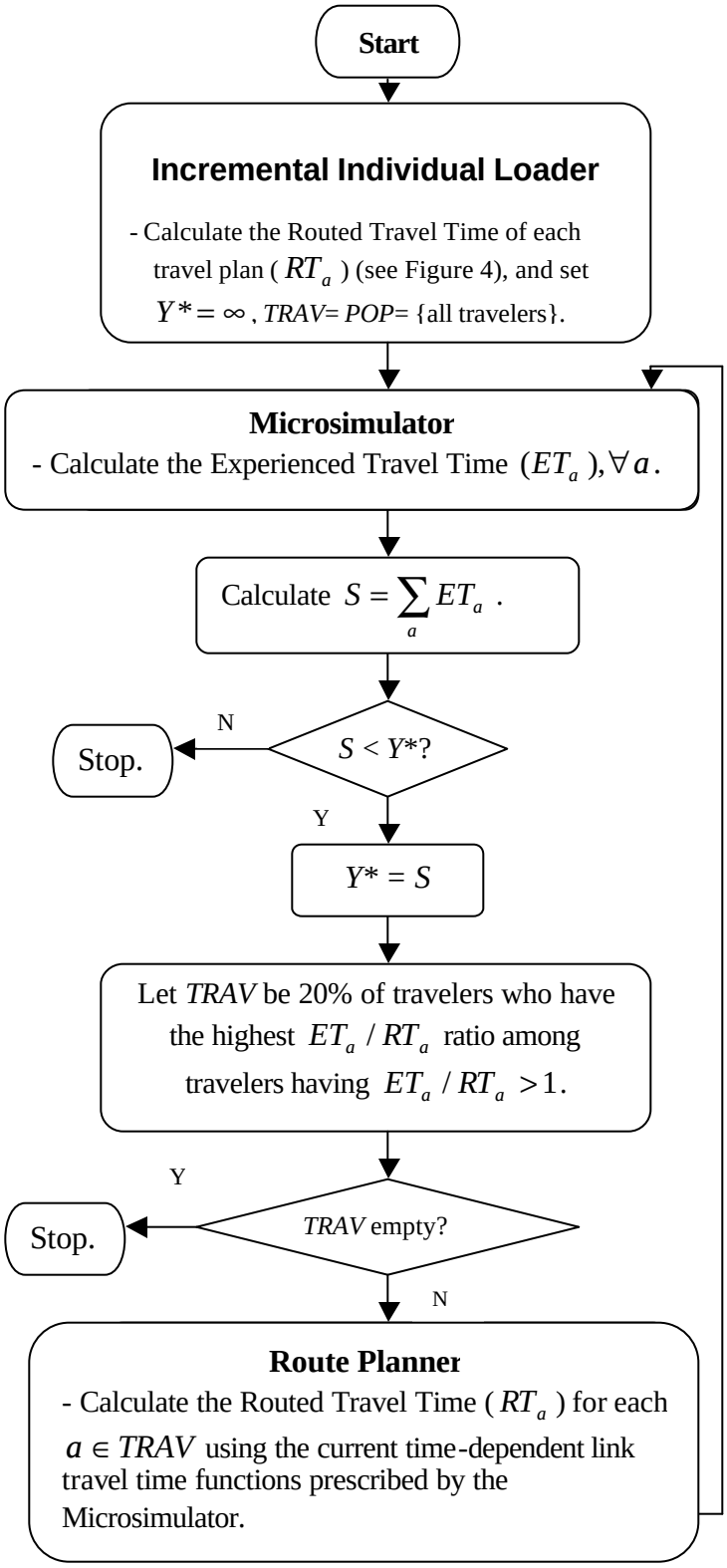


Figure-5.5: A Flow-chart for the Implemented Algorithm within TRANSIMS.

5.6 Experimental Analysis

The current equilibrating process of TRANSIMS, and also our proposed methodology, were applied to a large-scale network called Bignet; and a medium-scale network, Blacksburg. The Bignet network includes 3853 nodes and 7441 links with 1748 kilometers of roadway. This network represents about one-tenth the size of Portland city, Oregon. This network is tested under two different transportation demand patterns. First having 134,500 itinerant travelers in the network and second having 180,000 itinerant travelers. The Blacksburg network includes 1387 links and 1135 nodes.

5.6.1 The Bignet Network

The first demand pattern includes 134500 itinerant travelers who travel on this network. We tested this network on both current TRANSIMS and our proposed model. In the current TRANSIMS model, we randomly selected travelers to be re-routed. Sixteen iterations were performed between the Route Planner and the Microsimulator to re-distribute travelers to obtain improved solutions. These percentages are 30%, 30%, 20%, 20%, 20%, 20%, 20%, 15%, 15%, 15%, 10%, 10%, 10%, 10%, 10%, and 5%, respectively. The snapshots of the network for the first iteration and the last iteration are compared in Figure-5.6. Most of the travelers use the major roads during the first iteration, while some of them switch to minor roads in the last iteration. Therefore, the travelers are distributed in a better manner after some iterations. However, it is not clear if the last iteration yields a near-user equilibrium solution. In other words, it is not clear when we should stop the iterations between the Route Planner and the MicroSimulator. Also, there is no guarantee that better results would be obtained by running more iterations. The total vehicle hours traveled in the network are 5838 and 3750 in the first and last iterations, respectively. If we examine the distribution of vehicles in the 11th iteration, as presented in Figure-5.7, the travelers appear to be distributed better than after the 16th iteration. Therefore, the iterative procedure is probably cycling and it might be better to stop after the 11th iteration.

The proposed algorithm (modified as in Figure-5.5) was then applied to this Bignet network. The distribution of the travelers obtained using the initial Incremental Individual Loader (Revised Route Planner) was significantly better than that obtained using the current TRANSIMS' first iteration. The proposed algorithm terminated after three iterations between the Route Planner and the Microsimulator. Figure-5.8 presents the distribution of travelers in the first iteration (using the Incremental Individual Loader) and after the 3rd iteration. Travelers appear to be distributed better after the 3rd iteration of the proposed method as compared with the results after running the existing version of TRANSIMS for 16 iterations. The total vehicle hours traveled in the network are 3628 and 3541 after the first and last (third) iterations (as compared with 3750 after 16 iterations of TRANSIMS). Meanwhile, the CPU time consumed by our proposed method is considerably lesser than that for the current TRANSIMS method: 30 hours as compared to 148 hours, using four 1.6-GHZ CPUs. (Note that the parallel processing is utilized only by the Microsimulator.) We also arbitrarily selected eight travelers to observe their travel times over different iterations in both TRANSIMS and in our proposed method. Five of these travelers have very high travel times, while three of them have medium-to-low travel times. The results are presented in figures 5.9 and 5.10. All the travelers appear to reach a stable travel time using our method, while their travel times oscillate when applying TRANSIMS. Furthermore, the travelers have lower initial travel times using our proposed dynamic individual routing method in comparison with TRANSIMS. It is also instruction to examine the different routes and distances traveled by selected travelers in the three iterations of the dynamic individual routing method. For example, to initialize one particular traveler uses a route having 56 links and 7210m in length with a travel time of 2202 seconds. This traveler then switches to a longer route having 66 links and 14410m in length in the first iteration with a travel time of 1766 seconds. In the second iteration, the traveler chooses a route having 57 links, 8110m in length, and with a travel time of 1748 seconds, while in the last iteration, the route used has 82 links, with a total length of 16660m, and a travel time of 1720 seconds.

The current Incremental Individual Loader is run on a single processor because we need to update the travel times after each run. However, subsequent iterations could be run under parallel processing. Using a single processor for the Incremental Individual

Loader and parallel processing for subsequent runs of the Route Planner in both the current version of TRANSIMS and in the proposed method, the CPU times were reduced to 64 and 21 hours, respectively. In the case of very large networks in which running the Individual Incremental Loader might be prohibitive, this method could be changed to update travel times every 3000 re-routings or so, rather than after each re-routing, as explained before. We can process these batches of 3000 travelers in parallel, updating the travel times between each such run.

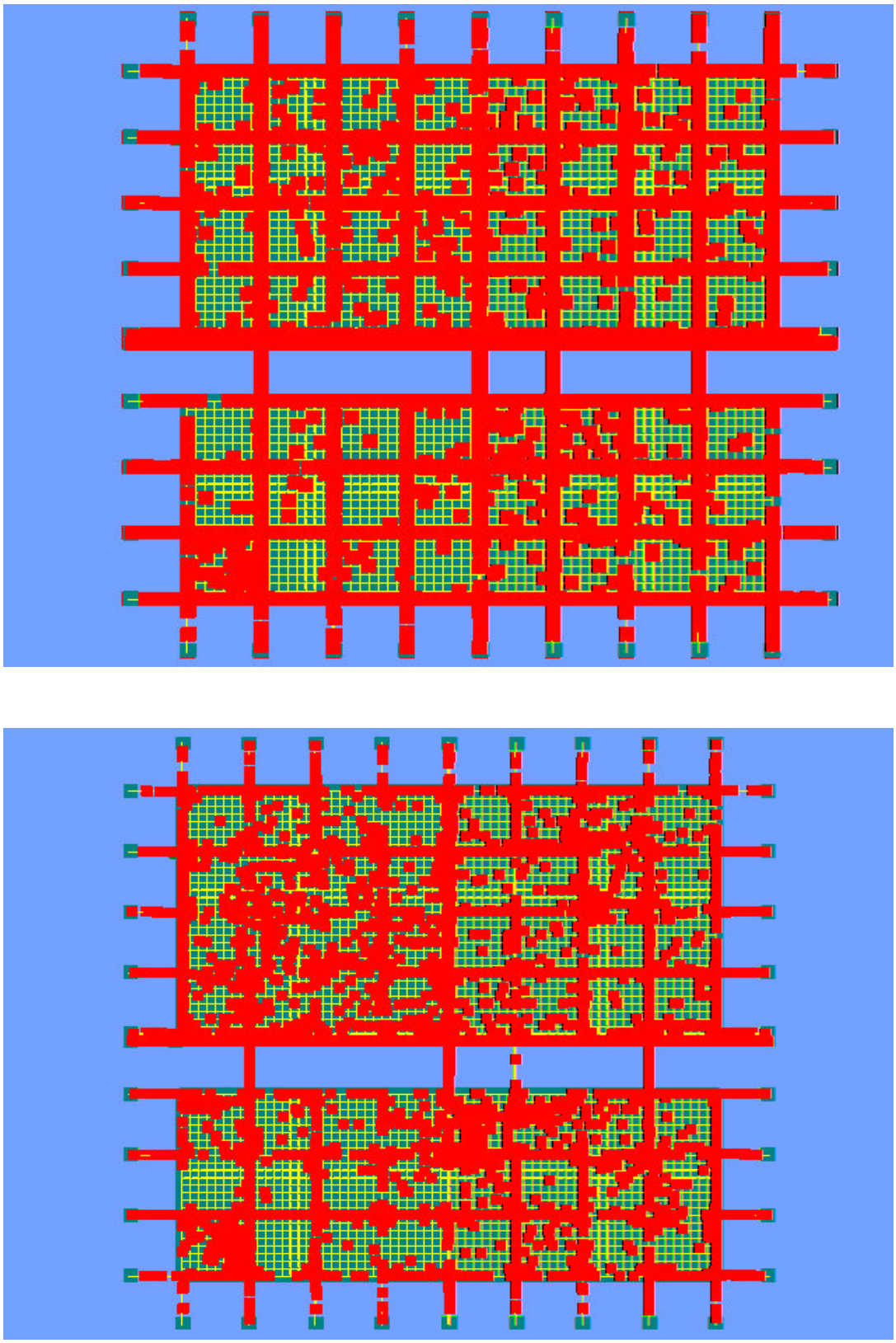


Figure -5.6: The Distribution of Travelers on the Bignet Network for the First and Last (16th) Iteration of TRANSIMS.

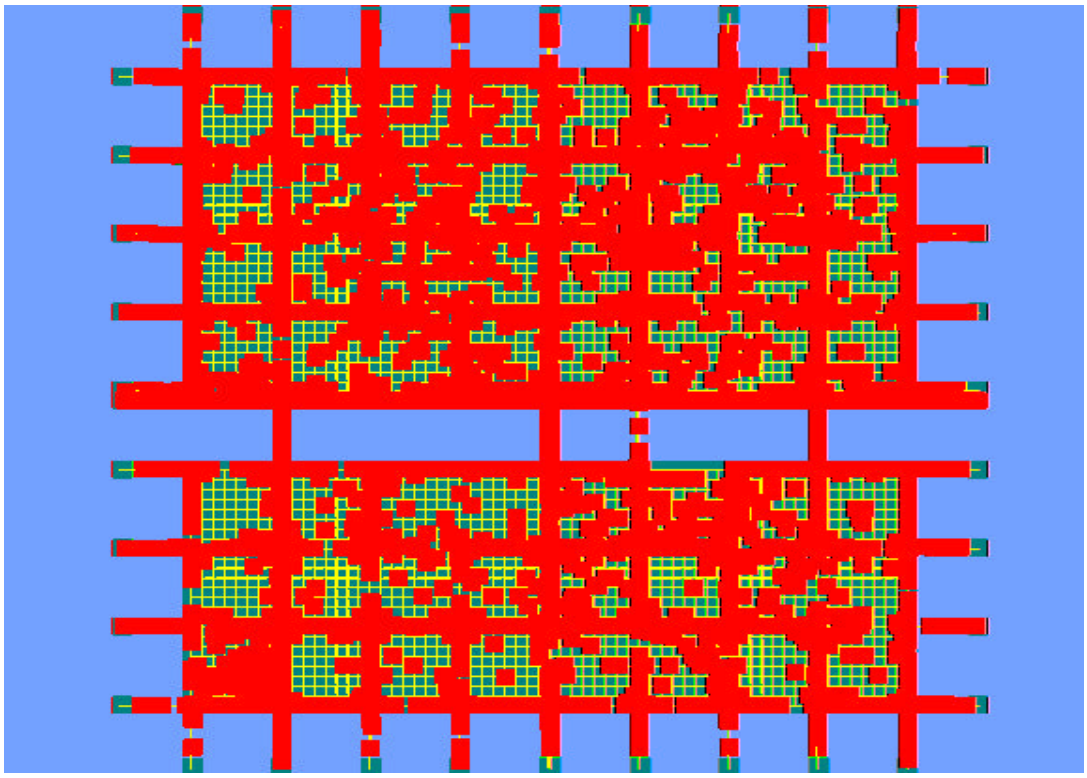


Figure -5.7: The Distribution of Travelers on the Bignet Network in the 11th Iteration of TRANSIMS.

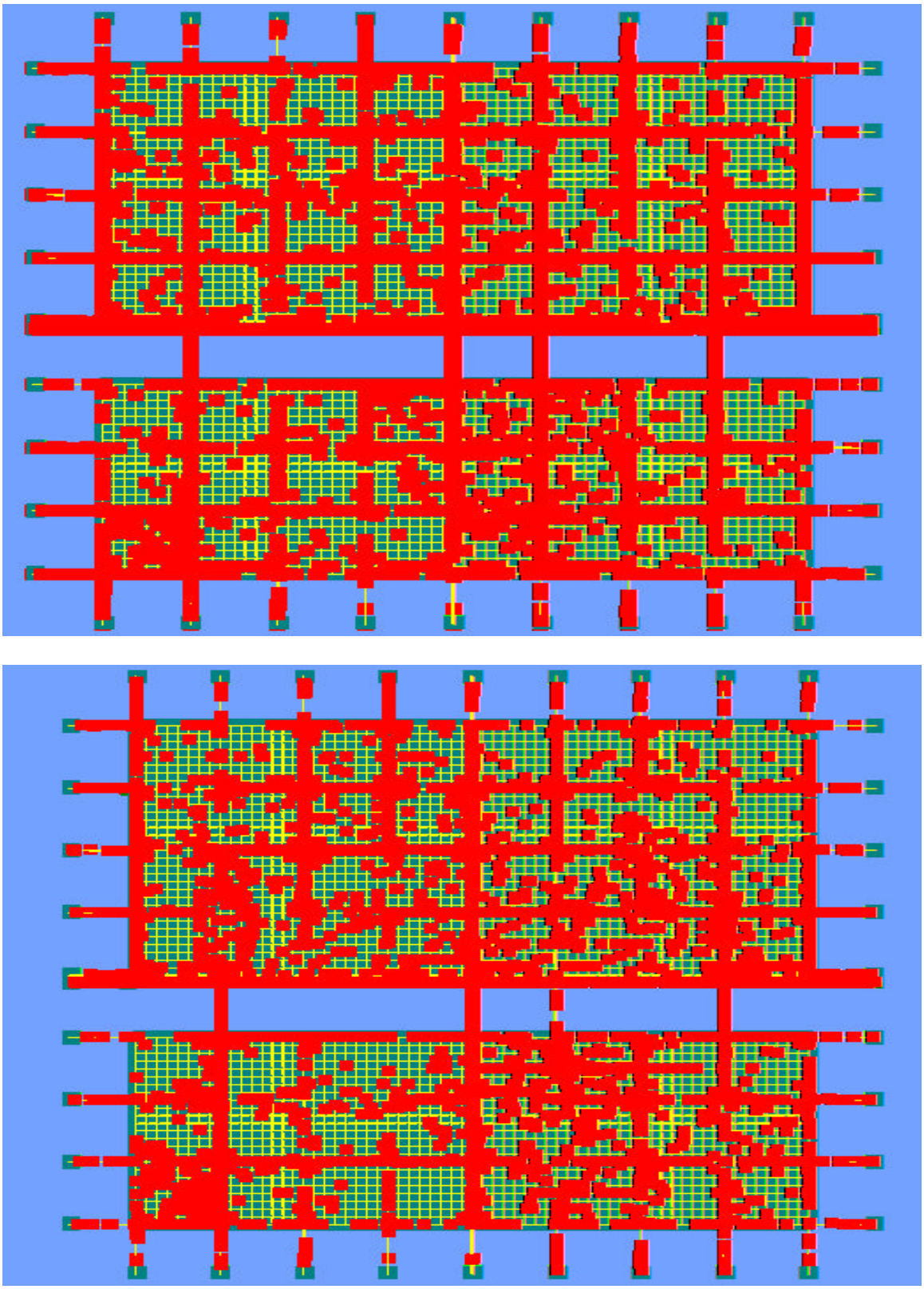
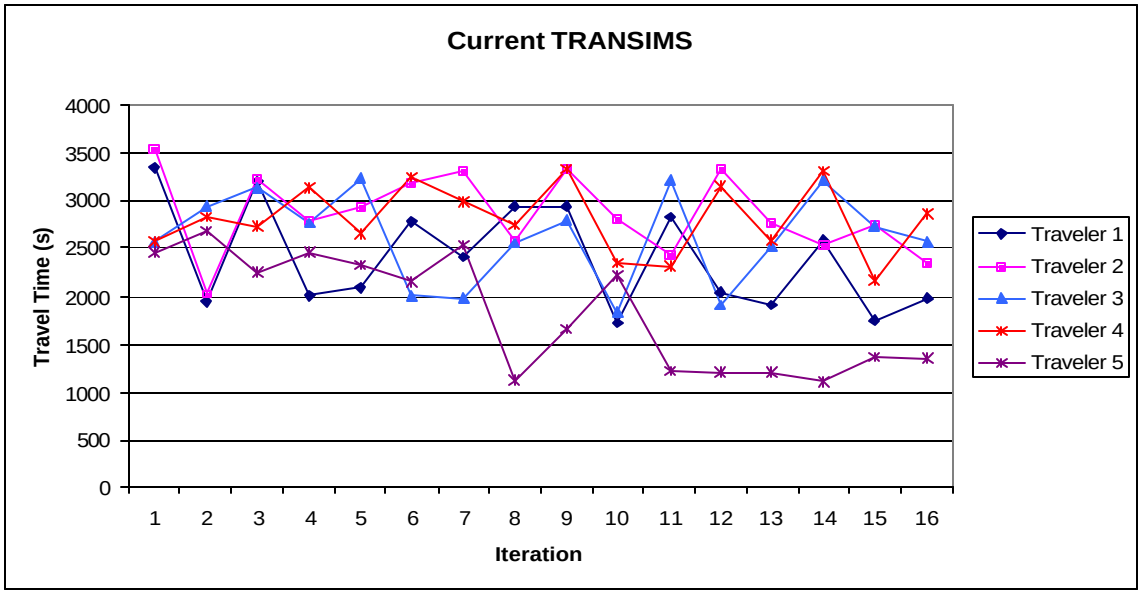


Figure-5.8: The Distribution of Travelers on the Bignet Network Using the Incremental Individual Loader and after Three Iterations of the Proposed Algorithm.



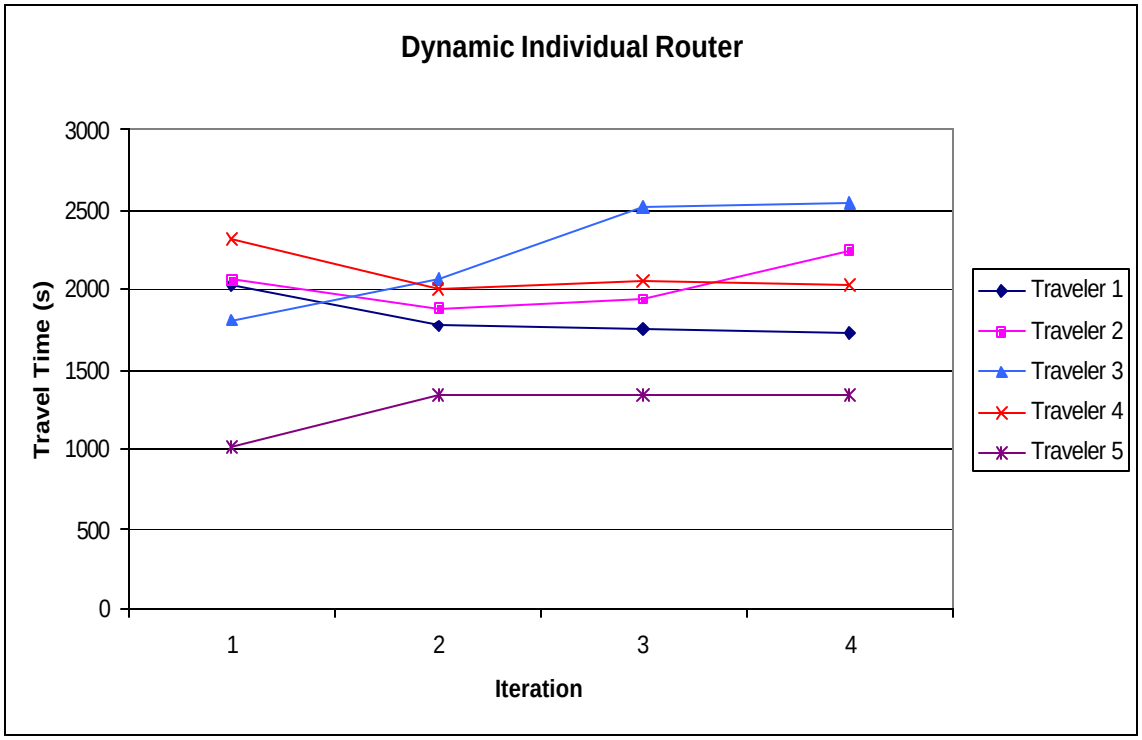
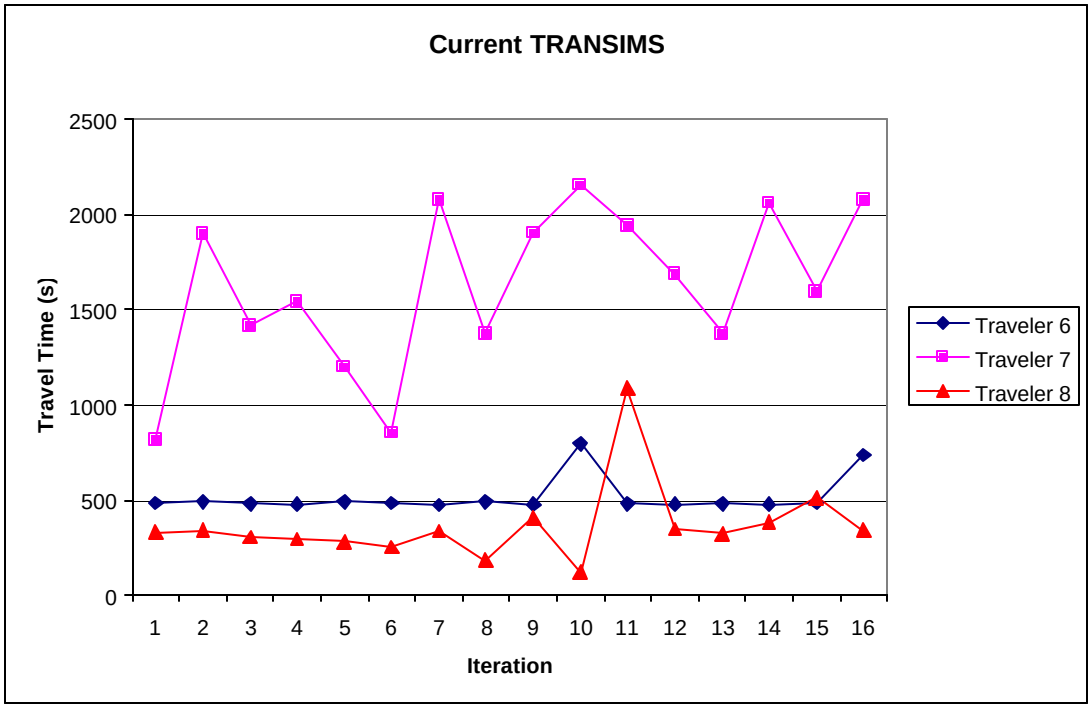


Figure-5.9: Travel Times versus Iterations for Five Travelers Having High Travel Durations Using TRANSIMS and Our Proposed Method.



