INTEGRATING ADAPTIVE QUEUE-RESPONSIVE TRAFFIC SIGNAL CONTROL WITH DYNAMIC TRAFFIC ASSIGNMENT

BY

LEE-FANG CHOW

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2003

ACKNOWLEDGMENTS

I wish to express my gratitude to my advisor, Dr. Gary Long, for his guidance and financial support over the course of this research. Working for him is an invaluable experience for me as a graduate student at this institution.

I would like to thank Professor Kenneth G. Courage, Dr. Bon A. Dewitt, Dr. Sherman X. Bai and Dr. Myron N. Chang for their constructive comments and recommendations.

Finally, very special thanks go to my husband, Min-Tang Li, my son, Andrew, and my parents for their love and support through my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

INTEGRATING ADAPTIVE QUEUE-RESPONSIVE TRAFFIC
SIGNAL CONTROL WITH DYNAMIC TRAFFIC ASSIGNMENT

by

Lee-Fang Chow

August 2003

Chairman: Gary Long
Major Department: Civil and Coastal Engineering

Recently, with their promise for Intelligent Transportation Systems (ITS), dynamic traffic assignment (DTA) and adaptive traffic-responsive signal control have received greater attention. However, the mathematical models used to describe the interaction between these two systems are tenuous and considerable effort is still needed to improve the limitations described in the existing literature, especially regarding the solution capabilities, the oversaturated-queue phenomena, efficacious estimates of link travel disutilities, and a reliable evaluation method. The objectives of this research are to implement non-linear programming techniques to model dynamic traffic assignment and adaptive queue-responsive traffic control separately, and develop an iterative procedure to solve the components of traffic assignment and signal control by minimizing the overall system-wide signal delay.

For traffic signal control, total intersection delays in a network are minimized by allocating appropriate time splits. The generalized intersection delay model in the Highway Capacity Manual was used to take the oversaturated queue problem into account.

Improvements to DTA methods include formulating the dynamic user-optimal route choice model as a variational inequality (VI) model, calculating the expected travel time on each link using a simulation model, applying a relaxation algorithm to produce an equivalent optimization formulation of the VI model, and developing a solution algorithm which can be implemented using existing traffic software.

There is not much hope for developing exact solution algorithms to solve these two models simultaneously because of the computational complexity of the non-linear programs. Therefore, a heuristic procedure involving an iterative optimization assignment is used to solve the combined models.

A computerized procedure was developed to implement the solution procedures and a numerical example of a traffic network was prepared to test the program. An accepted traffic simulation model, CORSIM 5.1, was used to validate the results for both static and dynamic optimizations. The test results showed that dynamic traffic assignment with adaptive traffic-responsive signal settings reduced the network-wide delays by nearly 15%.

CHAPTER 1
INTRODUCTION

1.1 Background

Traffic congestion, especially during peak travel periods, is well-known to travelers in most urban areas. Most travelers evaluate the degree of congestion by the amount of time in which they are delayed. Consequently, minimizing delay is important to transportation engineers for setting the timing controls on traffic signals.

Delay is related to both traffic volumes and the capacities of roadway links. Traffic volumes fluctuate with different levels of traffic demand during a typical day, and also with diversions of traffic to alternative paths as motorists strive to find the fastest paths between their origins and destinations to minimize their delays. When traffic congestion becomes intense with long queues of traffic waiting at traffic signals such that portions of the queues must wait for the next signal cycle, known as oversaturated queues, minimizing delay becomes more complicated.

In the traditional traffic management system, link capacities and traffic flow volumes are considered to remain fixed during a whole analysis period. Therefore, the standard traffic management problem is to optimize traffic network performance with given fixed demands for travel and a fixed supply of transportation facilities.

Currently, Intelligent Transportation Systems (ITS) are being designed and developed to improve the efficiency of existing traffic network performance. Computer and

electronic communication technologies are exploited to provide real-time information and route guidance to motorists through Advance Traveler Information Systems (ATIS). Real-time information and route guidance are produced by Dynamic Traffic Assignment (DTA) systems based on real-time traffic data, historical databases and dynamic traffic-responsive signal control strategies. Therefore, the interaction in the Advanced Traffic Management Systems (ATMS) will become dynamic because the traffic flow on each link and control actions are time-dependent.