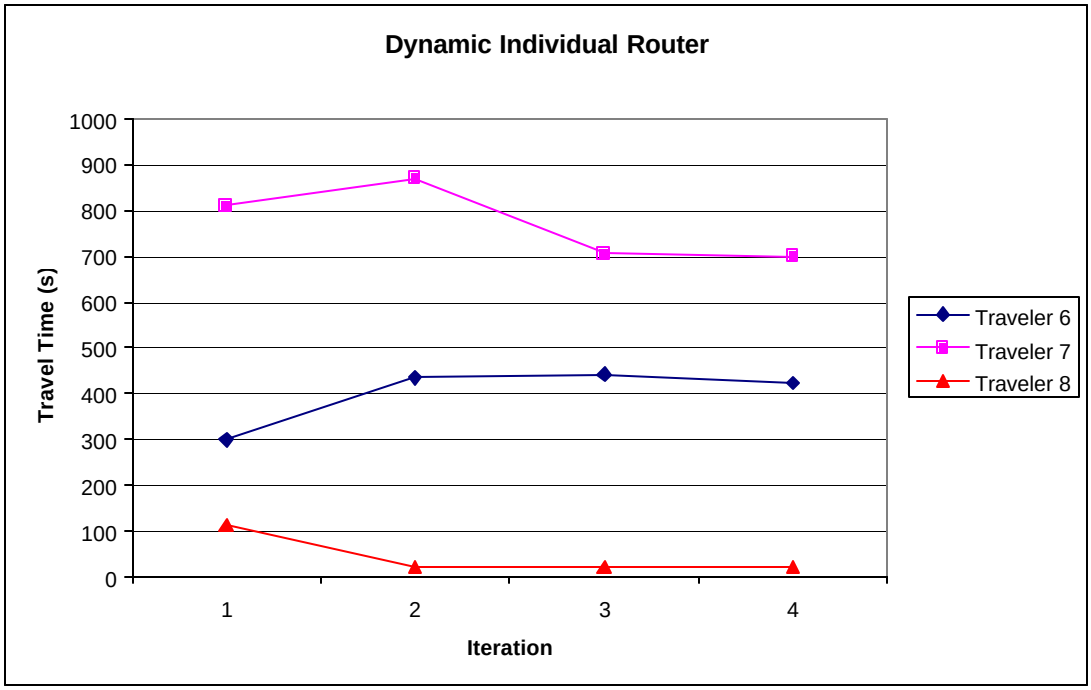


Figure-5.10: Travel Times versus Iterations for Three Travelers Having Medium-To-Low Travel Durations Using TRANSIMS and Our Proposed Method.

5.6.2 The Bignet Network Under a More Congested Regime

We increased number of travelers to be 180,000 and we tested this network on the current TRANSIMS and on our proposed method. In the current TRANSIMS, twenty iterations were performed between the Route Planner and the Microsimulator. These percentages of travelers that are re-routed in each iteration are 30%, 30%, 20%, 20%, 20%, 20%, 20%, 15%, 15%, 15%, 10%, 10%, 10%, 10%, 10%, 10%, 5%, 5%, 5%, and 5%, respectively. The snapshots of the network for the first iteration and the last iteration are compared in Figure-5.11. The results seem to be consistent with the first demand pattern. The total vehicle hours traveled in the network are 6901 and 4073 in the first and the last iterations, respectively.

The distribution of travelers in our proposed method is compared in Figure-5.12. The algorithm terminated after four iterations between the Route Planner and the Microsimulator. Similar to the first demand pattern, the distribution of travelers is much better compared to that in the current TRANSIMS. The total vehicle hours traveled in the network in the first iteration (Incremental Individual Loader) and the fourth iteration are 4287 and 4073. The CPU time consumed by our proposed method is considerably lesser than that for the current TRANSIMS method: 60 hours as compared to 300 hours.

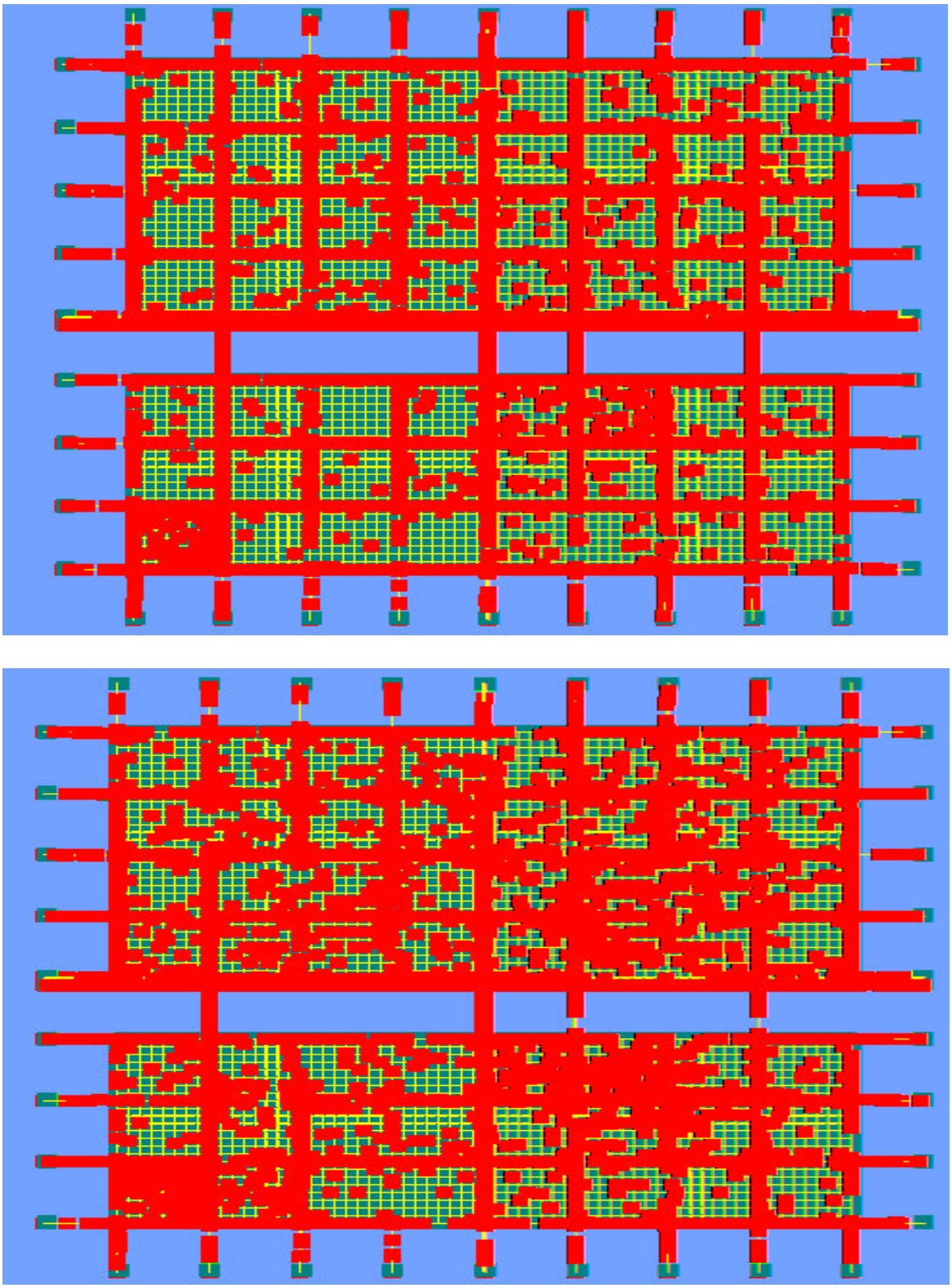


Figure -5.11: The Distribution of Travelers on the Bignet Network for the First and Last (20th) Iteration of TRANSIMS.

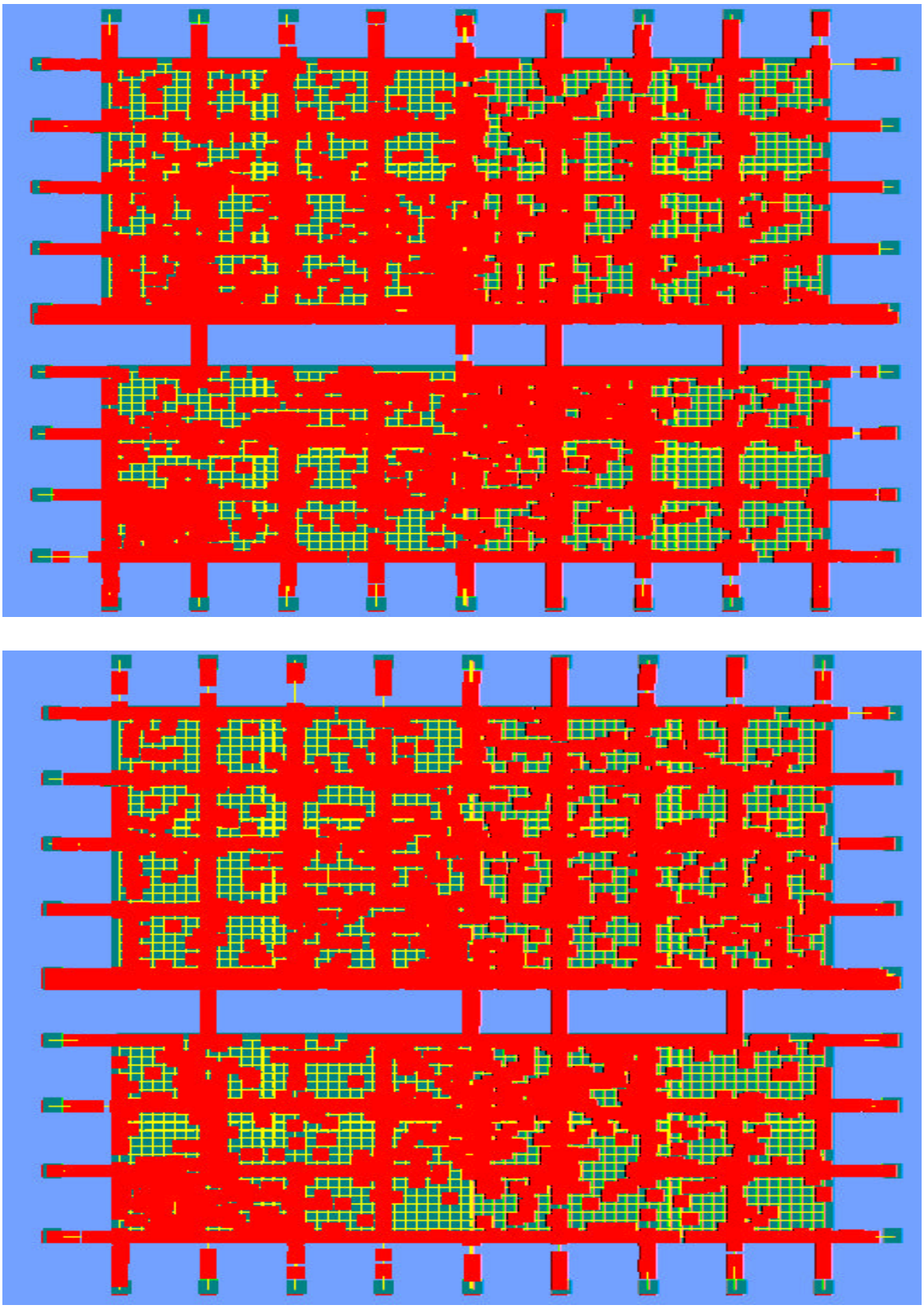


Figure-5.12: The Distribution of Travelers on the Bignet Network Using the Incremental Individual Loader and after Four Iterations of the Proposed Algorithm.

5.6.3 The Blacksburg Network

The Blacksburg network is a medium-scale network having 1387 links and 1135 nodes. We obtained the population from the census data and increased it 20% to make it more congested representing future travel demand. We also collected an activity survey from 300 travelers and we performed the Population Synthesizer and the Activity generator to obtain activities on this network. This network includes 57334 households and 119169 travelers. Twelve iterations were performed between the Route Planner and the Microsimulator to re-distribute travelers to obtain improved solutions in the current TRANSIMS model. These percentages are 20%, 20%, 20%, 20%, 10%, 10%, 10%, 10%, 10%, 5%, 5%, 5%, and 5%, respectively. The snapshots of the network for the first iteration and the last iteration are presented in Figure-5.13. The total vehicle hours traveled in the network are 364 and 331.3 in the first and the last iterations, respectively.

We also applied this network to our proposed method, which is presented in Figure-5.13. The algorithm terminated after two iterations between the Route Planner and the Microsimulator. Similar to the Bignet network, the distribution of travelers is much better compared to that in the current TRANSIMS. The total vehicle hours traveled in the network in the first iteration (Incremental Individual Loader) and the second iteration are 373.41 and 329.98.

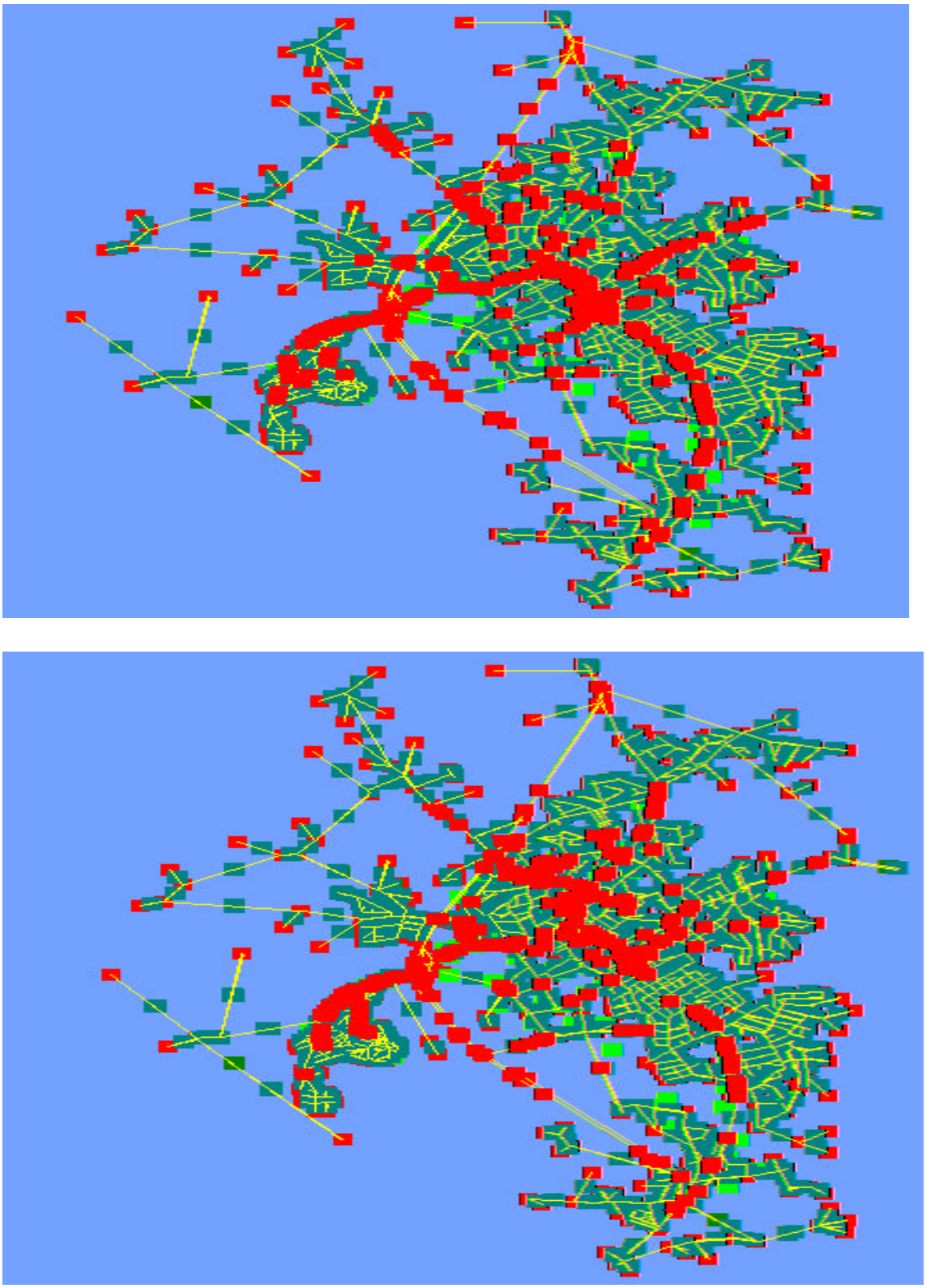


Figure -5.13: The Distribution of Travelers on the Blacksburg Network for the First and Last (12th) Iteration of TRANSIMS.

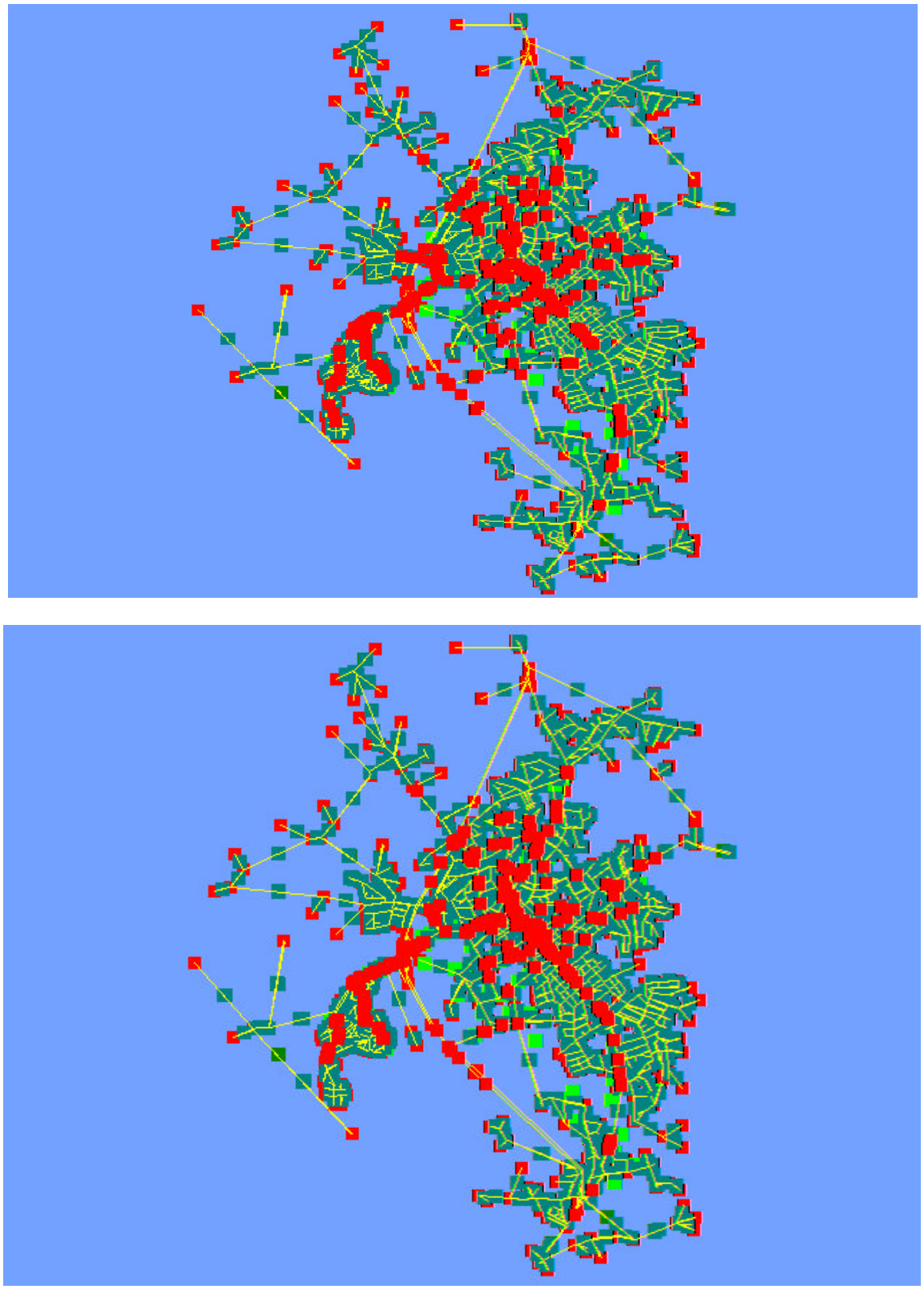


Figure -5.14: The Distribution of Travelers on the Blacksburg Network Using the Incremental Individual Loader and after Two Iterations of the Proposed Algorithm.

Evidently, examining the network snapshots provided by the Microsimulation does not provide a suitable measurement to evaluate the convergence to a near-equilibrium solution for computed by the two methods. Therefore, we compute the following measure to compare the results produced by the two models. Both models yield the Experienced Travel Time (ET_a) for each traveler a, as well as the time-dependent link travel times as outputs. Hence, for each traveler a, we determine the shortest path duration, given that the other travelers retain their routes, that is, using the current time-dependent link travel times as given. Call this travel time PT_a. Let $D_a = ET_a - PT_a$. As a measure of non-equilibrium, we examine different characteristics of the D_a-values (mean, standard deviation (stdv), and maximum) to compare current TRANSIMS and our proposed method. Table-5.6 presents this measurement for all the above scenarios.

Table-5.6: Evaluation of the Results for TRANSIMS Versus the Proposed Method for the Three Network Scenarios.

		$Mean(D_a)$ **(Minutes)**	$Stdv(D_a)$	$Max\{D_a\}$ **(Minutes)**	**# of Iterations**
Bignet (Scenario 1)	**TRANSIMS**	60.3	18231.4	809	16
	TRANSIMS	62.8	16531.4	820	11
	Proposed Method	37.8	10890.5	770	3
Congested Bignet (Scenario 2)	**TRANSIMS**	56.8	17467.9	796	20
	Proposed Method	36.7	10251	769	4
Blacksburg (Scenario 3)	**TRANSIMS**	21.8	9867.9	65	12
	Proposed Method	18.6	6548.4	49	2

5.7 Different Approaches to Compute Dynamic User Equilibria in TRANSIMS Using Post-Processing Actions

In this section, we test different strategies to find a dynamic user equilibrium given a link performance function and time-dependent volumes for the Example 1 (Section 4.4). These strategies that are based on post-processing actions rather than changing the current Route Planner could be compared to the dynamic individual routing method. In TRANSIMS, the Microsimulator module is used to find travel times on each link and an iterative process between the Microsimulator and the Route Planner modules is conducted to find a dynamic user equilibrium. We propose using the Route Planner to implement the traffic assignment process based on a suitable link performance function in order to avoid running the Microsimulator to obtain the travel times. We classify our strategies into three categories: aggregated routing, disaggregated routing, and incremental loading. The disaggregated routing is the dynamic individual routing procedure that was introduced in Section 5.3 and 5.4.

5.7.1 Aggregated Routing

In this method, we route all the travelers and then count the volumes on each link to find travel times using a link performance function. The selection of people to be rerouted can be done in different ways, as expounded below.

5.7.1.1 Fixed Random Selection

In this method we randomly select a fixed proportion of the travelers regardless of who they are and which routes they are using. We then find the new shortest path for them based on the new travel times. This is one of the methods that can be used in the current version of TRANSIMS.

The procedure for computing a dynamic user equilibrium is as follows.

1. Compute the travel time on each link using a link performance function (or using simulation). In the first iteration use free-flow speeds to compute the travel times and select all travelers to be routed.

2. Trace shortest path trees from each origin to all destinations by using the travel times from Step 1.
3. Assign all selected trips from each origin to each destination along the corresponding shortest paths (all-or-nothing assignment).
4. Randomly select 20% of travelers and reroute them (find shortest path for them based on the new travel times).
5. If the solution has converged sufficiently (see below), stop; otherwise return to Step 1.

The convergence criteria used could be based on any of the following remaining almost unchanged: (a) the travel times on the links, (b) the volumes on each of the links, (c) the paths of travelers, or (d) the travel times on the different routes having the same origin-destination pair. The last of these criteria is used for our example.

Table-5.7 presents the results for this algorithm applied to above test network. Since we randomly select 20% of travelers and reroute them, we do not need any extra information about the network (supply side). In other words, in this method, only the demand side of traffic is being considered. The traffic assignment is performed regardless of the supply side of the traffic, such as the volume to capacity ratio.

The results seem to be oscillating. The reason for this is that we take a fixed percentage of people to be rerouted. It might be better if we vary the proportion of people to be rerouted in each iteration. For example, in the early iterations, we could select a higher proportion of people to be rerouted to reduce the number of travelers on congested links faster, and then select a lower proportion of travelers to be rerouted in the later iterations to avoid oscillations. Although after 24 iterations the travel times on the routes are still different, they are relatively close to each other.

5.7.1.2 Variable Random Selection

The previous algorithm could be improved by varying the proportion of travelers to be rerouted. The volume to capacity ratio, V/C, provides an idea of what percentage of users should be rerouted. We chose this proportion to be the link-maximum (V/C)*10% over all the routes, where the link-maximum V/C is the V/C of the most congested link in

the network. (This choice is open to further study.) This method uses supply side information, *V*/*C*, to assign traffic in a better manner. The resulting algorithm can be described as follows.

1. Compute the travel time on each link using a link performance function (or using simulation). In the first iteration use free-flow speeds to compute the travel times and select all travelers to be routed.
2. Trace shortest path trees from each origin to all destinations by using the travel times from Step 1.
3. Assign all selected trips from each origin to each destination along the corresponding shortest paths (all-or-nothing assignment).
4. Randomly select link-max(*V*/*C*)*10% of travelers to be rerouted.
5. If the solution has converged sufficiently (using the same criteria as delineated in Section 3.1.1), stop; otherwise, return to Step 1.

Table-5.8 presents the results for this algorithm using the above test network. Almost the same solution as the previous model is attained in half the number of iterations (15th iteration versus 24th iteration).

Another way of variable random selection is a method similar to the method of successive averaging. In this method the proportion of travelers to be selected is gradually decreased. In our example, the fraction of selected travelers to be re-routed were taken as 20%, 20%, 20%, 20%,20%, 15%, 15%, 15%, 15%, 15%, and 10%. Almost the same solution as the previous model (even better results) is attained in 10 iterations as shown in Table-5.9.

Table-5.7: Dynamic User Equilibrium Assignment Solution (Fixed Random Selection)

Iteration	Link A		Link B		Link C	
	Flow	Time	Flow	Time	Flow	Time
0	8000	9231.0	0	20.0	0	21.0
1	6400	3789.9	1600	20.24	0	21.0
2	5120	1561.2	2880	22.54	0	21.0
3	3779	474.05	2621	21.75	1600	25.08
4	3770	469.49	2664	21.86	1566	24.70
5	2366	85.50	4196	31.48	1438	23.66
6	1655	31.86	4068	30.14	2277	37.74
7	1481	25.82	4276	32.38	2243	36.74
8	1490	26.08	4269	32.30	2241	36.69
9	3077	216.70	3859	28.21	1064	21.80
10	1595	29.56	3792	27.65	2613	50.02
11	1329	22.01	5203	47.14	1468	23.89
12	2298	77.70	4556	35.96	1146	22.07
13	1872	42.65	4476	34.59	1652	25.63
14	1193	19.56	3809	27.80	2998	71.24
15	1629	30.82	3737	27.22	2634	50.95
16	1249	20.47	5008	43.30	1743	26.74
17	1637	31.16	4925	41.80	1437	23.66
18	1397	23.57	3892	28.49	2711	54.62
19	1907	44.78	3518	25.67	2575	48.35
20	521	15.16	4920	41.69	2560	47.70
21	549	15.20	4898	41.31	2554	47.46
22	713	15.58	4752	38.88	2535	46.69
23	1019	17.42	4592	36.47	2389	41.27
24	1520	27.02	4292	32.56	2188	35.26

Table-5.8: Dynamic User Equilibrium Assignment Solution
(Variable Random Selection based on *V/C*)

Iteration	Link A		Link B		Link C		*V/C*	*V/C*	*V/C*	%
	Flow	Time	Flow	Time	Flow	Time	A	B	C	Rerouting
0	8000	9231.0	0	20.0	0	21.0	8	0	0	80%
1	1600	29.75	6400	82.14	0	21.0	1.60	2.13	0	21%
2	635	15.36	5661	58.04	1704	26.25	0.63	1.89	1.13	18%
3	1546	27.85	4794	39.57	1659	25.72	1.55	1.60	1.11	16%
4	1451	24.97	4237	31.94	2312	38.77	1.45	1.41	1.54	15%
5	2228	70.44	3758	27.38	2015	31.25	2.23	1.25	1.34	22%
6	1259	20.65	5228	47.66	1514	24.29	1.26	1.74	1.01	17%
7	2587	115.77	4041	29.87	1373	23.21	2.59	1.35	0.91	25%
8	1928	46.09	3887	28.46	2185	35.17	1.93	1.29	1.46	19%
9	407	15.06	5423	52.03	2170	34.80	0.41	1.81	1.45	18%
10	1120	18.54	4778	39.30	2102	33.15	1.12	1.59	1.40	15%
11	1891	43.77	4367	33.47	1742	26.73	1.89	1.45	1.16	18%
12	854	16.20	4319	32.89	2827	60.72	0.85	1.44	1.88	18%
13	1931	46.28	3415	25.04	2654	51.89	1.93	1.14	1.77	19%
14	481	15.12	4906	41.45	2613	50.01	0.48	1.63	1.74	17%
15	1629	30.84	4583	36.33	1789	27.37	1.63	1.53	1.19	

Table-5.9: Dynamic User Equilibrium Assignment Solution
(Variable Random Selection Similar to Method of Successive Average)

Iteration	Link A		Link B		Link C		%
	Flow	Time	Flow	Time	Flow	Time	Rerouting
0	8000	9231.0	0	20.0	0	21.0	20%
1	5120	1561.2	2880	22.54	0	21.0	20%
2	4437	887.12	1963	20.55	1600	25.08	20%
3	3932	552.83	3117	23.5	951	21.5	20%
4	2593	116.72	3067	23.28	2340	39.66	20%
5	2140	62.22	3665	26.68	2194	35.43	15%
6	1599	29.71	4724	38.44	1677	25.92	15%
7	809	15.96	4473	34.83	2718	54.96	15%
8	1080	18.05	4317	32.86	2603	49.58	15%
9	1907	44.75	3855	28.18	2238	36.62	15%
10	1567	28.57	4317	32.87	2116	33.46	10%

5.7.1.3 Variable Selection of Targeted Travelers

In this method we select users experiencing relatively high travel times to be rerouted, rather than make a random selection. This method is supply-demand based that uses the interaction between demand and supply to hopefully find a better solution. The procedure used could be described as follows.

1. Compute the travel time on each link using a link performance function (or using simulation). In the first iteration use free-flow speeds to compute the travel times and select all travelers to be routed.
2. Trace shortest path trees from each origin to all destinations by using the travel times from Step 1.
3. Assign all selected trips from each origin to each destination along the corresponding shortest paths (all-or-nothing assignment).
4. Select link-max (V/C)*10% of travelers who have the highest travel time to be rerouted.
5. If the solution has converged sufficiently (using the same criteria as delineated in Section 3.1.1), stop; otherwise, return to Step 1.

Table-5.10 displays the results for this algorithm. In this method, we target the travelers who have the highest travel time, and corresponding to that, we reroute some proportion of these travelers based on the link-maximum V/C value (the most congested link). In other words, we use the information from both sides of traffic, the demand side and the supply side, to obtain an improved traffic assignment methodology. This method appears to produce lesser oscillations but still, the results are not as good as for the static method. The solution at the third iteration is almost as good as the 15th iteration for the method in Section 3.1.2 and the 24th iteration for the method in Section 5.3.1.1. However, as shown in Table-5.10, we cannot derive better results by performing more iterations which is true for all proposed aggregated methods.

Table-5.10: Dynamic User Equilibrium Assignment Solution (Variable Selection of People Having the Highest Travel Time)

Iteration	Link A Flow	Link A Time	Link B Flow	Link B Time	Link C Flow	Link C Time	V/C A	V/C B	V/C C	%
0	8000	9231.0	0	20	0	21	8	0	0	80%
1	1600	29.75	6400	82.14	0	21	1.6	2.13	0	21%
2	1600	29.75	5056	44.2	1344	23	1.6	1.68	.896	17%
3	1600	29.75	4196	31.48	2204	35.68	1.6	1.4	1.47	15%
4	1931	46.28	4196	31.48	1873	28.66	1.9	1.4	1.25	19%
5	1564	28.46	4196	31.48	2240	36.66	1.56	1.4	1.49	14%
6	1878	42.96	4196	31.48	1926	29.56	1.88	1.4	1.28	19%
7	1540	27.56	4196	31.48	2203	35.66	1.54	1.4	1.47	

5.7.2 Incremental Loading

All the above dynamic user equilibrium methods seem to need a considerable number of iterations to give relatively good results, which could be time-consuming.

Since each traveler minimizes his/her own travel time, he/she uses the fastest links to travel. If each person does the same, the fastest routes would become congested. The traffic assignment process (for finding static or dynamic user equilibria) tries to reroute some of these people to avoid congestion. It might be less time consuming if we first load only 50% of people to take the fastest routes, and then update the travel time of each link (using link performance functions or simulation) and then route the next 30% of travelers, and finally load the remaining 20%. In this way, we imitate the disaggregated routing method but using batches of travelers in lieu of routing each individual user one at a time, and also, we do not perform any rerouting loop. Hence, the procedure is faster. This method has been implemented in the Portland study, selecting travelers in order of a specified list.

Since we are going to use link performance functions in order to prevent running the Microsimulator frequently in TRANSIMS, we need information regarding the volume on each link, which is not available in the current Route Planner. One way to circumvent this problem might be to only consider the supply side of traffic. We can begin by loading

only the first 50% of travelers, then increase (virtually) the travel time of the used links by decreasing (virtually) the free-flow speeds of the links and then load the next 30% of travelers, and finally the remaining 20%. In this way we do not need to use link performance functions or simulation to find the travel times. Thus, we do not need to change the current Route Planner to count volumes on each link. In other words, we do not need any demand side information to assign the traffic. We just use the supply side of traffic, and thereby obtain a faster assignment process. Link travel times depend mainly on the free-flow speeds and the volumes on the links. In this method, we increase the travel time of those links that are used by the first loaded group of travelers by simply virtually increasing their free-flow speeds to inhibit other travelers from choosing these links and making them more congested.

Since we are going to use these methods in TRANSIMS, the Microsimulator will adjust the travel times after the Route Planner, and within a few iterations, the traffic would hopefully reach a dynamic user equilibrium.

5.8 Summary and Conclusion

A new heuristic algorithm for finding dynamic user equilibria has been presented. This algorithm, which is a modification of the convex-simplex method, routes each traveler within an identified set one at a time on the network and updates the link volumes and link travel times after each routing. Using an appropriate descent-based stopping criterion, finite termination of the procedure is guaranteed. The solution produced depends upon the order in which the travelers are considered for (re-)routing.

The proposed model is disaggregated and is suitable for the microscopic and mesoscopic models such as TRANSIMS, CONTRAM, DYNASMART, and DynaMIT. Although some existing models such as CONTRAM update link travel times after routing each packet of travelers, they address the non-convergence issue of the algorithm differently. The contribution of this study is that it adopts a more effective update criterion for re-routing travelers based on the user equilibrium definition, thereby facilitating the determination of a near-equilibrium after a few iterations.

The proposed algorithm was tested on two small networks and was shown to converge rapidly to an equilibrium solution (in the first iteration) in the sense that all the routes having the same origin-destination pair have the same travel time, and no traveler can achieve a lesser travel time by switching to another route.

A large-scale version of the proposed model was implemented within TRANSIMS as a two-stage process to design a framework for real-world transportation systems. The implemented algorithm within TRANSIMS is both analytical and simulation-based to take advantage of their individual features, speed, and precision in calculating link travel times. This implemented method was applied to the Bignet network, which represents one-tenth of the transportation network in Portland, Oregon. Using the Incremental Individual Loader in the first stage of this process causes the travelers to be distributed better in the network with regard to other travelers, thereby reducing the subsequent number of iterations required between the Route Planner and the Microsimulator. In our experiment, a near-user equilibrium was found after only three iterations, consuming less than 20-33% of the effort required by the current version of TRANSIMS (over 16 iterations) and yet producing an improved distribution of travelers as well as stable travel times for the travelers.

Chapter 6

Enhancements to Emission Estimation in TRANSIMS

6.1 Introduction

Emissions from vehicle sources have a major effect on urban air quality. According to the Clean Air Act Amendments of 1990 (CAAA), non-attainment areas are required to submit emission estimates for all proposed traffic-improvement projects. Therefore, a reliable and accurate emission estimation model is needed. It has been found that most of the current emissions estimators do not provide accurate results. These models use an aggregate representation of traveler behavior and then estimate emissions based on typical driving cycles, vehicle miles traveled, and average speeds, and then supplement the formulation by corrections for cold-start, evaporation, and high emitting vehicles. Vehicle emissions are mostly produced from off-cycle driving, cold-starts, vehicles malfunctioning, and climbing steep grades. Typical driving cycles, on the other hand, produce low emissions. Thus, the existing techniques underestimate the emissions. In addition, the existing models also do not estimate the effects of green wave signalization. Green wave signalization allows vehicles to be in phase with the green lights by traveling at or near the speed limit. The Transportation Analysis and Simulation System (TRANSIMS), sponsored by the U.S. Department of Transportation, the Environmental Protection Agency, and the Department of Energy, addresses most of the above problems in its emissions estimator module (see TRANSIMS homepage, 2004). The module considers cold-starts, enrichment cycles, grades, and vehicle malfunctioning.

A brief overview of the framework of TRANSIMS is presented next. TRANSIMS consists of the following modules, all of which are integrated with respect to their inputs and outputs.

- **Population Synthesizer:** This estimates the number of synthetic households, the demographics, and characteristics of each individual within these households, and the locations of these households on the network.

- **Activity Generator:** This creates an activity list for each synthetic traveler. These activities include work, shopping, school, etc. These activity estimations are based on the activity demographic characteristics of individuals, and available survey data. In addition, activity times and activity locations are determined for each individual.

- **Route Planner:** This computes combined route and mode trip plans to accomplish the desired activities of each individual, such as work, shopping, etc.
- **Microsimulator:** This uses the intermodal paths developed in the Route Planner module to perform a regional microsimulation of vehicle interactions. The microsimulation continuously computes at every second the operating status, including locations, and speeds of all vehicles throughout the simulation period. The output can provide a detailed, second-by-second history of every traveler in the system over a 24-hour period.
- **Feedback Controller:** This manages the feedback of information among the Activity Generator, the Route Planner, and the Traffic Microsimulator modules of TRANSIMS.
- **Emissions Estimator:** Using the vehicle information generated in the Microsimulation module, the emission module forecasts the nature, amount, and location of motor vehicle emissions. The focus of the present chapter is on this module.

The remainder of this chapter is organized as follows. We first briefly explain the emissions estimation model within TRANSIMS and describe the algorithmic procedure of the module. We then apply the Emissions Estimator module to a real-world network and compare the results with on-road emission-measurement (OEM) data. Finally, a summary of the chapter and the conclusions are provided.

6.2 TRANSIMS' EMISSIONS ESTIMATOR

The Emissions Estimator in the present version of TRANSIMS produces estimates for:

1) tailpipe emissions from light-duty vehicles (LDVs),

2) tailpipe emissions from heavy-duty vehicles (HDVs), and

3) evaporative emissions,

as a function of vehicle fleet composition, fleet status, and fleet dynamics. This chapter focuses on the tailpipe emissions from LDVs. The Emissions Estimator module thus requires information regarding:

- the fleet composition developed from the Population Synthesizer,

- inspection and maintenance test results obtained from local and national databases, and

- traffic patterns produced by the Traffic Microsimulator module.

The emission inventory obtained from TRANSIMS form the basis for the computation of pollutant concentrations in a metropolitan area based on atmospheric conditions, local transport and dispersion, and chemical reactions.

The Microsimulator adopts a cellular-automata principle that gives the vehicle position in units of cells, velocity in units of cells per second, and the acceleration in units of cells per second per second. Since the cell size is 7.5 meters, the resulting movement in 16mph increments is too coarse to estimate the emissions. Therefore, smooth vehicle trajectories are generated and used in the emissions estimation. The output of the Microsimulator is aggregated into 30 meter segments, 7.5 m/s speed bins, and over an hour. This converted output from the Microsimulator is used as the major input to the Emissions Estimator. The following algorithm is used in the Emissions Estimator module.

6.2.1 Emissions Estimator Algorithm Overview

In TRANSIMS, empirical information on power demands is used to estimate the emissions due to the lack of information describing the range of driving behavior under various circumstances. Three sets of empirical data are used as follows.

The Comprehensive Modal Emissions Model, CMEM, was developed by Barth and his colleagues at the University of California at Riverside and the University of Michigan. Barth and his co-investigators were contracted by the National Cooperative Highway Research Program (NCHRP) to develop an improved modal emission model for LDVs. CMEM computes the tractive power by taking into account the engine-friction losses, rolling resistance, wind resistance, changes in kinetic energy, changes in potential energy, and the power necessary to drive accessories such as air conditioning. With engine power known, CMEM calculates the rate of fuel consumption and tailpipe emissions. It treats enrichment, enleanment, and stoichiometric operations, as well as cold-start operations. There are two CMEM arrays, one that reflects emissions at constant power and another that reflects differences in emissions associated with changes in power from one second to the next.

The EPA three-cities data is used to estimate the distribution of high power and hard braking events. This three-cities study gives the cumulative distribution of acceleration for hard acceleration (VA > 50), hard braking (VA < -50), and insignificant acceleration (-50 < VA < 50), where VA is the velocity-acceleration product and is used to represent power.

The California Air Resource Board (CARB) data is a collection of vehicle trajectories on freeways and arterials having different levels of congestion. It is also used for deriving calibration constants. CARB sponsored a chase car study in Los Angeles to collect the data. The resulting Cal Poly on-ramp data, collected by the California Polytechnic State University at San Luis Obispo, gives the distribution of velocities and accelerations on ramps.

The converted Microsimulator output is used to calculate the number of vehicles for each 4-mile per hour (mph) speed bin and 10-mph squared per second power bin. This is done to obtain an array having the same structure as that of the CMEM arrays. The choice of emission arrays to have 4 mph speed bins and 10-mph squared per second power bins is driven from the sensitivity of emissions to power and speed. The range of power is considered from -150 to 180-mph squared per second in 10-mph squared per second increments. Therefore, there are a total of 34 power bins. The speeds are

considered to range from 2 mph to 78 mph in 4 mph increments. Thus, there are a total of 20 speed bins. A flow chart of the TRANSIMS overall emission estimator module is shown in Figure-6.1.

The first step is to estimate the population of vehicles in each 4 mph speed cell from the population in the 7.5 meter per second speed bins for each link segment. This is carried out by constructing a continuous distribution of the number of vehicles by speed group by assuming that the continuous speed distribution within each speed bin can be approximated by a linear function of speed. The result is also used to calculate the standard deviation of speed, the average square of the speed, and the average cube of the speed in each segment.

There is a nonlinear relationship between power and emissions. The average power in each segment is estimated using the average cube of the speed. The probability of hard acceleration, hard braking, and insignificant acceleration is influenced by the average power and also by the standard deviation of speed. It is found that there exists a simple linear relationship between either the gradient of the average cube of the speed or the standard deviation of the speed, and the probability of hard acceleration. In practice, the probability of hard acceleration or hard braking is calculated using the average cube of the speed and the standard deviation of speed. Using the approach developed by investigators at the University of Michigan, the total flux[2] is broken into thirds and the probability of hard acceleration (braking) is calibrated for each third separately. These are distributed over the power bins using the cumulative VA distributions from EPA's three-cities data. An adjustment is made to represent the emissions associated with the step change in power.

Fleet composition is developed from vehicle registration data, inspection, and maintenance testing, or from data developed from EPA's Mobile5 model runs. The registration data is used to produce vehicle populations in each of the 23 LDV vehicle categories. The categories include factors such as low or high engine-to-weight ratio, car or truck, mileage above or below 50,000, type of catalyst (2-way or 3-way), carbureted or fuel-injected, and high or normal emitting vehicles.

[2] The first moment of speed.