1.2 Problem Statement

To date, the mathematical models used to describe the dynamic interactions in transportation systems are still tenuous. Two simulation-based, real-time computer systems have been developed under the DTA Project sponsored by the Federal Highway Administration (FHWA) with Oak Ridge National Laboratories (ORNL) as the program manager [1]:

1. DynaMIT is the result of approximately 10 years of research and development at the Intelligent Transportation Systems Program of the Massachusetts Institute of Technology (MIT).

2. DYNASMART is the outgrowth of several years of research and development at the University of Texas, with the participation of researchers currently at the University of Maryland, Purdue University, and Northwestern University.

These DTA programs provide real-time computer systems for traffic estimation, prediction, and generation of traveler information and route guidance. They support the operation of ATIS and ATMS at Traffic Management Centers (TMC). However, they are simulation-based dynamic traffic assignment systems with micro-simulation of individual user decisions in response to traveler information and a macroscopic traffic flow simulation approach [2]. Although the analytical approach which formulates the entire system as a mathematical

model has difficulties in formulating and solving the program, the solutions obtained can be proven to be optimal and satisfy the equilibrium conditions (i.e., the dynamic generalization of Wardrop's principle of user-optimal network equilibrium [3]).

Most current research addresses the development of DTA systems [4, 5] and real-time traffic-responsive control systems [6] independently. Allsop [7] was the first to address the interaction between traffic control and traffic assignment, followed by Smith [8], Sheffi and Powell [9], Smith and Vuren [10], and Yang and Yagar [11]. However, all of them only considered the static traffic situation. Some of the current development work utilizes frameworks for the integration of DTA and traffic control systems into a combined model. Chen and Hsueh [12] proposed a combined model for a discrete-time dynamic traffic-responsive signal control system. A heuristic approach involving an iterative optimization and assignment (IOA) method was applied to solve a numerical example involving a small network. However, the traffic queue phenomenom was not considered in the traffic control model nor was a good estimate of travel disutility considered in the assignment model. Gartner and Stamatiadis [13] presented only a framework to integrate dynamic traffic assignment with real-time traffic adaptive control without any mathematical model or formulation. Chen and Ben-Akiva [14] formulated the combined dynamic traffic control-assignment system as a non-cooperative game between a traffic authority and users but did not give any algorithm to solve it.

1.3 Goal and Objectives

The goal of this dissertation research is to develop a model to integrate the effects of queue-responsive signal timings on discrete-time traffic flow patterns in dynamic traffic assignment systems. The specific objectives of this research are the following:

1. To review the state-of-the-art in the areas of dynamic traffic assignment and dynamic traffic-responsive signal control.

2. To develop a combined model for dynamic traffic assignment and adaptive traffic control. In the traffic control part, the vehicle delays in the network are represented by the generalized delay model for signalized intersections in the *Highway Capacity Manual* [15]. In the traffic assignment part, good estimates of link travel times are considered for motorists to find dynamic user-optimal paths over the network for given origin-destination (O-D) demands and signal timings from the traffic signal control model.

3. To propose an algorithm to solve this combined model.

4. To demonstrate how to program and apply existing traffic software to implement the solution algorithms and test by a numerical example of a traffic network.

5. To demonstrate the results of the combined model by simulation.

1.4 Scope and Limitations

A state-of-the-art algorithm is devloped in this study to optimize total intersection delays in a given traffic network by considering the dynamic interaction effect between the traffic control system and traffic assignment model. The dynamic interaction is considered by adding discrete time periods as an additional dimension in the model structures and solution algorithm. Delays arising from oversaturated queues are included in optimizing signal settings and traffic flows. In this study, both of the cycle length and signal phase plan at each intersection in a given traffic network are presumed fixed and are not altered during the optimization process. In addition, the travel demand between each traffic analysis zone pair is assumed fixed during the analysis period. To test the model, microscopic simulation was used as a surrogate for field data collection to enable controllabilty of input information and the due to limitations on resources available to the researcher. CORSIM 5.1 was chosen to perform the simulation because of its availability to the researcher and because of its widespread acceptance and utilization in the USA.

1.5 Dissertation Organization

In Chapter 1, an introduction to the research topic and the needs for the research are presented. Improvements needed in previous research are identified and the specific objectives of the research are stated.

A literature review on transportation management systems is presented in Chapter 2. The static transportation management system is first reviewed, followed by the dynamic transportation management system. An evaluation tool, the simulation software, and the generalized delay model which improves the oversaturated queue problem in dynamic traffic control are also reviewed.

Chapter 3 presents an improved optimization model for the dynamic traffic control system. A generalized intersection delay model is used to take the oversaturated queue problem into account. A strategy to improve the situation where the optimization program with a non-convex objective function may not reach the global minimum is developed. This chapter ends with an application of the Frank-Wolfe method to solve the optimization program.

Chapter 4 deals with the dynamic traffic assignment system. A dynamic user-optimal route choice model is formulated as a variational inequality (VI) model. A simulation model, CORSIM 5.1, is used to estimate the travel time on each link. A relaxation algorithm is applied to produce an equivalent optimization formulation of the VI model when the travel time on each link is known. A solution algorithm which can be implemented by existing traffic software is developed.

Chapter 5 presents strategies for combining the real-time traffic-responsive signal control model and the dynamic traffic assignment model. In Chapter 6, a computer program

that was prepared to implement all the solution algorithms developed for dynamic transportation management is introduced. Several existing traffic software systems are built into the program to simplify the input processes for the transportation network, the O-D demand and the traffic control actions. A numerical example is used to test the program.

Chapter 7 presents the comparison of the traffic performance between static and dynamic transportation management systems. Chapter 8 presents a summary and conclusions and recommends areas for further research.

CHAPTER 2
REVIEW OF STATIC AND DYNAMIC TRANSPORTATION MANAGEMENT SYSTEMS

2.1 Introduction

The performance of a transportation system is the result of a set of complex interactions between traffic operational controls and the demand for transportation services. This is the basis for the development of traffic control and assignment models for the design and planning of transportation management systems.

In this chapter, the static transportation management system, which includes static traffic control, static traffic assignment and a "balancing" process to integrate these two systems together, is first reviewed, followed by recent research which extends these concepts to the dynamic case. Greater details are given for the algorithms more directly related to those needed in adaptive queue-responsive traffic signal control with dynamic traffic assignment as developed in this research.

2.2 Static Traffic Control Systems

Traffic control systems deal with how to determine the optimal traffic signal timings to reduce fuel consumption, minimize delay and also improve safety. TRANSYT-7F [16] is the best-known software for determining the static network-wide optimal signal timing settings. With fixed-flow input, TRANSYT-7F uses a macroscopic, deterministic simulation and optimization procedure to find the best solution. The optimization procedure can be briefly described as follows:

Step 1: An initial signal timing plan is simulated by the traffic model and an initial Performance Index (PI) is calculated.

Step 2: The signal timing setting is changed by a specified amount and the resulting traffic flow is re-simulated. A new PI is then calculated.

Step 3: The new PI is compared with the previous value. If no further improvement can be made by varying the signal setting, the procedure will stop. Otherwise, go to Step 2.

Although TRANSYT-7F is a powerful tool to determine static optimization, it is not feasible to enhance it to the dynamic case because it is too complicated to modify the macroscopic simulation procedure with time-dependent traffic flows.

2.3 Static Traffic Assignment

The traffic assignment system can be stated as finding the link flows when given the O-D traffic volumes, the network, and the link performance functions. To solve the problem, a rule of how motorists choose their routes has to be specified. The equilibrium traffic flows are determined according to the rule and link performance functions. Three definitions of equilibrium are user equilibrium, system optimization, and stochastic user equilibrium.