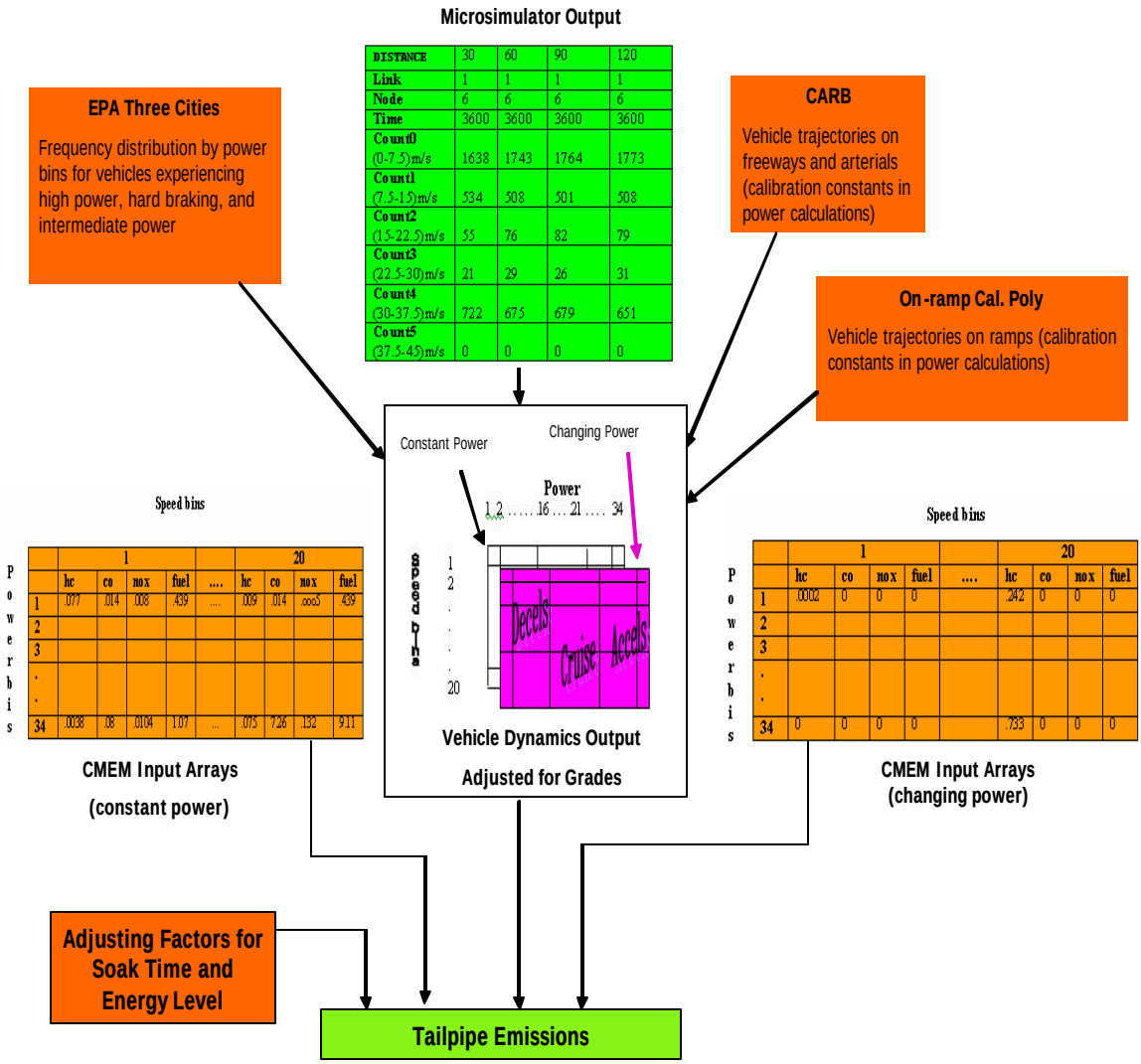


Figure-6.1: TRANSIMS Emission Estimator Framework.

After calculating the proportion of vehicles in each of the 20 speed bins and 34 power bins for each segment of each link and constructing two speed-power matrices, one for constant power and another for the distribution curve for the difference in vehicle power, these matrices are multiplied by the corresponding CMEM emission arrays. The obtained matrices for the constant power and for the difference in emissions between constant power trajectories and those with the same speed and power, but with a step change in power, are summed up. The CMEM arrays give the composite vehicle emissions in terms of HC, CO, NOx, and fuel consumption in 4-mph speed bins and 10-mph squared per second power bins. The street grades are incorporated into the analysis by using different CMEM emissions arrays for each street grade classification.

Finally, the calculated values are adjusted for soak-time and energy level by multiplying them by the pollutant adjustment factors. The pollutant adjustment factors are calculated from the converted Microsimulator output and from the CMEM emission rates for one hour soak-time versus zero soak-time. The emission rates or multipliers that are computed for eight energy groups, represent the ratio of emissions for vehicles on a link in the group to the emissions of a vehicle with the same driving pattern pertaining to a completely warmed-up engine and catalyst. The converted Microsimulator output is the number of vehicles entering each link over each hour grouped by soak-time and by the integrated velocity-acceleration product. The velocity-acceleration product, which presents the eight energy levels, is used as a surrogate for fuel consumption to give the engine and catalyst warm-up level. Soak-time is the time the engine was off before the start of the current trip. Four different soak-times are used: no soak, short soak, medium soak, and long soak-time.

6.2.2 The Detailed Emissions Estimator Algorithm

The stepwise algorithm of the Emissions Estimator module is presented here along with an example.

Step 1 : Input data from Microsimulator

Each link in the highway network is divided into 30m segments. (The last segment on the link in the direction from the traveling node could be less than 30m in length.) Each

segment is divided into 4 cells of 7.5 meters each. The output from the Microsimulator provides the number of vehicles in each 7.5 meter per second speed cell for each 30-meter segment. Figure-6.2 shows an example of the output file obtained from the Microsimulator. (This has been taken from an LANL document[3], and is an example used for illustrative purposes only, and does not represent a real-world situation.) This file records the number of vehicles N_{ij} (where i represents the speed bin, and j represents the segment) for 6 different velocity bins for the 17 segments of link I.D 1, traveling from node I.D 6, and starting at time 3600 seconds from midnight. Notice that the last segment is only 15 meters long instead of 30 meters. A schematic description of link I.D.1 and its segments are shown in Figure-6.3.

The distribution of vehicles per speed bin on segment 9 of this link is depicted in Figure-6.4. (Segment 9 is used as the demonstration segment for all the calculations to follow.)

DISTANCE	LINK	NODE	TIME	COUNT0 0-7.5 m/s	COUNT1 7.5-15 m/s	COUNT2 15-22.5 m/s	COUNT3 22.5-30 m/s	COUNT4 30-37.5 m/s	COUNT5 37.5-45 m/s
…	…	…	…	…	…	…	…	…	…
270	1	6	3600	16381	5346	551	218	722	0
300	1	6	3600	17431	5086	764	293	675	0
330	1	6	3600	17640	5015	827	268	679	0
360	1	6	3600	17733	5084	791	314	651	0
390	1	6	3600	17985	5135	802	274	665	0
420	1	6	3600	18042	5099	857	259	654	0
450	1	6	3600	18083	5141	835	288	620	0
480	1	6	3600	18157	5328	807	299	605	0
495	1	6	3600	9156	3107	358	184	234	0

Figure-6.2: Example of Speed Summary Output File from the Traffic Microsimulator.

[3] Los Alamos National Laboratory documents, LA-UR-00-1725, TRANSIMS 3.0, Volume three, Chapter 7

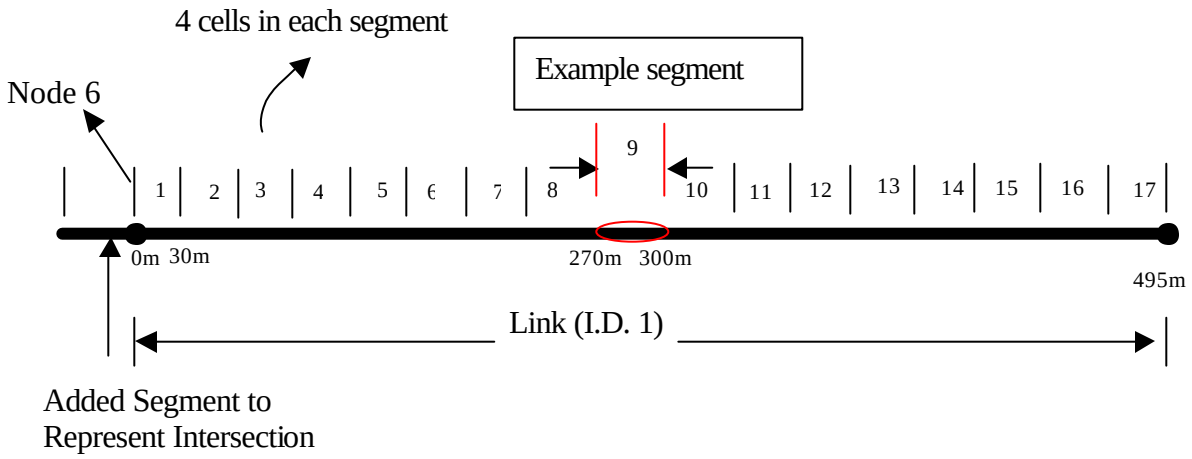


Figure-6.3: Schematic Description of Link I.D. 1.

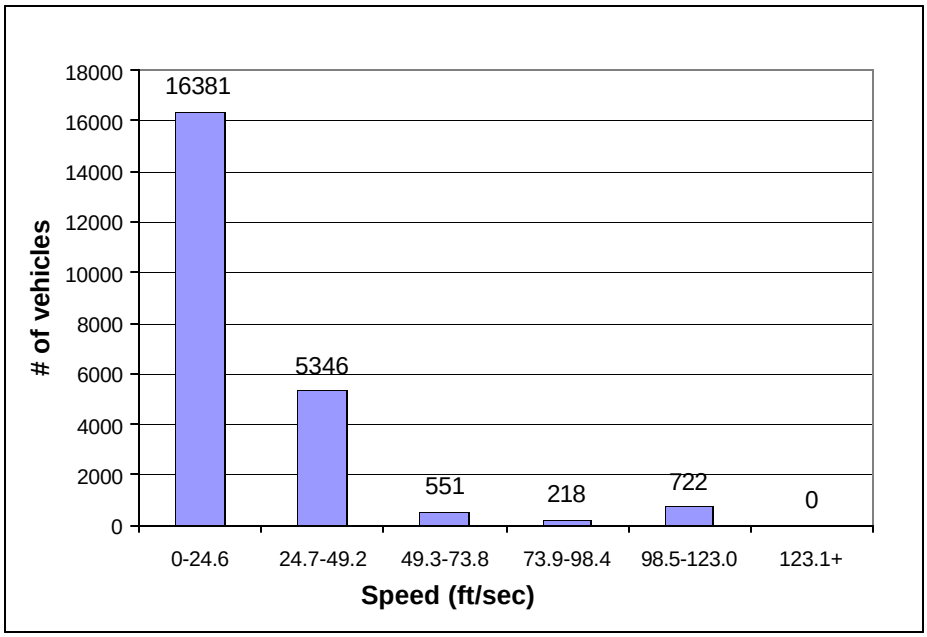


Figure-6.4: Distribution of Number of Vehicles per Speed Bin on Segment 9 of Link I.D 1.

<u>Step 2</u> : Creating a Continuous Distribution of Vehicle Densities within and among the Speed Bins

As stated earlier, the population of vehicles in each 4 mph (i.e., 5.867 ft/sec or 1.79 m/s) speed cell is estimated from the population in the 7.5 m/s speed bins for each link segment. That is, a continuous fit is obtained for the densities d_{ij} (number of vehicles per unit speed and per unit space). This is done by assuming that the continuous speed distribution within each speed bin can be approximated by a linear function of the form:

$d_{ij}(dv) = f_{ij} + h_{ij}dv$ (i represents the speed bin and j represents the particular spatial cell under consideration)

where $\delta v = v - v_c$, v being the chosen speed within the speed bin and v_c is the speed at the center of the bin. Hence, at the top of any of the speed bins, $\delta v = v - v_c = \Delta/2$ (with Δ being 7.5 meters per second). Similarly, at the bottom of the speed bin, $\delta v = v - v_c = -\Delta/2$. The foregoing relationship is therefore defined for each speed bin for values of $dv \in [-\Delta/2, \Delta/2]$. The term f_{ij}, which is the average vehicle density per spatial cell (4×24.6 feet) and per speed cell (24.6 ft/sec), is given by the following, assuming that vehicles are evenly distributed among the 4 spatial cells:

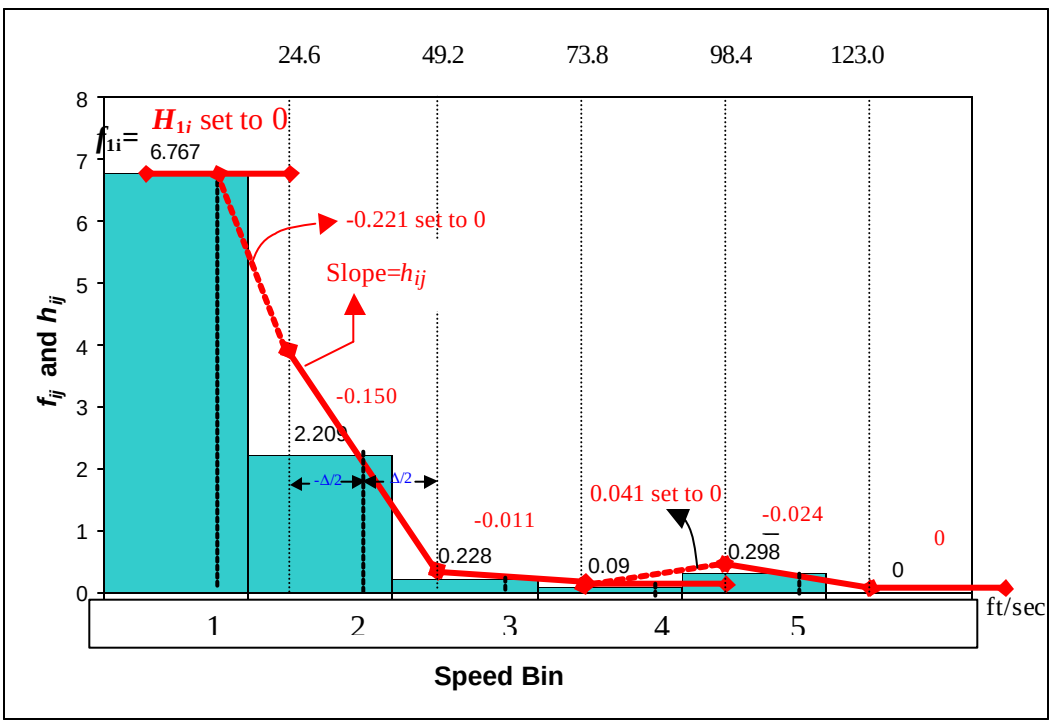
$$f_{ij} = N_{ij}/(4\Delta^2) \qquad (1)$$

where the length of the segment box is 4Δ and the width in velocity space is Δ. Hence, note that $\int_{l=0}^{\Delta} \int_{dv=-\Delta/2}^{\Delta/2} [f_{ij} + h_{ij}dv] \cdot d(dv) \cdot dl = N_{ij}$. The slopes ($h_{ij}$) are found by setting d_{ij} to be zero at the top of the highest speed bin that contains vehicles, and solving for h_{ij}. Continuity relationships are then used to determine the h_{ij} values for the slower speed bins. Also, h_{1j} is always arbitrarily set to zero. However, this procedure may lead to negative densities over a portion of a speed bin. If a negative density is obtained, it is resolved in one of two ways:

1. If the potential negative value for the density occurs in an intermediate speed bin i, the continuity condition at the speed bin boundary is relaxed and h_{ij} is set to zero.

2. If the negative value occurs in the slowest of the speed bins that have vehicles, the density relationship is assumed to hold down to a value of dv, denoted dv_I , where the density falls to zero and remains there. In other words, in the slowest populated speed bin, the distribution extends from dv=(D/2) to dv=dv_I rather than to dv=-D/2.

A plot of the relationship between f_{ij} and speed is shown Figure-6.5. Figure-6.5 also displays the d_{ij} values based on the foregoing continuity relationships at the boundaries of the speed bins, as well as the linear relationship within each speed bin.



* All positive h_{ij} are set to zero because they yield negative densities at the lower bound δv=-Δ/2, and also, the h_{1j} value of -0.221 for the lowest speed bin is set to zero.

Figure-6.5: The f_{ij} and h_{ij} Values for Example Segment 9.

<u>Step 3</u>: Determining Densities and the Flows (Flux)

Once a continuous distribution to the spatial and speed populations from the Microsimulator is constructed, the spatial average vehicle densities in the various speed bins are computed. The spatial average density for each speed bin is obtained by integrating the continuous linear density relationship from -Δ/2 to Δ/2 (i.e. using the zero eth moment) as follows:

$$\text{spatial average density} = \int_{-\Delta/2}^{\Delta/2} \left[f_{ij} + h_{ij} dv \right] \cdot ddv = f_{ij}\Delta \equiv \frac{N_{ij}}{4\Delta} \quad \text{veh/ft}$$

The number of vehicles over the entire cell of length Δ is calculated by:

$$Num_i(j) = \int_0^{\Delta} \int_{-\Delta/2}^{\Delta/2} \left[f_{ij} + h_{ij} dv \right] \cdot ddv \cdot dl = f_{ij}\Delta^2 \equiv \frac{N_{ij}}{4} \; veh. \qquad (2)$$

The total spatial density of the segment is obtained by summing the densities over all the speed bins within the segment, and is given by $\sum_i \int\int d_{ij}(dv).d(dv).dl$.

The flux over the entire cell, which is defined as the first moment of speed, is determined by integrating the product of density and speed for each speed bin and for each cell. This is computed as follows:

$$flux_i(j) = \int_{l=0}^{\Delta} \int_{dv=-\Delta/2}^{\Delta/2} [\text{speed} \times \text{density}] \cdot d(dv) \cdot dl$$

$$flux_i(j) = \int_0^{\Delta} \int_{-\Delta/2}^{\Delta/2} [\Delta(i-0.5) + dv][f_{ij} + h_{ij} dv] \cdot ddv \cdot dl = \Delta^3 (i-0.5) f_{ij} + \frac{\Delta^4 h_{ij}}{12} \quad \text{veh/sec} \qquad (3)$$

A plot of the spatial average density versus speed for example segment 9 is given in Figure-6.6.

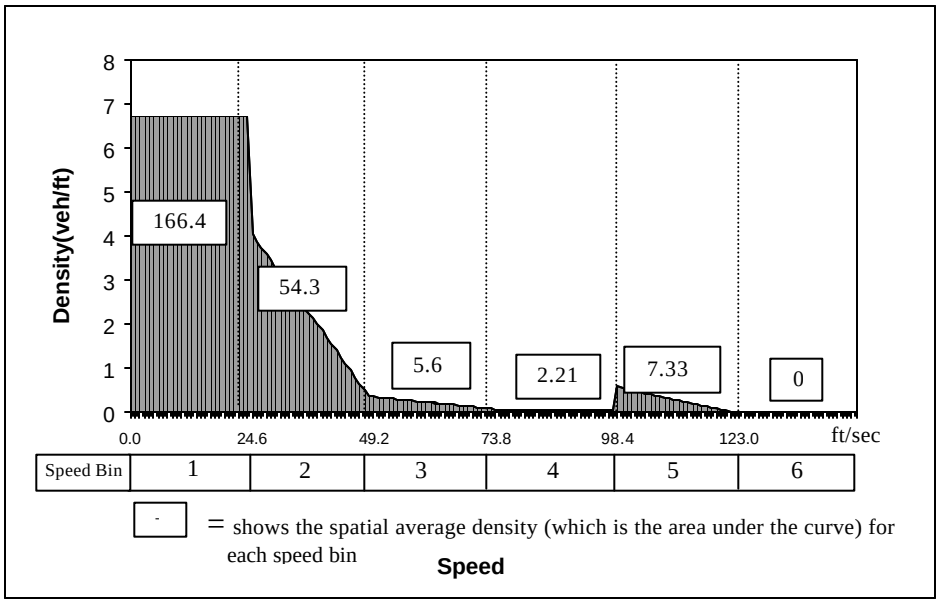


Figure-6.6: Spatial Average Density within Each Speed Bin for Example Segment 9.

Since dv is considered to be the speed relative to the center of the bin and varies between $-\Delta/2$ and $\Delta/2$, the speed in each speed bin is calculated by: $v = \Delta(i-.5) + dv$. Note that TRANSIMS considers the speed in each speed bin as given by $v = \Delta(i-1) + dv$, which is demonstrated to be incorrect using an example as shown below.

Suppose that we calculate the speed of the 4th speed bin when $dv = -1/4$. By our calculation, the speed is given by $v = 3.5\Delta - 1/4$, while the TRANSIMS procedures would estimate the speed as $3\Delta - 1/4$ which is incorrect because this speed lies in the third bin as opposed to the fourth bin, as demonstrated in Figure-6.7. Therefore the flux that is computed in TRANSIMS as follows

$$flux_i = \int_{-\Delta/2}^{\Delta/2} [\text{speed} \times \text{density}] = \int_{-\Delta/2}^{\Delta/2} [\Delta(i-1) + dv]\ [f_{i0} + h_{i0} dv] \cdot ddv$$

would be incorrect given that the speed is computed incorrectly.

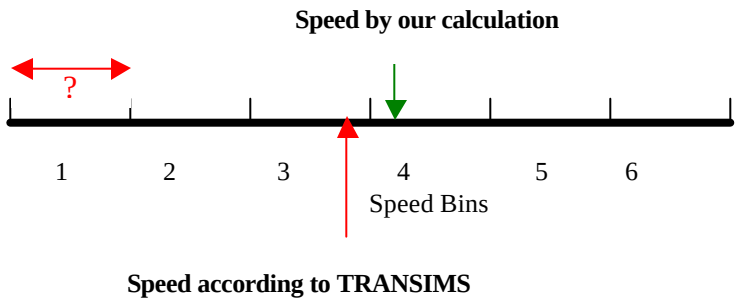


Figure-6.7: Speed Calculation for the Speed Bin 4 and $dv = -1/4$.

A plot of (speed × density) within each speed bin is shown in Figure-6.8. The area under this curve over each speed bin *i*, times Δ, yields the $flux_i(j)$, and the sum of these quantities yields the total flux, $flux_{total}(j)$.

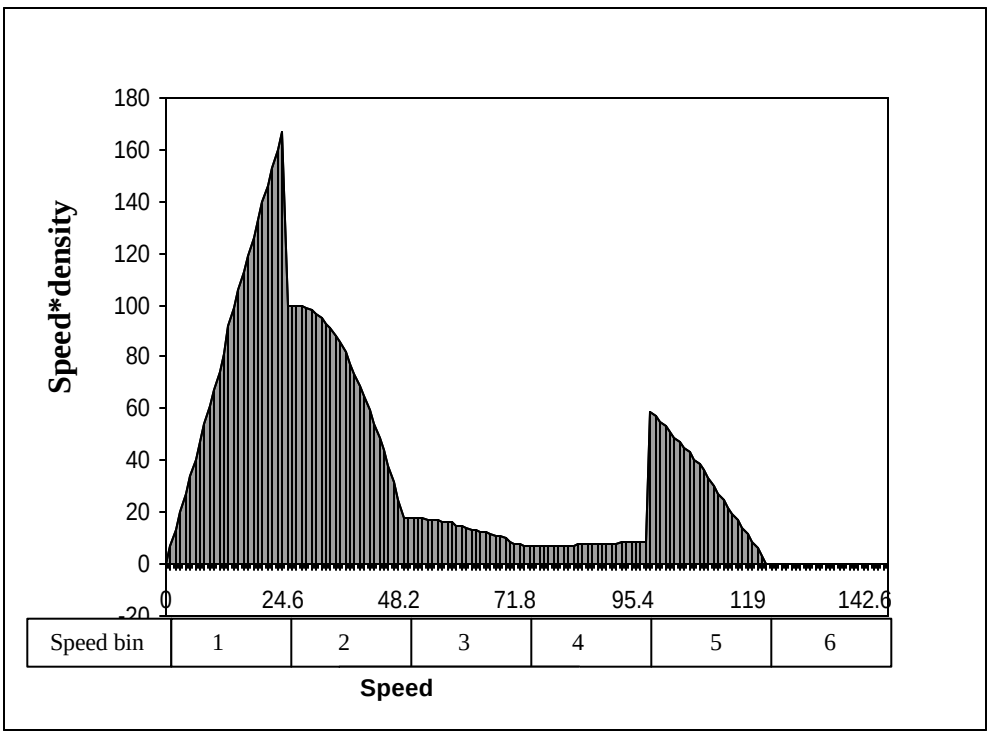


Figure-6.8: Distribution of Flux within each Speed Bin for Example Segment 9.