2.3.1 User Equilibrium

User-equilibrium (UE) assumes that each motorist has full information about link performance relationships and he or she will choose the route with minimum travel impedance. The travel impedance can include many components. However, travel time is often used as the sole measure of link impedance [3]. For each origin-destination zone-pair, at user equilibrium, the travel disutilities on all used paths are equal and no traveler can improve his or her travel time by unilaterally changing routes [3]. Note that a network is defined mathematically as a set of nodes and a set of links connecting these nodes. A path is a sequence of directed links leading from one node to another.

In the mid-1950s, a formulation representing the UE condition was developed and is known as Beckmann's transformation [17]. The formulation can be expressed as follows:

$$\min\ z(\mathbf{x}) = \sum_{m} \int_{0}^{x_m} d_m(\omega)d\omega \tag{1}$$

subject to

$$\sum_k f^{rs}_k = q_{rs} \qquad \forall\ r, s \tag{2}$$

$$f^{rs}_k \geq 0 \qquad \forall\ k, r, s \tag{3}$$

with the incidence relationship

$$x_m = \sum_{rs} \sum_k f^{rs}_k \delta^{rs}_{m,k} \qquad \forall\ m \tag{4}$$

where

x_m = traffic volume assigned to link m

q_{rs} = total traffic volume interchanging from origin r to destination s

f^{rs}_k = traffic volume flowing along path k connecting origin r to destination s

$d_m(x_m)$ = travel disutility on link m related to traffic volume, x_m, on link m

$\delta^{rs}_{m,k}$ = 1, if link m is on path k between origin-destination pair r-s
= 0, otherwise

This program does not have any intuitive economic or behavioral interpretation. It is only a mathematical model that is utilized to solve the equilibrium problem. To prove the equivalence between the solution of Beckmann's transformation and the UE condition, the method of Lagrange multipliers, which involves an auxiliary function known as the Lagrangian, is applied as follows.

Given the incidence relationship, $x_m = x_m(\mathbf{f})$ in Eq. 4, the Lagrangian whose stationary point coincides with the minimum of the constrained optimization in Eq. 1 to Eq. 4 can be formulated as:

$$L(\mathbf{f}, \mathbf{y}) = z[\mathbf{x}(\mathbf{f})] + \sum_{rs} y_{rs}(q_{rs} - \sum_k f^{rs}_k) \tag{5}$$

where y_{rs} denotes the dual variable associated with the flow conservation constraint for O-D pair r-s in Eq. 2. The stationary point of the unconstrained Lagrangian can be found by solving for the root of the gradient, $\nabla_{\mathbf{f}} L(\mathbf{f}, \mathbf{y}) = \mathbf{0}$ and $\nabla_{\mathbf{y}} L(\mathbf{f}, \mathbf{y}) = \mathbf{0}$. $\nabla_{\mathbf{y}} L(\mathbf{f}, \mathbf{y}) = \mathbf{0}$ simply states the flow constraints. However, given the nonnegativity constraints of $\mathbf{f} \geq \mathbf{0}$ in Eq. 3, the following two conditions, associated with $\nabla_{\mathbf{f}} L(\mathbf{f}, \mathbf{y}) = \mathbf{0}$, have to hold because the stationary point of $L(\mathbf{f}, \mathbf{y})$ can occur either for positive $\mathbf{f}$ or it can be on the boundary of the feasible region where some $f^{rs}_k = 0$. Obviously, the condition $\mathbf{f} \geq \mathbf{0}$ has to hold as well.

$$f^{rs}_k \frac{\partial L(\mathbf{f}, \mathbf{y})}{\partial f^{rs}_k} = 0 \quad ; \quad \frac{\partial L(\mathbf{f}, \mathbf{y})}{\partial f^{rs}_k} \geq 0 \qquad \forall\ k, r, s \tag{6}$$

The above first-order condition can be obtained by the following calculation:

$$\frac{\partial}{\partial f^{rs}_k} L(\mathbf{f}, \mathbf{y}) = \frac{\partial}{\partial f^{rs}_k} z[\mathbf{x}(\mathbf{f})] + \frac{\partial}{\partial f^{rs}_k} \sum_{rs} y_{rs}(q_{rs} - \sum_k f^{rs}_k) \tag{7}$$

$$\frac{\partial z[\mathbf{x}(\mathbf{f})]}{\partial f^{rs}_k} = \sum_m \frac{\partial z(\mathbf{x})}{\partial x_m} \frac{\partial x_m}{\partial f^{rs}_k}$$

$$= \sum_m \left(\frac{\partial}{\partial x_m} \sum_m \int_0^{x_m} d_m(\omega) d\omega\right) \delta^{rs}_{m,k} = \sum_m d_m \delta^{rs}_{m,k} = u^{rs}_k \tag{8}$$

$$\frac{\partial}{\partial f^{rs}_k} \sum_{rs} y_{rs}(q_{rs} - \sum_k f^{rs}_k) = \sum_{rs} y_{rs}\left(-\sum_k \frac{\partial f_k}{\partial f^{rs}_k}\right) = -y_{rs} \tag{9}$$

where u^{rs}_k = travel disutility along path k connecting origin r to destination s.

The general first-order conditions for the minimization program in Eqs. 1 to 4 can be expressed as

$$f_k^{rs}(u_k^{rs} - y_{rs}) = 0 \qquad \text{Œ } k, r, s \tag{10}$$

$$u_k^{rs} - y_{rs} \$ 0 \qquad \text{Œ } k, r, s \tag{11}$$

$$' {}_k f^{rs}{}_k = q_{rs} \qquad \text{Œ } r, s \tag{12}$$

$$f^{rs}{}_k \$ 0 \qquad \text{Œ } k, r, s \tag{13}$$

From Eq. 11, y_{rs} is less than or equal to u_k^{rs}, the travel disutility along path k connecting origin r to destination s. Therefore, y_{rs} equals the minimum path travel disutility between origin r and destination s. Eq. 10 holds at the point that minimizes the objective function for either $f_k^{rs} = 0$ and $(u_k^{rs} - y_{rs}) > 0$, or $f_k^{rs} > 0$ and $(u_k^{rs} - y_{rs}) = 0$. That means if the travel disutility, u_k^{rs}, on path k is greater than y_{rs}, the minimum path travel disutility, the flow on this path is zero; or if $u_k^{rs} = y_{rs}$, the flow f_k^{rs} is positive. This simply interprets the principle of UE.

For the solution of the UE program to be unique, it is sufficient only to prove that the objective function is convex with respect to link flow, since the convexity of the feasible region is assured for linear equality and nonnegative constraints. To prove that z(**x**) is convex, the matrix of the second derivatives of z(**x**) with respect to **x** (the Hessian) has to be positive definite:

$$\text{since} \quad \frac{\partial^2 z(x)}{\partial x_m \partial x_n} = \frac{\partial d_m(x_m)}{\partial x_n} = \begin{cases} \dfrac{dd_n(x_n)}{dx_n} & \text{for } m = n \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where m and n are both link indexes. When m is not equal to n, the second derivatives would be zero.

$$\text{Therefore,} \quad \nabla^2 z(x) = \begin{bmatrix} \frac{dd_1(x_1)}{dx_1} & 0 & 0 & \cdots \\ 0 & \frac{dd_2(x_2)}{dx_2} & 0 & \cdots \\ 0 & 0 & \ddots & \\ \vdots & \vdots & & \frac{dd_A(x_A)}{dx_A} \end{bmatrix} \tag{15}$$

The link travel disutility related to link traffic volume, $\mathbf{d}(\mathbf{x})$, is normally positive and increasing. Therefore, the above matrix is positive since it is a diagonal matrix with positive entries.