<u>Step 4</u>: Determining the Properties of the Speed Distribution within a Segment

As stated earlier, the variation in speeds influence the acceleration and deceleration rates and the use of power by vehicles, and consequently, impacts their emissions. In this step, the variation of speeds within a segment are determined by finding the average speed ($\bar{v}$), the speed coefficient of variation (ratio of standard deviation of speed to average speed) (*sdevrat*), the lower and upper limits of speed (*vlowri*, and *vuppri*), and the comparison of standard deviation of speed to that of an uncongested freeway (*v2sdev*). The average speed is given by:

$$\bar{v} = \frac{\sum_i \int \int v.d_{ij}(dv).d(dv).dl}{\sum_i \int \int d_{ij}(dv).d(dv).dl} = \frac{total\ \ flux(j)}{total\ \ number(j)} \equiv \frac{flux_{total}(j)}{Num_{total}(j)}. \qquad (4)$$

To calculate the standard deviation of speed (σ), the second moment of speed ($\overline{v^2}$) is computed as

$$\overline{v^2} = \frac{\sum_i \int_0^l \int_{-\Delta/2}^{\Delta/2} [\Delta(i-0.5)+dv]^2 \left[f_{ij} + h_{ij}dv\right] \cdot ddv \cdot dl}{Num_{total}(j)} \qquad (5)\ .$$

Finally, the standard deviation σ of speed, is computed as

$s = \sqrt{(\overline{v^2} - \bar{v}^2)}$. The coefficient of variation (s_{rat}) is computed as;

$$s_{rat} = \frac{s}{\bar{v}}.$$

For determining the lower and upper limits of speeds, the breakpoints between the slowest one-third (*vlowri*), the middle one-third, and the fastest one-third (*vuppri*) of the flux need to be determined. This is done by first finding the index i_{low} such that:

$$\sum_{i=1}^{i=i_{low}-1} flux_i(j) < flux_{total}(j)/3 \le \sum_{i=1}^{i=i_{low}} flux_i(j),$$

(where the first sum is zero if i_{low}=1), and then solving the following equation to obtain $d_{low.}$

$$\int_{0}^{\Delta}\int_{-\Delta/2}^{d_{low}}\left[\Delta(i_{low}-0.5)+dv\right]\left[f_{i_{low}j}+h_{i_{low}j}dv\right]\cdot ddv\cdot dl = flux_{total}(j)/3-\sum_{i=1}^{i=i_{low}-1}flux_i(j).$$

Once δ_{low} has been calculated, *vlowri* is calculated as:

$$vlowri = d_{low}+\Delta(i_{low}-0.5). \qquad (6)$$

The procedure for determining *vuppri* is similar, except that the coefficient of $flux_{total}$ is two-thirds rather than one-third as in the equation for d_{low} and in the inequality equations.

The variable *v2sdev* is given by:

$$v2sdev = \bar{v}^2(s - s_r) \qquad (7)$$

where,

s = is the standard deviation of speed (calculated earlier)

s_r = is the standard deviation of speed for uncongested freeways, used as a constant of 11.1 ft/sec.

Figure-6.9 shows the breakpoints for each one-third of the flux for the Example Segment 9.

<u>Step 5</u> : Determining the Properties of the Flux Distribution and the Gradient of Power

In this step, the speed parameters for each third of the flux, namely, *spdcl* for the slowest third, *spdcm* for the middle third, and *spdch* for the fastest third are first estimated. Then speeds are estimated by averaging the speed gradients over two segments, *itar* and *iref*, in order to compensate for the faster average accelerations found for one of these segment in the Microsimulator.

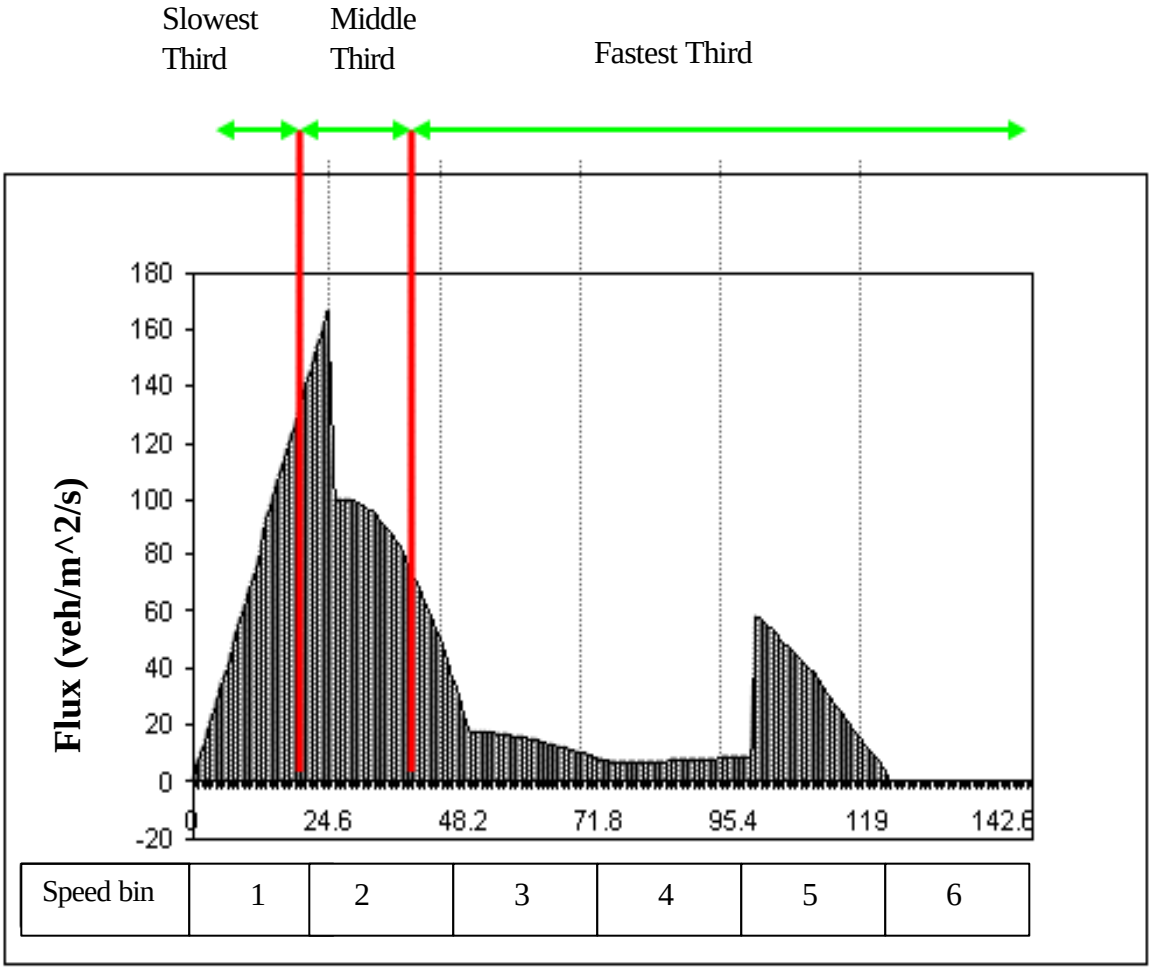


Figure-6.9: Breakpoints for each One-third of the Flux in Example Segment 9.

The two parameters, *itar* and *iref*, give the two segments from which the gradient is computed. The gradients are calculated as the differences observed between the preceding segment, denoted by *icx*-1 (*iref*), and the following one *icx*+1 (*itar*), to give a centered difference (*icx* is the current segment). For example when the current segment is 9 (*icx*=9), then *iref*= 8 and *itar*=10. In the case of the first segment (*icx*=1), a simple forward difference is used as an estimate of the gradient so that *iref* =1 and *itar* =2 are used.

To estimate these speeds, the average cube of the velocity, where rate of change with respect to distance represents power

$$(1/3)(dv^3/dx) = (1/3)(dv^3/dv)\frac{(dv/dt)}{(dx/dt)} = v(dv/dt) = vA,$$

is determined as follows for each third of the flux and for each segment in the gradient estimation:

$$\overline{v^3(j)} = \frac{\sum_i \int_0^{\Delta} \int_{-\Delta/2}^{d_i} [\Delta(i-0.5)+dv]^3 \left[f_{ij} + h_{ij}dv\right] \cdot ddv \cdot dl}{\sum_i \int_0^{\Delta} \int_{-\Delta/2}^{d_i} \left[f_{ij} + h_{ij}dv\right] \cdot ddv \cdot dl} = \frac{vcubedl(j)}{vehdl(j)}, \qquad (8)$$

where j refers to either *itar* or *iref*, and the sums ($\sum_i$) run over only the speed bins in the slowest third, and d_i is defined as follows:

$$d_i = \begin{cases} \Delta/2 & for\ i < i_{low} \\ d_{low} & for\ i = i_{low}. \end{cases}$$

For the middle third we can calculate a numerator (called *vcubedm*(j)) that runs to the top of the second third of the flux and subtract from it the numerator given above (called *vcubedl*(j)). The denominator (*vehdm*(j)) can be calculated similarly. We can likewise calculate the numerator for the top third (*vcubedh*(j)) by calculating a numerator that covers all the speed bins and subtract from the numerator for the slowest two-thirds. The denominator (*vehdh*(j)) is computed similarly.

The speed parameter for the slowest one-third is then given by

$$spdcl(j) = \frac{\overline{v^3(itar)} - \overline{v^3(iref)}}{(itar - iref) \bullet 4\Delta^3}, \qquad (9)$$

where the averages are for the slowest third as computed above. Averages for the middle and highest thirds produce *spdcm*(j) and *spdch*(j), respectively.

The next calculation provides the flux for the slowest third for the current segment as given by:

$$vehfluxl(j) = \sum_{i=1}^{i=i_{low}-1} flux_i(j) + \int_0^{\Delta} \int_{-\Delta/2}^{d_{low}} [\Delta(i_{low} - 0.5) + dv][f_{i_{low}j} + h_{i_{low}j} dv] \cdot ddv \cdot dl, \qquad (10)$$

which is merely one-third of the total flux by the definition of $d_{low,}$ and i_{low}. The other thirds of the flux are calculated similarly. The middle third (*vehfluxm*(*j*)) and the fastest third (*vehfluxh*(*j*)) of the flux are also calculated similarly.

Figure-6.10 shows the average cube of the velocity for each third of the flux (where $\overline{v^3(j)}$ is the area under the figure divided by *vehdl*(*j*)) for example segments 9 and 10.

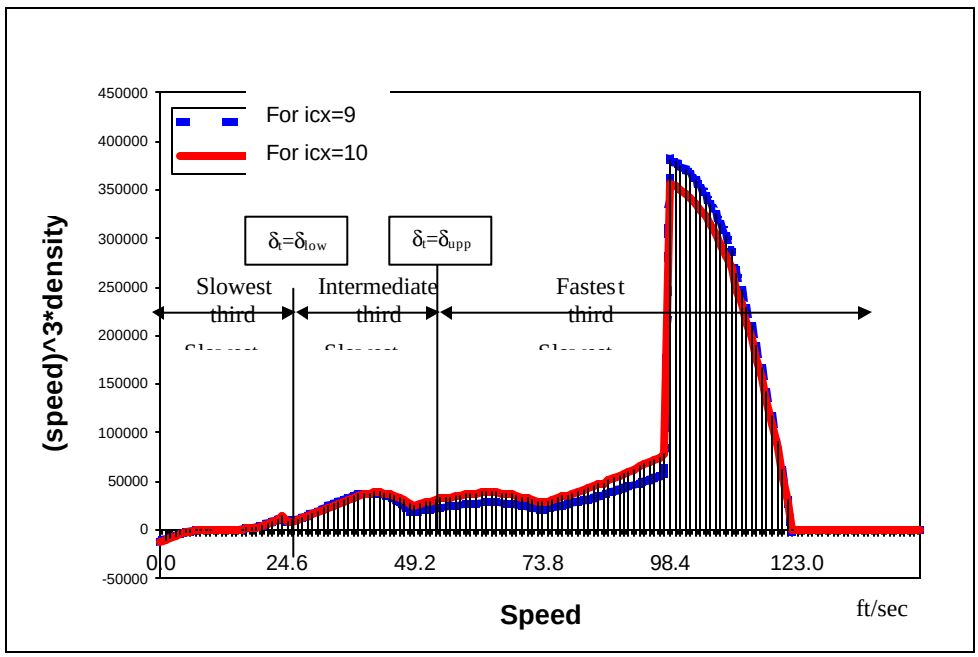


Figure-6.10: Average Cube of the Speed for each Third in Example Segments 9 and 10.

<u>Step 6</u> : Determining the Probability of Hard Acceleration, Hard Braking, and Intermediate Driving

As stated earlier, the probability of high power (hard braking) driving is influenced by the standard deviation of speeds, because a large standard deviation of speeds implies that vehicles are below their desired speed and will accelerate to higher speeds.

The probability of high power (hard braking) driving is influenced also by the average power. The average power is estimated by changes in the average cube of the speed.

In practice the probability of hard acceleration (braking) is calculated using the average cube of the speed and also the standard deviation of speed and the maximum of them is used. The coefficients are from CARB an On-Ramp data.

The probability of intermediate driving is calculated by the fact that the summation of the three probabilities (hard acceleration, hard braking, and intermediate power) is equal to one.

$$P = \frac{powa \cdot a}{(1 + \{10 + 40\ \ ef)a\,]vehd(j) \cdot fluxcor(j)} \quad , \quad Pd = \frac{powd \cdot b}{(1 + [10 + 40ef\,]b).vehd(j) \cdot fluxcor(j)}$$

$$Pns = 1 - P - Pd$$

where P is the probability of a hard acceleration, Pd is the probability of a hard braking, and Pns is the probability of an intermediate power. Other variables and parameters are explained as follows.

- $powa$ ($powd$) is the maximum of the two power factors based on the average cube of the speed and also the standard deviation of speed.
- $fluxcor(j)$ is a factor used to normalize various quantities to a single trajectory basis. It is calculated by: $fluxcor(j) = \frac{\Delta}{Avgflux(j)}$, where $Avgflux(j)$ is the average of the total flux in each segment.

- a and b are the exponent parameters in the exponential decay of the cumulative frequency of power for high power driving and hard braking for different link types and driving types, as obtained from CARB.

Step 7: Distributing Vehicles into Power and Speed Bins

Using the speed variation and the empirical data power distribution, the high power, intermediate power, or hard braking vehicles are distributed into power bins. Two matrices of 34 power bins and 20 speed bins (similar to CMEM arrays) are constructed: one for constant power and another for changing power. The choice of 4 mph speed bins and 10 mph squared per second power bins is driven from the sensitivity of emissions to power and speed.

The power levels include:

- 14 high-power levels (hard acceleration) (50-180 in increments of 10)
- 5 insignificant power levels (cruise) (0-40 in increments of 10)
- 15 low-power levels (deceleration) (-150 to-10 in increments of 10).

High-power demands are defined by velocity-accelerations greater than 50 mph squared per second, which corresponds with the 10% point on the cumulative distribution. Conversely, low-power demands are defined by velocity-acceleration products less than –50 mph squared per second.

The Emission Module begins by estimating the number of vehicles that are demanding high power. Fifteen different high-power levels are selected to represent different levels of driver aggressiveness. These levels are selected from the curve found in Figure-6.11. They have equal spacing in power and cover a range from a cumulative frequency of 0.1 to 0.0045. The total vehicle population demanding high power from a given speed is then distributed over the 15 power levels. The distribution of vehicles in each speed bin and power bin in the high power driving is calculated by multiplying the probability of hard acceleration by the population associated with driver aggressiveness, and is given by

$$pop = \frac{\Delta^2}{8}[\, f_{icvj} + (ivv - 4.5)h_{icvj}\,](\frac{e^{-0.3(ipa-1)-e_0} - e^{-0.3ipa-e_0}}{e^{-e_0}})$$

where $[f_{icv.j} + (ivv - 4.5)h_{icv.j}]$ is the density (d_{ij}) function value, and the item in the parenthesis in the formula is the fraction of high-power vehicles that have the three-cities power index *ipa*. The value of *ipa* is between 1 and 15, and corresponds to equidistant points in the power space along the curve of the cumulative probability versus power for vehicles that are in the high-power driving mode. The same procedure is used for hard braking and intermediate power driving.

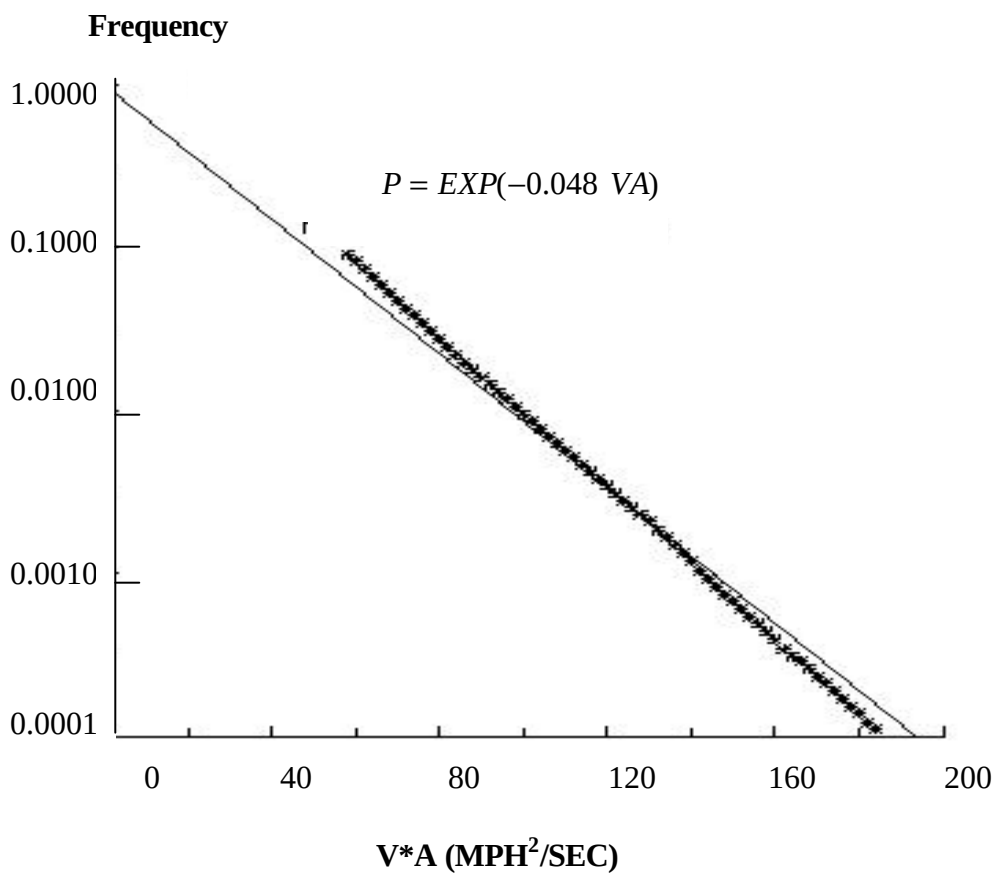


Figure-6.11: The Cumulative Distribution of Accelerations from the EPA's Three-City Studies.

Step 8: Determining Fraction of Vehicles that Change Power

Since all the above calculations relate to vehicles traveling under constant power, a correction for vehicles that are changing power needs to be made, because they can have different emissions than those traveling at constant power. It is assumed that each driver will have the same aggressiveness from one second to the next, such that if the car at the present time is at the 80th percentile, it will have been at the 80th percentile in the preceding second. Also, vehicles that do not change segments (their position) during the preceding second, would have constant power. For vehicles that change position, their percentile is the same, but the power level may change because the fraction of vehicles under high-power driving may be different for the preceding speed and segment. Therefore, the power at the preceding second can be calculated, and then the fractional power change relative to the current power can be found via the power at the current second and the preceding second.

Step 9: Calculate and Adjust the Emissions

The first sub-step is to multiply the elements of the calculated matrices by the corresponding CMEM arrays to find the emissions. Then the calculated emissions are adjusted according to the soak-time and energy level. The adjusting factors are calculated by multiplying the CMEM emissions ratios by the corresponding converted Microsimulator energy output. A multiplier is assigned for each parameter, hydrocarbon, carbon monoxide, nitrogen oxide, and fuel consumption. Each multiplier represents the corresponding ratio of emissions for vehicles traveling at the beginning of a link in a particular soak group to the emissions of a vehicle with the same driving pattern with a fully warmed-up engine and catalyst. The final Emissions Estimator output for the example is presented in Figure-6.12.

TIME	LINK	NODE	DIST.	LENGTH	VTT	NOx (mg)	CO (g)	HC (mg)	FE (g)	FLUX
3600	1	6	270	30.0	5.8	5115.3	470.2	97870.4	8780.8	29420.0
3600	1	6	300	30.0	8.5	17834.9	649.9	122554.8	11653.1	51106.0
3600	1	6	330	30.0	8.6	17964.6	657.4	124129.6	11793.0	52369.0
3600	1	6	360	30.0	8.6	17994.5	660.8	124856.8	11858.1	52582.0
3600	1	6	390	30.0	8.5	17841.8	665.7	126126.9	11960.3	52789.0
3600	1	6	420	30.0	8.5	17819.4	666.7	126390.5	11978.7	52871.0
3600	1	6	450	30.0	8.5	17680.6	667.2	126661.7	11996.3	52862.0
3600	1	6	480	30.0	8.4	17482.2	669.9	127492.7	12065.9	53000.0
3600	1	6	495	30.0	11.9	20268.2	499.9	86833.1	8736.3	48065.0

Figure-6.12: Emissions Estimator Output.

6.3 Comparison of TRANSIMS, CMEM, VT-Micro, and OEM Data

For the sake of comparisons with TRANSIMS, we used a portable on-road emission measurement device, OEM-2100^{TM}, manufactured by Clean Air Technologies International, Inc, which is designed to collect on-road emission data. This device consists of two analyzers, an engine diagnostic scanner, and an on-board computer that provides second-by-second emissions, fuel consumption, engine speed, and engine temperature. The emissions include hydrocarbons (HC), carbon monoxide (CO), carbon dioxide (CO_2), and oxides of nitrogen (NO_x). A more detailed description of the OEM device can be found in Rakha et al. (2004).

A 1999 Ford Crown Victoria was utilized for collecting field data. The data were gathered along a 3-km section of Route 460 in Blacksburg, Virginia, from the Tom's Creek Rd. exit to the North Main St. exit. The roadway is a 55 mph speed limit highway facility, and was selected for the study because it is part of the TRANSIMS Blacksburg network. The data were collected including six repetitions of normal driving along the study section. The speed profiles of the six runs are presented in Figure-6.13.

The input data that were used for the TRANSIMS Emissions Estimator were the CMEM emission arrays, distribution of vehicles, soak-ratio files, and the Microsimulator input data including velocity summary data and energy summary data.

The CMEM emission arrays yield the fuel consumption and tailpipe emissions (NOx, CO, and HC) for a composite vehicle in each speed bin and power bin. There are two types of emission arrays, one reflecting emissions at constant power (*arrayp.out*) and the other type reflecting differences in emissions associated with changes in power from one second to the next (*arraypd.out*). Street grades have been incorporated and four street grade categories for each type of the CMEM emission arrays were used. These grade categories are less than 1% and downhill grades, between1% and 3%, between 3% and 5%, and greater than 5%. The input files for TRANSIMS are depicted in Figure-6.14. Since the test road section is fairly flat, we used the emission arrays of the first category, i.e., less than 1% and downhill.