Since the objective function is convex and constraints are all linear, the minimization model can be solved by the Frank-Wolfe (FW) method [18]. The FW method is based on finding a descent direction by minimizing a linear approximation of the objective function at the current solution point. This linear approximation is given by $z(\mathbf{u}^n) = z(\mathbf{x}^n) + \nabla z(\mathbf{x}^n)(\mathbf{u}^n - \mathbf{x}^n)'$ with respect to $\mathbf{x}^n$ for the nth iteration. This linear function of $\mathbf{u}^n$ has to be minimized subject to the constraints of the original model:

$$\min \quad z(\mathbf{u}^n) = z(\mathbf{x}^n) + \nabla z(\mathbf{x}^n)(\mathbf{u}^n - \mathbf{x}^n)' \tag{16}$$

$$\text{subject to} \quad \sum_k g^{rs}_k = q_{rs} \quad \forall\, r, s \tag{17}$$

$$g^{rs}_k \geq 0 \quad \forall\, k, r, s \tag{18}$$

with the incidence relationship

$$u_m^n = \sum_{rs} \sum_k g^{rs}_k \delta^{rs}_{m,k} \quad \forall\, m \tag{19}$$

where g denotes the path flows. The objective function can be simplified to min $\nabla z(\mathbf{x}^n)\,\mathbf{u}^{n\prime}$ because $z(\mathbf{x}^n)$ and $\nabla z(\mathbf{x}^n)\mathbf{x}^n$ are constant when $\mathbf{x}^n$ is known. Moreover, min $\nabla z(\mathbf{x}^n)\,\mathbf{u}^{n\prime}$ = min $\sum_m (\partial z(\mathbf{x}^n)/\partial x_m)\, u_m^n$ = min $\sum_m d^n_m(x^n_m)\, u_m^n$. Since the travel times $\mathbf{d}^n$ are fixed for a given

$\mathbf{x}^n$, and the program calls for minimizing the total travel time over the network with flows independent of travel times, the solution of the above linear program, $\mathbf{u}^n$ which is an auxiliary flow representing the descent direction, can be solved by assigning all motorists to the smallest travel-disutility path connecting their origin and destination.

After the descent direction is decided, the optimal move size between $\mathbf{x}^n$ and $\mathbf{u}^n$ can be found by solving the following linear program. " is between 0.0 and 1.0 and is chosen so that the link volumes are as close as possible to the user-optimized equilibrium loadings. The bisection method can be used to solve the linear program.

$$\min \quad z[\mathbf{x}^n + \text{"}(\mathbf{u}^n - \mathbf{x}^n)] \tag{20}$$

$$\text{subject to} \quad o \# \text{"} \# 1 \tag{21}$$

The initial solution can be determined by applying an all-or-nothing network loading procedure to an empty network. The remaining algorithmic steps are to find the descent direction and the optimal move size iteratively until the stopping criterion for solving the UE program is met.

The assumption in the UE conditions that each motorist has full information about link performance relationships may be extended to the dynamic case. However, with the Advance Traveler Information System, motorists could be provided with travel information for pre-trip planning (i.e., travel mode, departure time, and route) and guidance for en-route diversion. Therefore, with the assumption that ATIS could continuously provide all travelers with full information about all link disutilities, the UE conditions may be adoptable in the dynamic situation. Moreover, enhancing the model structures and solution algorithms to the dynamic case is feasible by adding discrete time periods as an additional dimension.

2.3.2 System Optimization

The system-optimization (SO) program minimizing the total travel time spent in the network while satisfying the flow conservation constraints is expressed as follows [19]:

$$\min \quad z(\mathbf{x}) = \sum_m x_m d_m(x_m) \tag{22}$$

$$\text{subject to} \quad \sum_k f^{rs}_k = q_{rs} \quad \forall\, r, s \tag{23}$$

$$f^{rs}_k \geq 0 \quad \forall\, k, r, s \tag{24}$$

with the incidence relationship

$$x_m = \sum_{rs} \sum_k f^{rs}_k \delta^{rs}_{m,k} \quad \forall\, m \tag{25}$$

The general first-order conditions for the minimization of the SO program can be derived using a similar method as for UE and expressed as follows [3]:

$$f^{rs}_k (u^{rs}_k - y_{rs}) = 0 \quad \forall\, k, r, s \tag{26}$$

$$u^{rs}_k - y_{rs} \geq 0 \quad \forall\, k, r, s \tag{27}$$

$$\sum_k f^{rs}_k = q_{rs} \quad \forall\, r, s \tag{28}$$

$$f^{rs}_k \geq 0 \quad \forall\, k, r, s \tag{29}$$

where $u^{rs}_k = \sum_m \delta^{rs}_{m,k} [d_m(x_m) + x_m \, d d_m(x_m)/d x_m]$ is the marginal total travel time on path k connecting O-D pair r-s.

The marginal total travel times on all of the used paths connecting a given O-D pair are equal for the SO program at its optimal value which minimizes the total travel time for the network, not for the users. It does not represent an equilibria situation and is not stable.

2.3.3 Stochastic User Equilibrium

Stochastic user equilibrium (SUE) models relax the assumption that motorists have full information about link travel times on all links in a network by assuming that each motorist may perceive a different travel time (disutility) but will still choose a path with the

least perceived disutility path from his or her origin to his or her destination. At SUE, no motorist can improve his or her perceived disutility (travel time) by unilaterally changing routes.

A minimization program was developed by Sheffi and Powell [20]. The solution of the program is the desired set of SUE flows. The unconstrained program is shown as follows:

$$\min_{\mathbf{x}} z(\mathbf{x}) = -\sum_{rs} q_{rs} E[\min_{k} U_k^{rs} \,|\mathbf{u}^{rs}(\mathbf{x})\,] + \sum_{m} x_m d_m(x_m) - \sum_{m} \int_0^{x_m} d_m(\omega)d\omega \qquad (30)$$

where U_k^{rs} represents the perceived travel disutility on route k between origin r and destination s. U_k^{rs} is a random variable with the mean equal to the actual travel disutility u_k^{rs} which is measured at a given flow level **x**.

The first term of the objective function includes the expected perceived travel disutility function from origin r to destination s, $E[\min U_k^{rs}| \mathbf{u}^{rs}(\mathbf{x})]$. The partial derivative of this function with respect to U_k^{rs} is the probability of choosing path k between r and s, P_k^{rs}, which is the probability that U_k^{rs} is less than the disutility of any other route, U_i^{rs}, between r and s.

The various models for the probability of selecting each alternative route differ from each other in the assumed distributions of the variance of U_k^{rs}, $>^{rs}_k$. If $>^{rs}_k$ are identically and independently distributed Gumbel variables, P_k^{rs} can then be expressed as a logit model:

$$P_k^{rs} = \frac{e^{-u_k^{rs}}}{\sum_{l} e^{-u_l^{rs}}} \qquad (31)$$

The network loading approach (in which the link disutility function is not flow dependent) associated with this multinomial logit choice model, known as the STOCH method, includes a forward step to calculate the "link weights" according to the probabilities of selecting each reasonable alternative route (which includes only links that take the traveler further away from the origin and closer to the destination) and a backward step to assign the flows at destinations back to the origins [21].

If $>^{rs}_k$ is normally distributed, the joint density function of $\mathbf{>^{rs}}$ is the multivariate normal (MVN) function with mean vector **0** and variance matrix $\mathbf{V}^{rs}$. The distribution of $\mathbf{U}^{rs}$ can then be modeled as multivariate normal, too: $\mathbf{U}^{rs} \sim \text{MVN}\ (\mathbf{u}^{rs}(\mathbf{x}), \mathbf{V}^{rs})$. However, P_k^{rs} cannot be expressed analytically since the cumulative normal distribution function cannot be evaluated in closed form. One of the approaches to compute P_k^{rs} is based on a Monte Carlo simulation procedure which has no restriction on the number of alternative routes unlike other analytical approximation methods. A network loading algorithm based on the Monte Carlo procedure was tested by Sheffi and Powell [22] and summarized by Sheffi [3] as:

Step 0: Initialization. Set $n = 1$ where n is the number of iterations.

Step 1: Sampling. Sample D^n_m from $D_m \sim N(d_m, \text{"}d_m)$ for each link m. (" is variance of the perceived travel time over a road segment of unit travel time.)