One of the inputs of the Emissions Estimator is the proportion of the fleet in each of the 23 vehicle categories. As stated before, the CMEM emission arrays provide the emissions for a composite vehicle of the 23 vehicle types. Therefore, the distribution of vehicles in these 23 categories is used to construct the CMEM emission arrays. The distribution of vehicles and the composite CMEM emission arrays for the Portland, Oregon, study are presented in Table-6.1 and Figure-6.14.

The test vehicle is vehicle Type-11 in the 23 CMEM vehicle types and the results of the OEM are based on this vehicle type. Due to the unavailability of single vehicle emission arrays and the availability of the Portland composite vehicle emission arrays, the latter inputs are used in TRANSIMS for the purpose of emission estimation. However, in an attempt to validate the use of a composite vehicle as opposed to the use of a single vehicle, the VT-Micro model was run twice once using an identical composite vehicle and once with data for the Crown Victoria that was tested as part of the study.

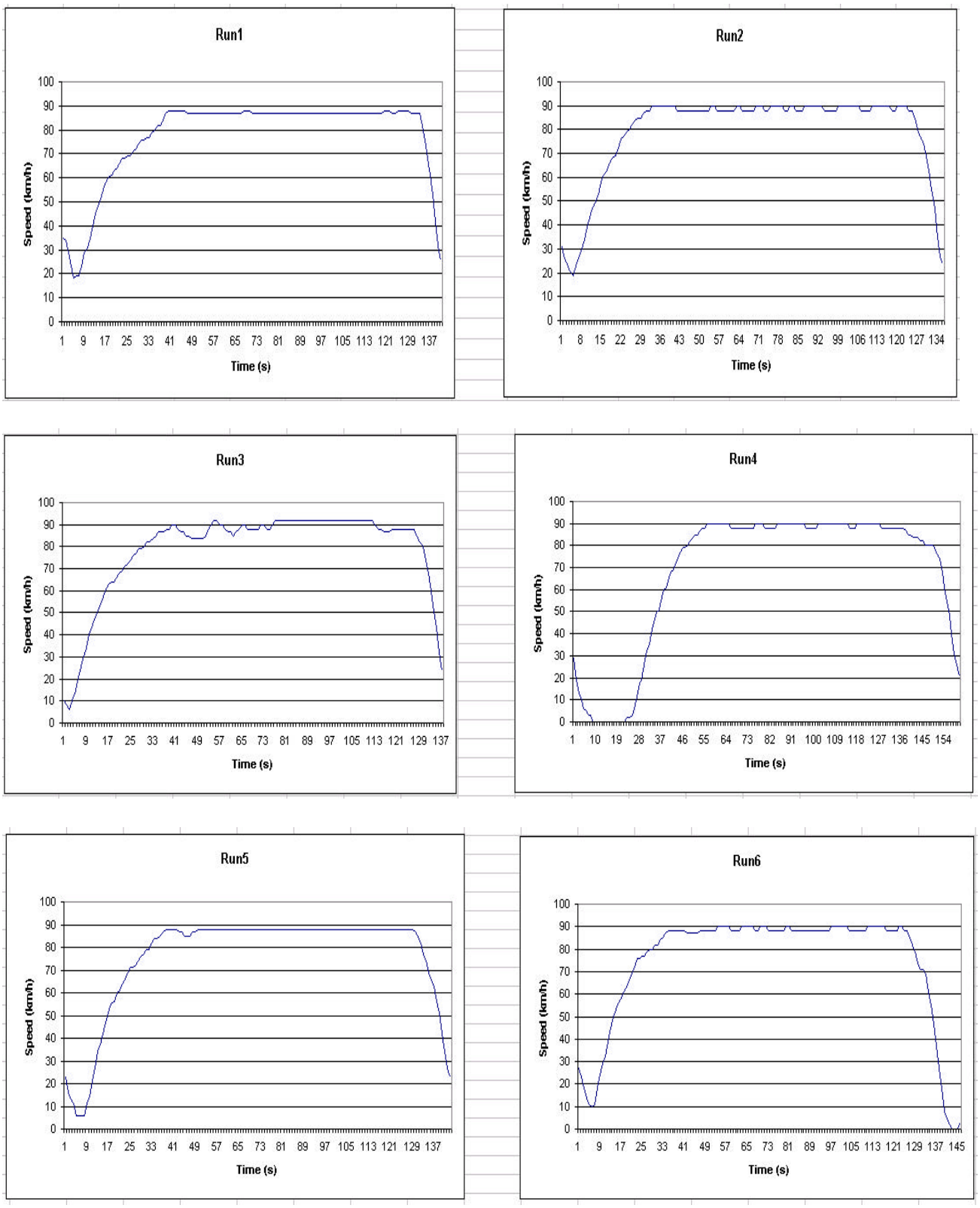


Figure-6.13: The Speed Profile of the Test Runs 1 to 6.

AARAYP.OUT

V	pow	hc (mg/s)	co (mg/s)	nox (mg/s)	fuel (g/s)
2.0000	-150.0000	0.0777	0.0140	0.0008	0.4393
6.0000	-50.0000	0.0478	0.0140	0.0008	0.4393
10.0000	-30.0000	0.0264	0.0140	0.0008	0.4393
14.0000	-21.4286	0.0136	0.0140	0.0008	0.4393
18.0000	-16.6667	0.0077	0.0140	0.0007	0.4393
22.0000	-13.6364	0.0101	0.0140	0.0007	0.4393
26.0000	-11.5385	0.0096	0.0140	0.0006	0.4393
30.0000	-10.0000	0.0092	0.0140	0.0005	0.4393
34.0000	-8.8235	0.0088	0.0140	0.0004	0.4393
38.0000	-7.8947	0.0087	0.0140	0.0004	0.4393
42.0000	-7.1429	0.0086	0.0140	0.0004	0.4393
46.0000	-6.5217	0.0086	0.0140	0.0004	0.4393
50.0000	-6.0000	0.0088	0.0140	0.0007	0.4393
54.0000	-5.5556	0.0085	0.0140	0.0004	0.4393
58.0000	-5.1724	0.0106	0.0140	0.0007	0.4393
62.0000	-4.8387	0.0086	0.0140	0.0004	0.4393
66.0000	-4.5455	0.0085	0.0140	0.0004	0.4393
70.0000	-4.2857	0.0086	0.0140	0.0004	0.4393
74.0000	-4.0541	0.0087	0.0140	0.0004	0.4393
78.0000	-3.8462	0.0089	0.0140	0.0005	0.4393
2.0000	-140.0000	0.0711	0.0140	0.0008	0.4393
6.0000	-46.6667	0.0419	0.0140	0.0008	0.4393
…	…	…	…	…	…

ARRAYPD.OUT

V	pow	hc (mg/s)	co (mg/s)	nox (mg/s)	fuel (g/s)
2.0000	-150.0000	0.0002	0.0000	0.0000	0.0000
6.0000	-50.0000	0.0028	0.0000	0.0000	0.0000
10.0000	-30.0000	0.0297	0.0000	0.0000	0.0000
14.0000	-21.4286	0.0776	0.0000	0.0000	0.0000
18.0000	-16.6667	0.1329	0.0000	0.0000	0.0000
22.0000	-13.6364	0.2145	0.0000	0.0000	0.0000
26.0000	-11.5385	0.2979	0.0000	0.0000	0.0000
30.0000	-10.0000	0.2036	0.0000	0.0000	0.0000
34.0000	-8.8235	0.1905	0.0000	0.0000	0.0000
38.0000	-7.8947	0.1879	0.0000	0.0000	0.0000
42.0000	-7.1429	0.1885	0.0000	0.0000	0.0000
46.0000	-6.5217	0.1815	0.0000	0.0000	0.0000
50.0000	-6.0000	0.1845	0.0000	0.0000	0.0000
54.0000	-5.5556	0.1906	0.0000	0.0000	0.0000
58.0000	-5.1724	0.1978	0.0000	0.0000	0.0000
62.0000	-4.8387	0.2067	0.0000	0.0000	0.0000
66.0000	-4.5455	0.2139	0.0000	0.0000	0.0000
70.0000	-4.2857	0.2198	0.0000	0.0000	0.0000
74.0000	-4.0541	0.2285	0.0000	0.0000	0.0000
78.0000	-3.8462	0.2420	0.0000	0.0000	0.0000
2.0000	-140.0000	0.0002	0.0000	0.0000	0.0000
6.0000	-46.6667	0.0028	0.0000	0.0000	0.0000
…	…	…	…	…	…

Figure-6.14: Part of the Portland's *ARRAYP.OUT* and *ARRAYPD.OUT* Files.

Table-6.1: Distribution of LDV vehicles in Portland

#	Vehicle Technology Category	Fraction
	Normal Emitting Cars	Portland
1	No Catalyst	.0161
2	2-way Catalyst	.0323
3	3-way Catalyst, Carbureted	.0673
4	3-way Catalyst, FI>50K miles, low power/weight	.1330
5	3-way Catalyst, FI,>50k miles, high power/weight	.1330
6	3-way Catalyst, FI,<50K miles, low power/weight	.0247
7	3-way Catalyst, FI,<50K miles, high power/weight	.0247
8	Tier 1, >50K miles, low power/weight	.0112
9	Tier 1, >50K miles, high power/weight	.0112
10	Tier 1, < 50K miles, low power/weight	.0676
11	Tier 1,<50K miles, high power/weight	.0676
	Normal Emitting Trucks	
12	Pre-1979 (<=8500 GVW)	.0251
13	1979-1983(<=8500 GVW)	.0315
14	1984 to 1987(<=8500 GVW)	.0423
15	1988 to 1993, <=3750 LVW	.0461
16	1988 to 1993,>3750 LVW	.1062
17	Tier 1 LDT2/3 (3751-5750 LVW or Alt. LVW)	.0251
18	Tier 1 LDT4(6001-8500 GVW,>5750 Alt. LVW)	.0165
	High Emitting Vehicles	
19	Runs lean	.0180
20	Runs rich	.0332
21	Misfire	.0410
22	Bad catalyst	.0138
23	Runs very rich	.0150

Source: Hobeika et al., 2003

The three **SoakRatios* files (*shortSoakRatios*, *mediumSoakRatios*, and *longSoakRatios*) contain the composition ratios of cold emissions to hot engine emissions for each of eight power levels and for each of the emissions components. These three 4-by-8 ratios are obtained from the CMEM model. These input files exist in the TRANSIMS package. Figure-6.15 presents part of these input files.

The inputs from the Microsimulator are the most important inputs to the Emissions Estimator including information about vehicles' speeds and energy levels. In this study, the velocity summary file was obtained from the collected OEM data rather than the Microsimulator's output. The field data were converted to the Microsimulator's output data format and were fed to this module. The data consist of the number of vehicles in each velocity bin in each 30-meter segment on the test link, summed up over 3600

seconds. Since only one car on only one section of a road was tested, the input data and therefore the output are summed up for the entire trip on this link, which is approximately 140 seconds rather than 3600 seconds. There are 97 segments of 30 meters on this particular link. The collected second-by-second speed from the OEM was converted into the six velocity bins and 30-meter distances (as explained before). The OEM data yields the second-by-second speeds of the vehicle, fuel consumption, and emissions. The distance traveled by the car was calculated and added to the OEM data.

longSoakRatios	Nox	CO	HC	FE
Energy1	9.123511	2.901801	2.961656	1.086365
Energy2	5.529905	1.700297	1.683761	1.06226
Energy3	3.025764	1.187867	1.389111	1.095758
Energy4	2.089955	1.068815	1.238639	1.119742
Energy5	1.620233	1.112526	1.186073	1.129309
Energy6	1.400889	1.095014	1.175341	1.117272
Energy7	1.263096	1.073836	1.163962	1.104985
Energy8	1	1	1	1

mediumSoakRatios	Nox	CO	HC	FE
Energy1	6.896555	1.739459	1.880872	1.051399
Energy2	3.814291	1.037508	1.355388	1.053422
Energy3	1.921431	1.009059	1.150231	1.087935
Energy4	1.548569	1.058795	1.16302	1.097733
Energy5	1.238308	1.088088	1.179852	1.099333
Energy6	1.183109	1.069759	1.135758	1.079037
Energy7	1.144809	1.025809	1.06619	1.04462
Energy8	1	1	1	1

shortSoakRatios	Nox	CO	HC	FE
Energy1	3.841492	0.997112	1.486536	1.029274
Energy2	1.748887	0.945518	1.145299	1.087439
Energy3	1.22432	1.061343	1.121923	1.067599
Energy4	1.171785	0.997115	1.016674	1.024959
Energy5	1.147291	0.978087	1.000541	1.010844
Energy6	1.120022	0.958046	0.982392	1.002553
Energy7	1.096779	0.928529	0.964877	0.996988
Energy8	1	1	1	1

Figure-6.15: The Soak-Ratio Files: *longSoakRatio*, *mediumSoakRatio*, and *shortSoakRatio*.

In the first 30-meter segment, the vehicle's speed was between 7.5-15 m/s in the first, second, and third second of travel. Therefore, in the velocity summary file, we put three for the Count1 and zero for all other columns. Note that the velocity summary file is the summation of the velocity over the entire travel. In the second 30-meter segment, the vehicle's velocity is between 0-7.5 m/s and the vehicle remains in this segment for 5 seconds. Therefore, in the velocity summary file, we put five for the Count0 and zero for all other columns. This procedure continues until the velocity summary file contains all segments of the 3-kilometer link. Figures 6.16 and 6.17 present the OEM data and the conversion of this data to the Emissions Estimator input data format.

The second Microsimulator's output, which is used as the Emissions Estimator input, is the energy summary data. This data includes four files: *energy.no.out*, *energy.short.out*, *energy.medium.out*, and *energy.long.out* that give the distribution of the fraction of vehicles entering the link, stratified by the time-integrated, velocity-acceleration product for each of the file's soak-times (the time the engine was off before the start of the current trip). The soak-times are considered to be less than 600 seconds, between 600 and 1800 seconds, between 1800 and 3600 seconds, and greater than 3600 seconds, respectively, for the four energy files. Since the tested car was fully warmed-up, the car is in the no-soak state. Therefore, only the *energy.no.out* file has a value of one for the test link, link 21088, as shown in Figure -6.17, and all other energy files are zero.

The Emissions Estimator was run using the above input data and the results of the HC, CO, NOx, and fuel consumption estimates for each of the six runs were recorded. Table-6.2 presents the total emissions calculated by TRANSIMS and also the emissions recorded by the OEM device. The results show that the estimation of NOx are consistent with the OEM measurements in terms of magnitude and not in terms of variation across the six test runs. The HC, CO, and fuel consumption estimates from TRANSIMS were significantly higher than the OEM measurements. Again, the variation across the various test runs was not consistent with the field measurements.

Second	fuel[g/s]	NOx[g/s]	HC[g/s]	CO[g/s]	Speed(m/s)	Distance (m)
1	0.26	0.00078	0.0005	0.00322	9.7222	9.7222
2	0.24	0.00067	0.00046	0.00304	9.4444	19.1667
3	0.23	0.0006	0.00043	0.00301	8.0556	27.2222
4	0.22	0.00053	0.00041	0.00292	6.3889	33.6111
5	0.21	0.00048	0.0004	0.00286	5.0000	38.6111
6	0.21	0.00046	0.0004	0.00296	5.2778	43.8889
7	0.21	0.00043	0.0004	0.00296	5.2778	49.1667
8	0.78	0.00158	0.00145	0.0106	6.3889	55.5556
9	1.76	0.00419	0.00319	0.02359	8.0556	63.6111
10	2.09	0.00638	0.00365	0.02733	8.6111	72.2222
11	2.11	0.00811	0.0038	0.02758	9.7222	81.9444
12	2.13	0.00906	0.00384	0.02757	11.1111	93.0556
13	2.2	0.00993	0.004	0.02793	12.5000	105.5556
14	2.24	0.01052	0.00411	0.02829	13.3333	118.8889
15	1.99	0.00953	0.00366	0.02516	14.1667	133.0556
16	1.79	0.00891	0.00341	0.02315	15.2778	148.3333
17	1.66	0.00835	0.00313	0.0215	16.1111	164.4444
18	1.6	0.00831	0.00302	0.02107	16.6667	181.1111
19	1.56	0.00824	0.00302	0.02099	16.9444	198.0556
20	1.46	0.00784	0.00284	0.02052	17.5000	215.5556
21	1.35	0.00736	0.00264	0.02027	17.7778	233.3333
22	1.3	0.00703	0.00254	0.0206	18.3333	251.6667
23	1.25	0.0067	0.00244	0.01984	18.8889	270.5556
24	1.15	0.00609	0.00224	0.01863	18.8889	289.4444
25	0.97	0.00519	0.0019	0.01615	19.1667	308.6111
….	….	….	….	….	….	….

Figure -6.16: The OEM Data.

COUNT0 (0-7.5)	COUNT1 (7.5-15)	COUNT2 (15-22.5)	COUNT3 (22.5-30)	COUNT4 (30-37.5)	COUNT5 (37.5+)	DISTANCE	LINK	NODE	TIME
0	3	0	0	0	0	30	21088	21173	28800
5	0	0	0	0	0	60	21088	21173	28800
0	3	0	0	0	0	90	21088	21173	28800
0	3	0	0	0	0	120	21088	21173	28800
0	1	1	0	0	0	150	21088	21173	28800
0	0	1	0	0	0	180	21088	21173	28800
0	0	2	0	0	0	210	21088	21173	28800
0	0	2	0	0	0	240	21088	21173	28800
0	0	2	0	0	0	270	21088	21173	28800
0	0	2	0	0	0	300	21088	21173	28800
0	0	1	0	0	0	330	21088	21173	28800
0	0	2	0	0	0	360	21088	21173	28800
0	0	1	0	0	0	390	21088	21173	28800
0	0	1	0	0	0	420	21088	21173	28800
0	0	2	0	0	0	450	21088	21173	28800
0	0	1	0	0	0	480	21088	21173	28800
0	0	1	0	0	0	510	21088	21173	28800
0	0	0	1	0	0	540	21088	21173	28800
0	0	0	1	0	0	570	21088	21173	28800
0	0	0	1	0	0	600	21088	21173	28800
0	0	0	1	0	0	630	21088	21173	28800
0	0	0	2	0	0	660	21088	21173	28800
0	0	0	1	0	0	690	21088	21173	28800
0	0	0	1	0	0	720	21088	21173	28800
0	0	0	1	0	0	750	21088	21173	28800
….	….	….	….	….	….	….	….	….	….

Figure-6.17: The Converted OEM Data.

The differences in the emission estimates between those produced by TRANSIMS and the field data can be attributed to two factors: the emissions estimator and/or the aggregation procedures. In an attempt to identify the causes of the observed differences the six test runs were simulated using the CMEM model. The CMEM model was run 23 times, once for each of the 23 vehicles that constitute the Portland composite vehicle. A weighted average emission estimate was then computed to reflect the Portland composite vehicle weight configuration. The HC and CO Emission results from TRANSIMS were found to be similar to the CMEM results. However, the TRANSIMS NOx emissions were underestimated in comparison with the CMEM estimates. Specifically, while the

CMEM absolute NOx emission estimates were higher than the OEM measurements, the cyclic variation was found to be consistent with the field measurements. However, the results demonstrate that both the CMEM and TRANSIMS models fail to capture the cyclic variation in HC and CO estimates that were observed in the field measurements. Since TRANSIMS uses CMEM emission arrays as an input, the inconsistency of HC and CO estimates is transferred to TRANSIMS. Therefore, this inconsistency does not appear to be due to the aggregation procedure.

While it could be argued that a direct comparison of field measurements for a single vehicle against aggregated composite vehicle emission estimates may not be a fair comparison, we demonstrate the consistency in cyclic variations across the two scenarios. Specifically, the VT-Micro model (Rakha et al., 2004) calibrated against the Crown Victoria vehicle was compared to the VT-Micro composite vehicle (23 vehicle average) emission and fuel consumption estimates. VT-Micro model is a nonlinear regression model that is developed based on on-road second-by-second OEM measurements. Although, the absolute emission and fuel consumption estimates were different, the trends of both models were almost identical, as demonstrated in Figure-6.18. The VT-Micro model yields a better estimation of emissions as compared with the other models.

ENERGY0	LINK	NODE	TIME
0	1500	1577	31200
0	1500	1648	31200
0	1501	1648	31200
0	1501	1572	31200
0	1502	1568	31200
0	21086	21171	31200
1	21088	21173	31200
0	21102	20517	31200
0	21102	20179	31200
0	21103	20026	31200
0	21104	20940	31200
0	21127	20951	31200
0	21143	21077	31200
0	21143	20006	31200
0	21148	3381	31200
0	21148	3382	31200
…	…	…	…

ENERGY0	ENERGY1	ENERGY2	ENERGY3	ENERGY4	ENERGY5	ENERGY6	ENERGY7	LINK	NODE	TIME
0	0	0	0	0	0	0	0	1500	1577	31200
0	0	0	0	0	0	0	0	1500	1648	31200
0	0	0	0	0	0	0	0	1501	1648	31200
0	0	0	0	0	0	0	0	1501	1572	31200
0	0	0	0	0	0	0	0	1502	1568	31200
0	0	0	0	0	0	0	0	1502	1582	31200
0	0	0	0	0	0	0	0	21086	21171	31200
0	0	0	0	0	0	0	0	21088	21173	31200
0	0	0	0	0	0	0	0	21102	20517	31200
0	0	0	0	0	0	0	0	21102	20179	31200
0	0	0	0	0	0	0	0	21103	20026	31200
0	0	0	0	0	0	0	0	21104	20940	31200
0	0	0	0	0	0	0	0	21127	20951	31200
0	0	0	0	0	0	0	0	21143	21077	31200
0	0	0	0	0	0	0	0	21143	20006	31200
0	0	0	0	0	0	0	0	21148	3381	31200
0	0	0	0	0	0	0	0	21148	3382	31200
…	…	…	…	…	…	…	…	…	…	…

Figure -6.18: The *energy.no.out* Input File; and the *energy.short.out* , *energy.medium.out,* and *energy.long.out* Input Files.

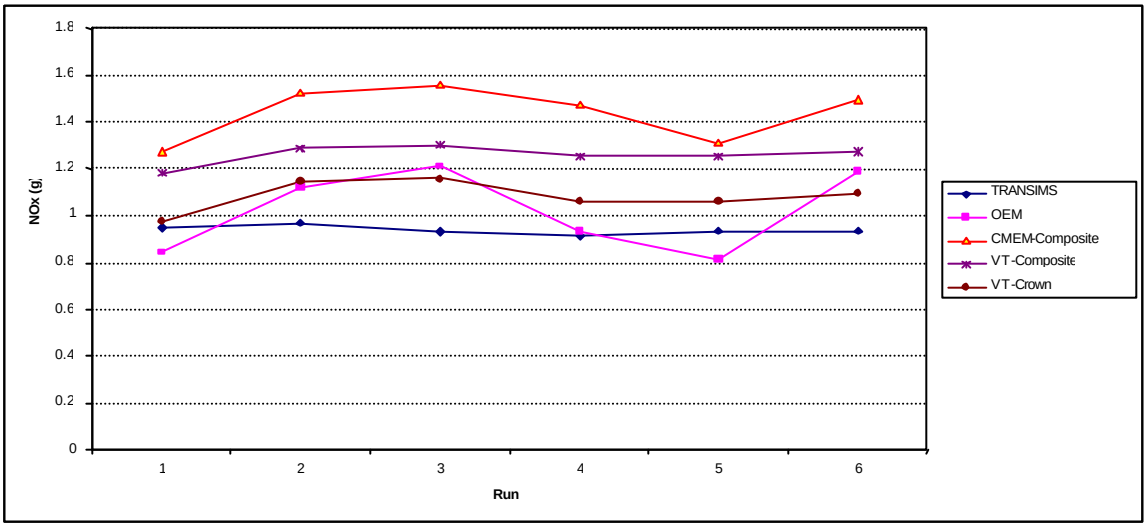
6.4 Summary and Conclusions

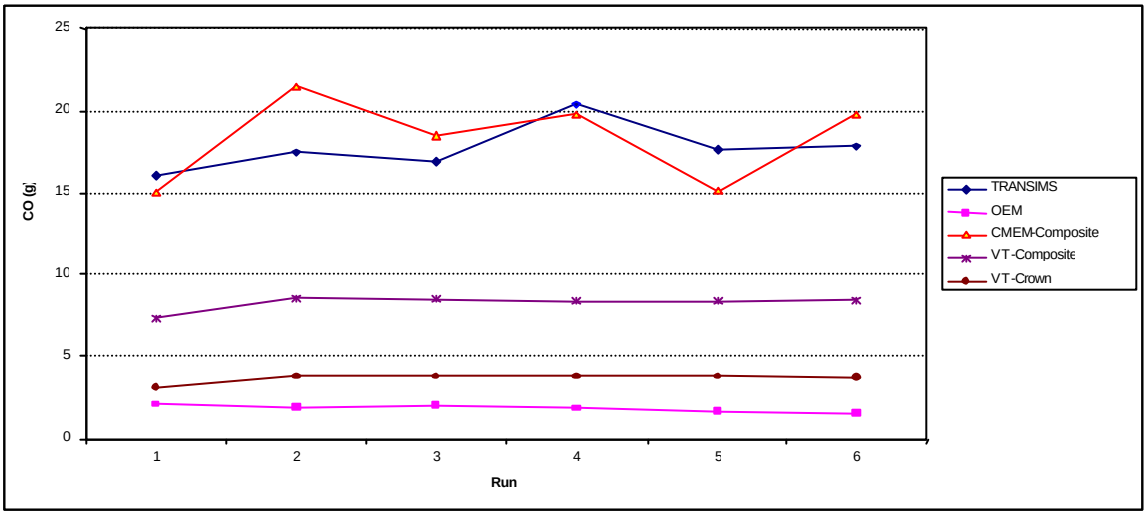
This chapter describes the emission estimation procedures within the TRANSIMS model. TRANSIMS relies heavily on three sets of empirical data pertaining to power demands, which are CMEM, EPA three cities, and CARB. It addresses most of the existing problems in the current emissions models that use an aggregate representation of travel behavior and estimate emissions based on typical driving cycles, using vehicle miles traveled and average speeds. TRANSIMS considers cold-starts, enrichment cycles, grades, and vehicle malfunctioning in emissions calculations. However, the results show that TRANSIMS' emission results do not reflect the differences across drive cycles that were observed in field measurements. The results of the comparison between the OEM data and the emissions estimates of TRANSIMS, CMEM, and the VT-Micro models are summarized as follows.

- Emission results from TRANSIMS, which were generated based on the CMEM model, follow the CMEM model estimates and are significantly different from the OEM data. This inconsistency in emission estimates is due to inherent problems with the CMEM model and not with the data aggregation procedures. Therefore, using other emission models within the TRANSIMS framework could lead to better results.
- The NOx emissions generated from CMEM follow the field measurement trends. However, the CMEM and TRANSIMS models do not follow the field observed trends for the HC and CO field measurements.
- The trends of the VT-Micro models using a composite vehicle and using a single vehicle are almost identical. However, the VT-Micro model emission estimates for the composite vehicle are higher, as expected, given that they include high emitting vehicles.
- The VT-Micro model emission trends appear to be consistent with the in-field emission measurements.

Table-6.2: Comparison of TRANSIMS results with the OEM

	Emission	TRANSIMS	OEM
RUN1	NOx (g)	.952	.845
	CO (g)	16	2.1
	HC (g)	1.139	.33
	FE (g)	172.6	149.64
RUN2	NOx (g)	.971	1.12
	CO (g)	17.5	1.9
	HC (g)	1.211	.393
	FE (g)	175.9	145.9
RUN3	NOx (g)	.930	1.21
	CO (g)	16.9	2.03
	HC (g)	1.229	.373
	FE (g)	203.3	135.05
RUN4	NOx (g)	.913	.929
	CO (g)	20.4	1.85
	HC (g)	1.773	.3624
	FE (g)	324.1	132.57
RUN5	NOx (g)	.934	.816
	CO (g)	17.6	1.65
	HC (g)	1.442	.385
	FE (g)	220.1	126.79
RUN6	NOx (g)	.936	1.186
	CO (g)	17.8	1.53
	HC (g)	1.372	.414
	FE (g)	212.1	136.6





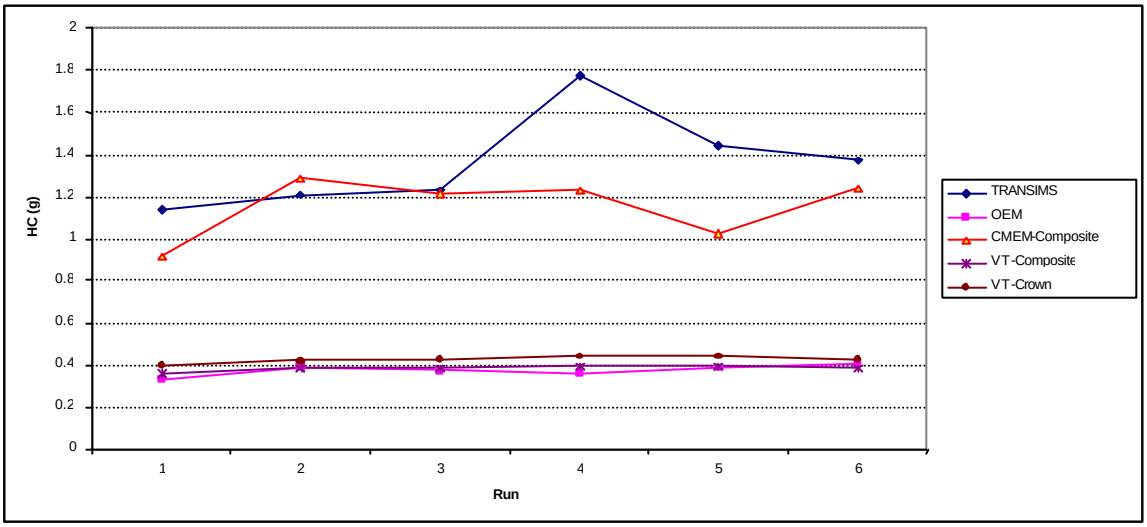
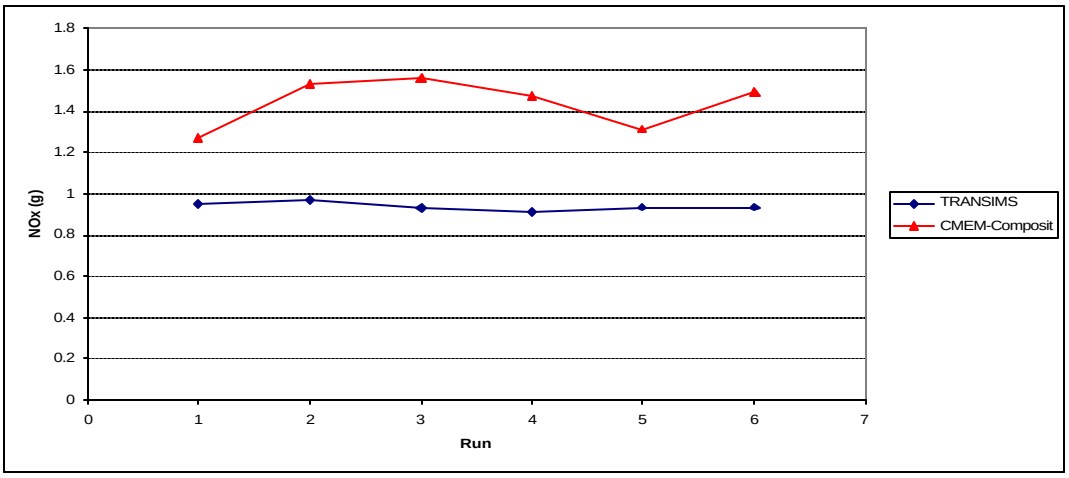
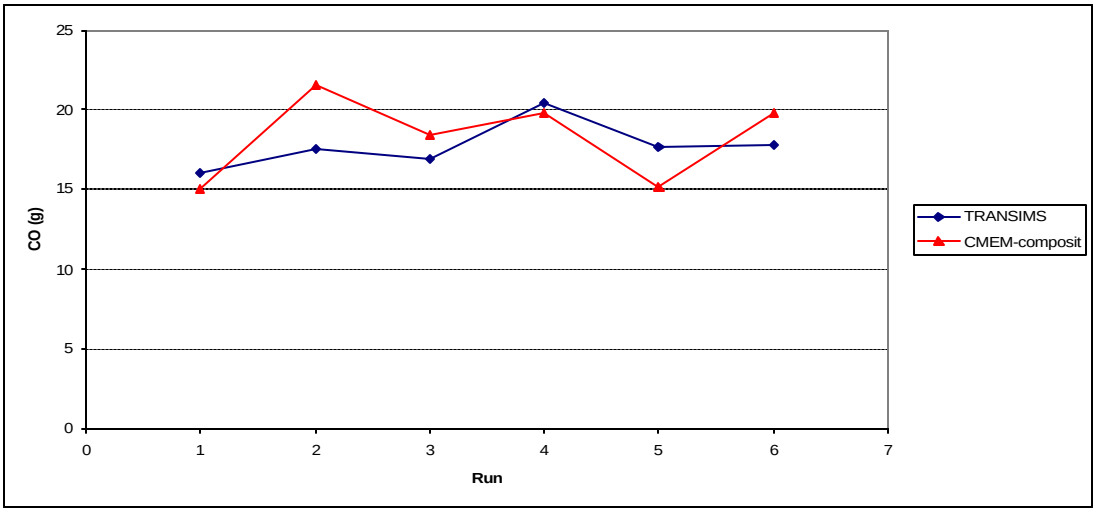


Figure-6.19: The Emission Estimate in TRANSIMS, CMEM, VT-Composite versus the OEM data.





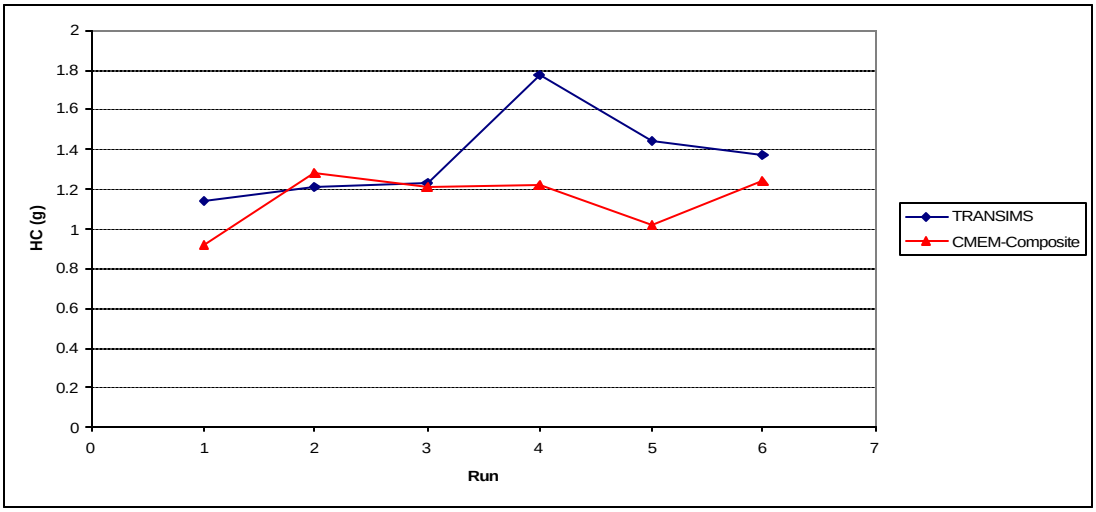
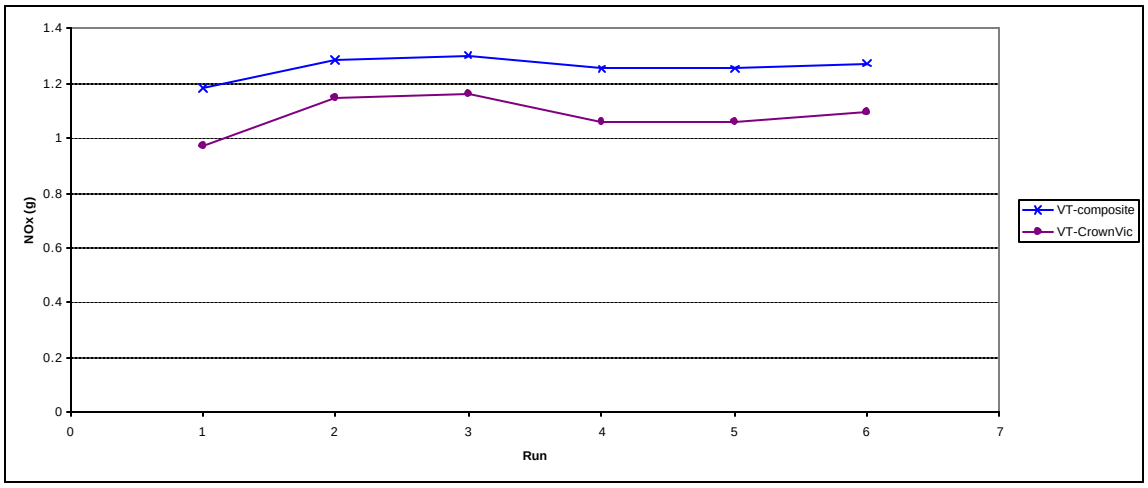
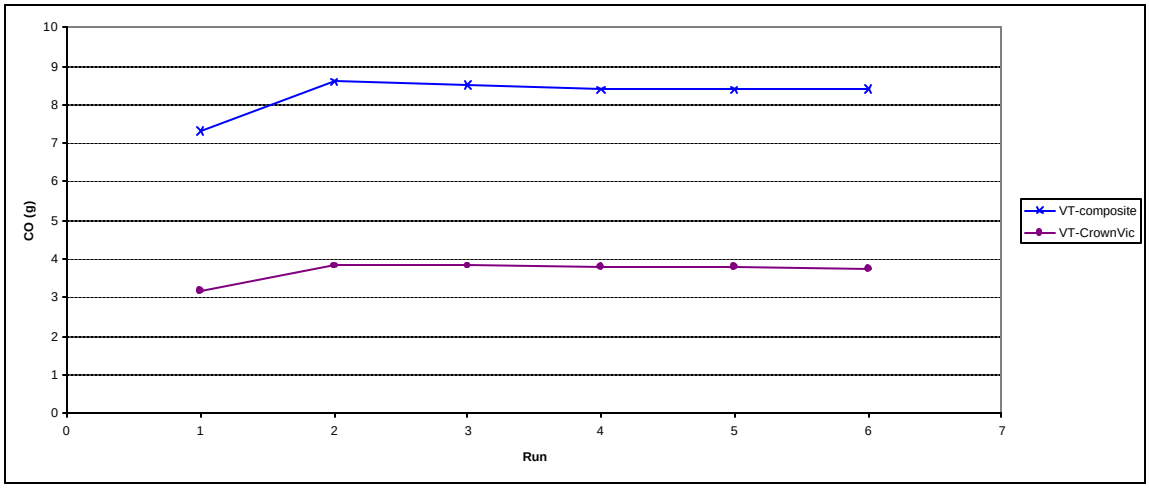


Figure-6.20: The Emission Estimates in TRANSIMS and CMEM.





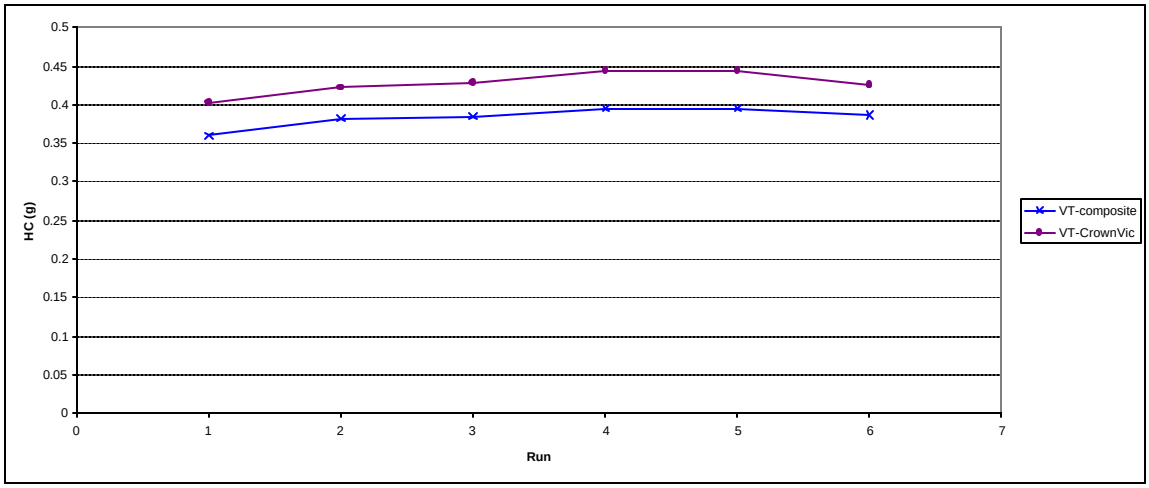
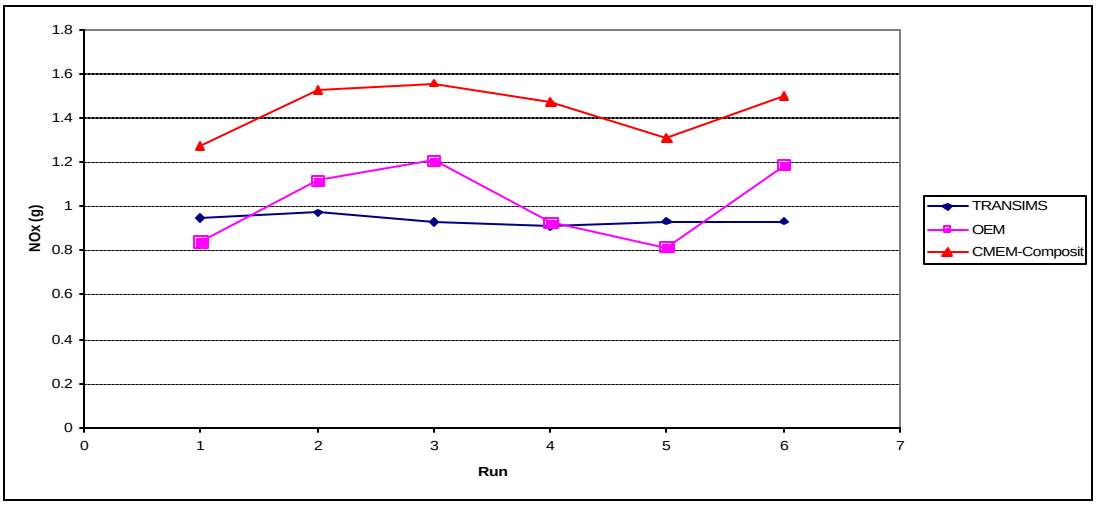
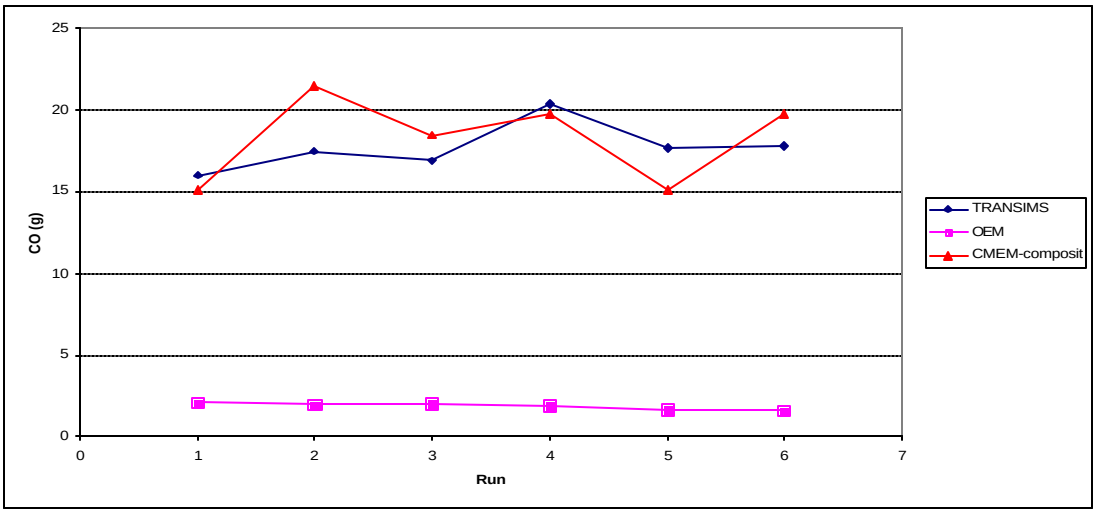


Figure-6.21: The Validation of Single Vehicle Estimation Against Composite Vehicle Estimation.





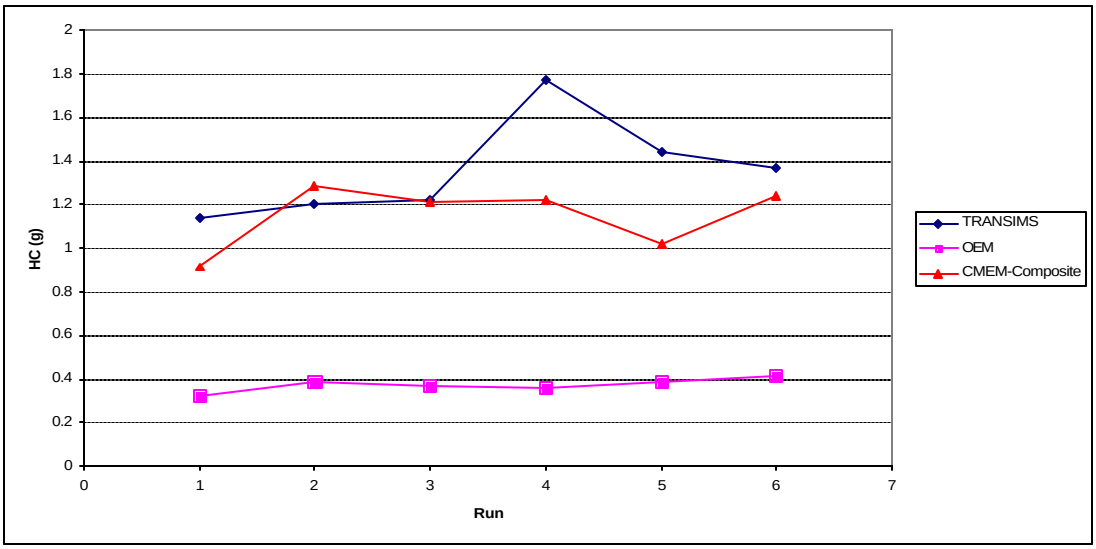


Figure-6.22: The OEM Data versus the Emission Estimates in TRANSIMS and CMEM.

Chapter 7

Summary and Conclusions

This chapter summarizes this dissertation, and presents some concluding comments. Possible extensions and future directions for research in this area are discussed as well.

TRANSIMS is part of the Travel Model Improvement Program (TMIP) sponsored by the U.S. Department of Transportation, the Environmental Protection Agency (EPA), and the Department of Energy. TMIP is a multi-year, multi-agency program designed to improve both analytical tools and the integration of these tools into the transportation planning process. TMIP was created in order to increase the ability of existing travel forecasting procedures to respond to emerging policy and technology issues. Moreover, it also redesigns the travel forecasting process to reflect changes in behavior, responds to greater information needs placed on the forecasting process, and takes advantage of changes in data collection technology. In addition, TMIP integrates the forecasting technique into the decision making process, providing a better understanding of the effects of transportation decisions. TMIP has focused on both short-term and long-term improvements to the models and planning procedures. The short-term improvements concentrated on changes to the existing four-step modeling process. TRANSIMS is the long-term effort to redesign the modeling process from the ground-up.

TRANSIMS is still under development and it needs some improvement to its modules. This dissertation has addressed some of the existing problems in the current version of TRANSIMS and has enhanced the Emissions Estimator module and the interaction between the Route Planner and the Microsimulator modules to alleviate these problems. From an overall perspective, the methodologies developed in this dissertation facilitate the improvement procedure of TRANSIMS. Although this dissertation enhances TRANSIMS, the proposed algorithm to compute dynamic user equilibria is independent of TRANSIMS and can be implemented within any software package. This algorithm addresses the non-convergence issue that exists in dynamic user equilibrium algorithms such as CONTRAM more efficiently.

A heuristic method to compute dynamic user equilibria was presented in this dissertation. This method assigns travelers one by one to the transportation network and updates the link travel times after each assignment. To guarantee finite termination, a suitable stopping criterion was adopted. The proposed algorithm was implemented within

TRANSIMS and was tested on a large-scale network, Bignet, which is part of the transportation city network of Portland, Oregon. It was shown that the algorithm converges rapidly to a near-equilibrium solution. This near-user equilibrium was found after only three iterations between the Route Planner and the Microsimulator, consuming less than 20-33% of the effort required by the current version of TRANSIMS (over 16 iterations) and yet producing an improved distribution of travelers.

One extension to this dissertation could be to perform parallel processing in the Route Planner in the case of very large-scale transportation networks. In this case, we can update the link travel times after routing a batch of travelers rather than afer routing each only one traveler. In this manner, we can process batches of travelers in parallel, updating the travel times between each such run.

Another extension to this work could be to make the Route Planner smarter. Specifically, the Route Planner could be designed to find the most convenient paths for different types of drivers rather than finding the shortest paths. In the current version of the Route Planner, it is assumed that drivers select the shortest path to commute. However, in the real-world, drivers do not necessarily select the shortest path. Some drivers prefer to drive on freeways to avoid intersections and stop signs, while others prefer to use local streets and arterials to avoid exits and entrances to freeways. Drivers could be categorized into different types and the shortest path algorithm could be then run for each class of driver. The label constraint feature in the Route Planner could be exploited to accommodate this extension by providing labels for different choice of transportation links as well as modes.

This dissertation has also described and validated the framework of TRANSIMS for modeling vehicle emissions. It has identified an error in the model calculations and has improved the emission modeling formulation. Emission results from the Emissions Estimator module of TRANSIMS, which were generated based on the CMEM model, were found to follow the CMEM model estimates and were significantly different from the OEM data. This inconsistency in emission estimates is due to inherent problems with the CMEM model and not with the emissions estimation procedures. Therefore, using other emission models within the TRANSIMS framework could lead to better results.

A future extension for this research area would be to use another emission model, such as the VT-Micro model to provide the input emission arrays for TRANSIMS in order to avoid the imprecision of the emissions estimation. This is under current development by the author.

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993), "*Network Flows Theory, Algorithms, and Applications*", Prentice Hall, Englewood Cliffs, New Jersey.

Akcelik, R. (1991), "Travel Time Function for Transport Planning Purposes: Davidson's Function, It's Time Dependent Form and an Alternative Travel Time Function", *Astralian Road Research*, 21(3): 49-59.

Arentz, T. A. and Timmermans, H. J. P. (2004), "A Learning-based Transportation Oriented Simulation System", *Transportation Research B*, 38: 613-633.

Aashtiani, H. Z. and Magnanti, T. L. (1981), "Equilibria on a Congested Transportation Network", *SIAM Journal of Algebraic and Discrete Methods*, 2(3): 213-226.

Aashtiani, H. Z. and Magnanti, T. L. (1981), "Equilibria on a Congested Transportation Network", *SIAM Journal of Algebraic and Discrete Methods*, 2(3): 213-226.

Barrett, C, Jacob, R., and Marathe, M. (2001), "Formal-Language-Constrained Path Problems", *SIAM Journal on Computing*, 30: 809-837.

Barth, M., Younglove, T., Wenzel, T., Scora, G., An, F., Ross M., and Norbeck, J. (1997), "Analysis of Modal Emissions for a Diverse in-use Vehicle Fleet" , *Transportation Research Record*, 1587: 73-84.

Bazaraa, M. S., Sherali, H. D., and Jarvis. J. J. (1990), *"Linear Programming and Network Flows*", *Second Edition*, John Wiley and Sons, New York, NY.

Beckman, R. J., Baggerly, K. A., and McKay M. D., 1996, "Creating Synthetic Baseline, Populations", *Transportation Research A*, 30(6): 415-429.

Ben-Akiva, M., 2001, "Network State Estimation and Prediction for Real-Time Traffic Management", *Networks and Spatial Economics*, 1: 293-318.

Ben-Akiva, M., De Palma, A., and Kanaroglou, P. (1986), "Dynamic Model of Peak Period Traffic Congestion with Elastic Arrival Rates", *Transportation Science*, 20(2): 164-181.

Bhat, Ch. R. (1998), "Analysis of Travel Mode and Departure Time Choice for Urban Shopping Trips", *Transportation Research B*, 32(6): 361-371.

Bonsall, P. and Parry, T (1991), "Using an Interactive Route-Choice Simulator to Investigate Drivers' Compliance with Route Guidance Advice", *Transportation Research Record*, 1306: 59-68.

Boyce, D., Ralevic-Dekic, B., and Bar-Gera H. (2004), "Convergence of Traffic Assignment: How Much Is Enough?" *Journal of Transportation Engineering*, 130(1): 49-55.

Breiman, L., J.H. Friedman, R.A. Olshen and C.J. Stone. (1984), "*Classification and Regression Trees*", Chapman and Hall, London, England.

Canada, J. R., Sullivan, W. G., and White, J. A. (1996), *"Capital Investment Analysis for Engineering and Management", Second Edition*, Prentice Hall, Englewood Cliffs, New Jersey.

Cantarella G. E., and Cascetta, E. (1995), "Dynamic Processes and Equilibrium in Transportation Networks: Towards a Unifying Theory", *Transportation Science*, 29(4): 305-329.

Carey M., Ge, Y. E. (2003), "Comparing Whole-link Travel Time Models", *Transportation Research B*, 37:905-926.

Carey, M. and McCartney, M. (2003), "Pseudo-periodicity in a Travel Time Model Used in Dynamic Traffic Assignment", *Transportation Research B*, 37:769-792.

Clark, L.A., and Pregibon, S. (1990), "Tree-*based models*", In Statistical models in Wadsworth S. and Brooks/Cole, 377-419.

DYNASMART, *www.dynamictrafficassignment.org*.

DynaMIT, *www.dynamictrafficassignment.org*.

Cairns, S., Hass-Klau, C., and Goodwin, Ph. (1998), "Traffic Impact of Highway Capacity Reductions: Assessment of the Evidence", *London Transport Planning*, London; *www.ucl.ac.uk/transport-studies/sc1.htm*.

De Palma, A., Ben-Akiva, M., Lefevre C., and Litinas N. (1983), "Stochastic Equilibrium Model of Peak Period Traffic Congestion", *Transportation Science*, 17(4): 430-453.

Esser J. and Nagel, K. (2001), "Iterative Demand Generation for Transportation Simulation", In Hensher D. King J., editors, *The Leading Edge of Travel Behavior Research*, Pergamon: 659-681.

Ettema, D. F., Borgers, A. W. J., and Timmermans, H. J. P. (1993), "A Simulation Model of Activity Scheduling Behavior", *Transportation Research Record*, 1413: 1-11.

Fisk, C. (1980), "Some Developments in Equilibrium Traffic Assignment", *Transportation Research B*, 14(3): 243-255.

Friesz, T.L., Luque, J., Tobin, R.L., and Wie B. (1989), "Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem", *Operations Research*, 37(6): 893-901.

Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L., and Wie, B. (1993), "A Variational Inequality Formulation of the Dynamic Network User-equilibrium Problem", *Operations Research*, 41(1): 179-191.

Gallo, G. and Pallottina, S. (1988), "Shortest Path Algorithms", *Annals of Operations* Research, 13: 3-79.

Gajowinczek, P., Pioro, M., and Arvidsson, A. (1996), "VP Reconfiguration Through Simulated Allocation", *The Thirteenth Nordic Teletraffic Seminar*, Department of Telematics, NTNU, Trondheim, Norway, 251-260.

Garling, T. (1998), "Behavioral Assumptions Overlooked in Travel-choice Modeling", In Ortuzar, J., Jara-Diaz, S., Hensher, D., editors, Transportation Modeling, Pergamon, Oxford, 1-18.

Gawron, C. (1998), "An Iterative Algorithm to Determine the Dynamic User Equilibrium in a Traffic Simulation Model", *International Journal of Modern Physics C*, 9(3): 393-407.

Glover, F., Klingman, D., and Phillips, N. (1985), "A New Polynomially Bounded Shortest Paths Algorithm", *Operations Research*, 33:65-73.

Hendrickson, C. and Kocur, G. (1981), "Schedule Delay and Departure Time Decisions in a Deterministic Model", *Transportation Science*, 15(1): 62-77.

Hendrickson, C. and Plank, E. (1984), "The Flexibility of Departure Times for Work Trips", *Transportation Research A*, 18: 25-36.

Hensher, D.A., and Button K.J. (2000), "*Handbook of Transportation Modeling*", PERGAMON, Amsterdam, New York, NY.

Hensher, D., Stopher, P. (1977). "*Behavioral Travel Modelling*", Croom Helm London.

Hillier, F. S., and Lieberman, G. L. (1986), "*Introduction to Operations Research*", *Fourth Edition.* Holden-Day, Oakland, CA.

Hobeika, A. G., Sherali, H. D., Jeenanunta, C., and Jeihani M. (2003), "Transportation Analysis and Simulation System (TRANSIMS): Short Course Instruction Manual",

Virginia Polytechnic Institute and State University, Blacksburg, VA, submitted to FHWA.

Horwitz, J. L.(1993), "Semiparametric Estimation of a Work-Trip Mode Choice Model", *Journal of Econometrics*, 58:49-70.

Jacob, R., Marathe, M. V., and Nagel, K. (1998), "A computational study of routing algorithms for realistic transportation networks", *Los Alamos National Laboratory*, LA-UR 98-2249, Los Alamos, NM.

Jou, R. Ch. and Mahmassani, H.S. (1996), "Comparability and Transferability of Commuter Behavior Characteristics between Cities: Departure Time and Route-Switching Decisions", *Transportation Research Record*, 1556: 119-130.

Kaufmann, D. E., Smith, R. L., and Wunderlich K. E. (1998), "User Equilibrium Properties of Fixed Points in Dynamic Traffic Assignment", *Transportation Research C*, 6: 1-16.

Kelly, T. and Nagel, K. (1998), "Relaxation Criteria for Iterated Traffic Simulations", *Los Alamos National Laboratory*, LA-UR 97-4453, Los Alamos, NM.

Kitamura, R., Fujii, S., 1998, "Two Computational Process Models of Activity-travel Behavior", In Garling, T., Laitila, T., Westin, K., editors, Theoretical Foundation of Travel Choice Modeling", Elsevier, Amesterdam, The Netherlands, 251-280.

Laborczi and Nowtony (2003), "A Heuristic Method to Determine the Dynamic User Equilibrium", *arsenal research*, *Transportation Technologies*, Vienna, Austria.

Leonard, D. R., Tough, J. B., and Baguley, P. C. (1978), "CONTRAM: A Traffic Assignment Model for Predicting Flows and Queues During Peak Periods", *Transportation and ROAD Research Laboratory*, TRRL Laboratory Report 841, Crowthorne.

Lin, W. and Lo, H. K. (2000), "Are the Objective and Solutions of Dynamic User-Equilibrium Models Always Consistent?", *Transportation Research A*, 34: 137-144.

Li, J., Fujiwara, O., Kawakami S. (2000), "A Reactive Dynamic User Equilibrium Model in Network with Queues", *Transportation Research B*, 34: 605-624.

Lock, J. B. and Gelling, M.J. (1976), "The Tassman Bridge Disaster", *Australian Road Research*, 6(2): 9-16.

Mahmassani, H.S (1990), "Dynamic Models of Commuter Behavior: Experimental Investigation and Application to the Analysis of Planned Traffic Disruption", *Transportation Research A*, 24(6): 465-484.

Mahmassani, H. M. (2001), "Dynamic Traffic Assignment and Simulation Methodology for Advanced System Management Applications", *Networks and Spatial Economics*, 1: 267-292.

Mahmassani, H.S (2002), *"In Perpetual Motion: Travel Behavior Research Opportunities and Application Challenges*", PERGAMON, Kidlington, Oxford, UK.

Mahmassani, H.S. and Jayakrishnan, R. (1991), "System Performance and User Response under Real-time Information in a Congested Traffic Corridor", *Transportation Research A*, 25(5): 293-307.

Mahmassani, H. S. and Herman, R. (1984), "Dynamic User Equilibrium Departure Time and Route Choice on Idealized Traffic Arterial", *Transportation Science*, 18(4): 362-384.

Mahmassani, H.S. and Liu, Y. (1999), "Dynamics of Commuting Decision Behavior under Advanced Traveler Information System", *Transportation Research B*, 7: 91-107.

Mannering, F. L., Abu-Eisheh, S., and Arnadottir, A. T. (1990), "Dynamic Traffic Equilibrium with Discrete/Continuous Econometric Models", *Transportation Science*, 24(2): 105-116.

Nagel, K. (1994), "Fast Low Fidelity Microsimulation of Vehicle Traffic on Supercomputers", *Transportation Research Board Annual Meeting*, Washington D.C.

Nagel, K. and Axhausen, K. (2001), "Microsimulation", In Hensher D. King J., editors, *The Leading Edge of Travel Behavior Research*, Pergamon, Oxford: 209-216.

Nagel, K., Barrett C. L. (1997), "Using Microsimulation Feedback for trip adaptation for realistic traffic in Dallas", *Los Alamos National Laboratory*, LA-UR 97-1334, Los Alamos, NM.

Nagel, K., Beckman, R.J., and Barrett, C. L. (1998a), "TRANSIMS for transportation planning", *InterJournal Complex Systems*, 244, http://interjournal.org.

Nagel, K., Jacob, R., and Madhav, M. (1998), "A Computational Study of Routing Algorithms for Realistic Transportation Networks", *TRANSIMS Report Series*, Los Alamos, NM.

Nagel, K., Rickert, M., Simon, P.M., and Pieck, M. (1998b), "The dynamics of iterated transportation simulation", *Los Alamos National Laboratory*, LA-UR 98-2168, Los Alamos, NM.

Nagel, K. and Rickert, M. (2001), "Parallel Implementation of the TRANSIMS Microsimulation", Parallel Computing, 27(12): 1611-1639.

Nagel, K., Wolf, D. E., Wagner, P., and Simon, P. E. (1998c), "Two-lane Traffic Rules for Cellular Automata: A Systematic Approach", *Physical Review E*, 58(2): 1425-1437.

Nagel, K., Rickert, M., Simon, P.M., and Pieck, M. (2000), "The dynamics of iterated transportation simulation", *Los Alamos National Laboratory*, LA-UR 98-2168, Los Alamos, NM.

Nagel, K. and Schleicher, A. (1994), "Microscopic Traffic Modeling on Parallel High Performance Computers", *Parallel Computing*, 20: 125-146.

Nguyen, S. (1974), "An Algorithm for the Traffic Assignment Problem", *Transportation Science*, 8(3): 203-216.

Noland, R. B. and Small, A. K. (1995), "Travel-Time Uncertainty, Departure Time Choice, and the Cost of Morning Commuters", *Transportation Research Record*, 1493: 150-158.

Oppenheim, N. (1995) "*Urban Travel Demand Modeling: from individual choice to general equilibrium*", John Wiley and Sons, New York, NY.

Pas, E. I., 1985, "State of the Art and Opportunities in Travel Demand: Another perspective", *Transportation Research A*, 19(5/6): 460-464.

Peeta, S. and Mahmassani, H.S. (1995), "System Optimal and User Equilibrium Time-dependent Traffic Assignment in Congested Networks", Annals *of Operations Research*, 60:81-113.

Peeta, S. and Ziliaskopoulos, A. (2001), "Foundations of Dynamic Traffic Assignment: The Past, the Present, and the Future", *Networks and Spatial Economics*, 1: 233-265.

Portland Study Report, (2001), "Postprocessing for Environmental Analysis", *Los Alamos National Laboratory*, LA-UR-01-5715, Los Alamos, NM.

Rakha, H., Ahn, K., El-Shawarby, I., and Jang S. (2004), "Emission Model Development Using In-vehicle On-Road Emission Measurement", *Environmental Quality and Analysis*. CD-ROM. Transportation Research Board, National Research Council, Washington, D.C., 1-12.

Ran, B., Rouphail N.M., Tarko, A., and Boyce D.E. (1997), "Toward a Class of Link Time Functions for Dynamic Assignment Models on Signalized Networks", *Transportation Research B*, 31:277-290.

Ran, B., Boyce, D.E., and LeBlanc, L.J. (1993), "A New Class of Instantaneous Dynamic User Optimal Traffic Assignment Models", *Operations Research*, 41(1):192-202.

Ran, B. and Boyce, D.E. (1996), "*Modeling Dynamic Transportation Networks*", Springer, Berlin, Heidelberg, Second Revised Edition.

Ran, B., Hall, R. W., and Boyce, D.E. (1996), "A Link-Based Variational Inequality Formulation of Ideal Dynamic User-optimal Route Choice Problem", *Transportation Research C*, 4(1): 1-12.

Raney, B. and Nagel, K. (2002), " Iterative Route Planning for Modular Transportation Simulation", *Department of Computer Science, ETH, Zurich, Switzerland*, *www.inf.ethz.ch/personal/nagel/papers*.

Sandholm, W. H. (2002), "A Local Condition Implying Optimality of User Equilibrium", University of Wisconsin, *www.wise.edu*.

Sheffi, Y. (1985), "*Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*", Prentice Hall, Englewood Cliffs, NJ.

Shen, P., Chen, T., and Mahmassani, H.S. (1993), "Dynamic Interactive Simulator for Studying Commuter Behavior under Real-Time Traffic Information Supply Strategies", *Transportation Research Record*, 1413: 12-21.

Sherali, H. D. (1991), "On the Equivalence between Some Shortest Path Algorithms", *Operations Research Letters*, 10: 61-65.

Sherali, H. D., Ozbay, K., and Subramanian, S. (1998), "The Time-Dependent Shortest Pair of Disjoint Paths Problem: Complexity, Models, and Algorithms", *Networks*, 31: 259-272.

Sherali, H. D., Hobeika, A. G., and Kangwalklai, S. (2003), "Time-Dependent Label-Constrained Shortest Path Problems with Applications to TRANSIMS: A Transportation Planning Model, *Transportation Science*, 37: 278-293.

Sherali, H. D., Jeenanunta, C., and Hobeika, A. G. (2004), "The approach-dependent, Time-dependent, Label-constrained, Shortest Path Problem with Application to TRANSIMS: a Transportation Planning System", *Virginia Polytechnic Institute and State University*, Blacksburg, VA.

Simon, P. M., and Nagel K. (1998), "A Simplified Cellular Automaton Model for City Traffic", *Physical Review E*, 58(2): 1286-1295.

Small K.A. (1982), "The Scheduling of Consumer Activities: Work Trips", *American Economic Review*, 72(3): 647-679.

Smith, M. (1984), "The Existence of a Time-Dependent Equilibrium Distribution of Arrivals at a Single Bottleneck", *Transportation Science*, 18(4): 385-393.

Subramanian, S. (1997), "Routing Algorithms for Dynamic, Intelligent Transportation Networks", *Master's Thesis*, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Supplement TRANSIMS Short Course Documents, (2002), *Virginia Polytechnic Institute and State University*, Blacksburg, VA.

Taylor, N. B., 2003, "The CONTRAM Dynamic Traffic Assignment Model", *Networks and Spatial Economics* 3, 297-322.

TRANSIMS_LANL website, (2004), *http://transims.tsasa.lanl.gov*.

TRANSIMS_DOT website, (2004), *http://www.transims.net*.

Vaughn, K.M., Speckman, P. and Sun, D. (1999), "Identifying Relevant Socio-Demographics for Distinguishing Household Activity-Travel Patterns: A Multivariate CART Approach", *National Institute of Statistical Sciences*, NC.

Wagner, P. and Nagel, K. (1999), "Microscopic Modeling of Travel Demand: Approaching the Home-to-work Problem", *Transportation Research Board Annual Meeting*, Paper 990910, Washington D.C.

Watson, P. L. (1973), "Predictions of intercity modal choice from disaggregate, behavioral stochastic models", Highway Research Record, 446: 28-35.

Watson, P.L. (1974), "Comparison of the model structure and predictive power of aggregate and disaggregate models of intercity mode choice", *Transportation Research Record*, 528: 59-65.

Wie, B.(1989), " An Application of Optimal Control Theory to Dynamic User Equilibrium Traffic Assignment", *Transportation Research Record*, 1251: 66-73.

Wie, B., Tobin, R. L., and Friesz, T. L. (1994), "The Augmented Lagrangian Method for Solving Dynamic Network Traffic Assignment Models in Discrete Time", *Transportation Science*, 28(3): 204-220.

Wie, B., Tobin, R. L., Friesz, T. L., and Bernstein, D. (1995), "A Discrete Time, Nested Cost Operator Approach to the Dynamic Network User Equilibrium Problem", *Transportation Science*, 29(1): 79-92.

Wie, B., Tobin, R. L., and Carey, M. (2002), "The Existence, Uniqueness and Computation of an Arc-Based Dynamic Network User Equilibrium Formulation", *Transportation Research B*, 36: 897-918.

Williams, M. D., Thayer, G., Smith, L. L., and Barth, M. J. (2000), "Recent Developments in the TRANSIMS Approach to Emissions Estimation", *TRANSIMS Documents, Los Alamos National Laboratory*, LA-UR-00-388, Los Alamos, NM.

Williams, M. D., Thayer, G., AND Smith, L. L. (2000), "The TRANSIMS Approach to Emissions Estimation", *TRANSIMS Documents, Los Alamos National Laboratory*, LA-UR-99-471, Los Alamos, NM.

Zhang, X. and Rice, J. A., "Short-term Travel Time Prediction", *Transportation Research C*, 11: 187-210.

Vitae

Mansoureh Jeihani was born in Tehran, Iran on June 16, 1973. After completing Molla-Sadra high school in Kerman, she was accepted in the Computer Engineering program of Iran National (Shahid Beheshti) University in 1990 via a very competitive entrance exam. Her rank was 113, in the national-wide university entrance competition among almost 300,000 competitors. After graduating with her bachelor degree, she joined the Iranian National TV as a project director to design the first Persian digital counter in 1995.

She entered graduate school in the Institute for Research in Planning and Development (IRPD) to obtain a master's degree in Socio-economic Systems Engineering (Industrial Engineering) program. She worked as a research assistant in IRPD. In 1998, after graduating, she worked as an instructor in Azad University, Kerman.

She enrolled in the Economics Department at Virginia Polytechnic Institute and State University (Virginia Tech) for her second master's degree in August 1999. She worked as a research assistant and a teaching assistant in the Economics Department. She taught *the Principal of Macroeconomics* course in Virginia Tech in summer 2001 as an instructor.

In January 2002, she entered the Department of Civil and Environmental Engineering at Virginia Tech to pursue her Ph.D. degree in Transportation Engineering. She worked as a research assistant on the TRANSIMS project sponsored by the Federal Highway Administration (FHWA).