Step 2: All-or-nothing assignment. Based on $\mathbf{D}^n$ assign **q** to the shortest path connecting each O-D pair r-s. This step yields the set of link flows $\mathbf{x}^n$.

Step 3: Flow averaging.

$$\text{Let } x^n_m = [(n-1)x^{n-1}_m + x^n_m]/n, \text{ œ } m \qquad (32)$$

Step 4: Stopping test.

(a) Let

$$\sigma_m^n = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} [x_m^i - x_m^n]^2} \qquad \forall\ m \tag{33}$$

(b) If $\max_m \{F^n_m/x^n_m\}$ # 6, stop. The solution is $\mathbf{x}^n$. Otherwise, set n = n + 1 and go to step 1.

To prove the equivalency between the above minimization program (Eq. (30)) and the SUE condition, the first-order conditions of this program have to coincide with the SUE conditions, which can be characterized as follows [3]:

$$f_k^{rs} = q_{rs}\, P_k^{rs,} \qquad \text{œ } k, r, s \tag{34}$$

The partial derivative of z (**x**) with respect to a typical path flow, f_k^{rs}, can be written as

$$\frac{\partial z[\mathbf{x}(\mathbf{f})]}{\partial f_k^{rs}} = \sum_m (-q_{rs}P_k^{rs} + f_k^{rs}) \frac{dd_m(x_m)}{dx_m}\, \delta_{m,\,k}^{rs} \tag{35}$$

Since the first-order condition for unconstrained minimizations requires only that the gradient vector of the objective function be equal to **0**, the above derivatives have to equal zero for all k, r, and s, meaning that

$$f_k^{rs} = q_{rs}\, P_k^{rs} \qquad \text{œ } k, r, s \tag{36}$$

which is the SUE condition. Moreover, the flow conservation constraints (' $_k f^{rs}_k = q_{rs}$) are automatically satisfied at equilibrium since 3$_k P_k^{rs}$ =1. The uniqueness conditions of z(**x**) were also demonstrated by Sheffi [3] by showing that the Hessian matrix of z(**x**) is positive definite.

The core of any descent method for solving the unconstrained minimization model with a nonlinear convex function of several variables is to find the descent direction and minimize the objective function along that direction at each iteration. However, the iterative

process to find the descent direction and move size of Eq. 30 is difficult because the direction vector computed at a particular iteration could be random in some cases.

An algorithm known as the method of successive averages (MSA) which is based on a predetermined move size along the descent direction was proposed to solve Eq. 30 and proved to converge to the minimum by Sheffi and Powell [22]. The algorithm is summarized in Sheffi [22]:

Step 0: Initialization. Perform a stochastic network loading based on a set of initial travel disutilities $\mathbf{d}^0$. This generates a set of link flows $\mathbf{x}^1$. Set n = 1.

Step 1: Update. Set $\mathbf{d}^n = \mathbf{d}(\mathbf{x}^n)$, œ n.

Step 2: Direction finding. Perform a stochastic network loading procedure based on the current set of link travel disutilities, $\mathbf{d}^n$. This yields an auxiliary link flow pattern $\mathbf{u}^n$.

Step 3: Move. Find the new flow pattern by using the predetermined move size 1/n. $\mathbf{x}^{n+1} = \mathbf{x}^n + (1/n)(\mathbf{u}^n - \mathbf{x}^n)$

Step 4: Convergence criterion.

Further research is expected to enhance the complicated model structures of SUE to the dynamic case. However, the MSA method using the predetermined move size to guarantee convergence can be applied in the solution algorithm of the DTA model developed in this study.

2.4 Combining a Static Traffic Signal Timing Plan and Traffic Assignment

Gartner and Improta [23] derived the following compound mathematical optimization formulation to describe the static traffic management system:

$$\min_{M} \left\{ Z_1 \left[M,\ \arg\min_{F} Z_2(F) \right] \right\} \tag{37}$$

subject to flow conservation constraints:

$$\sum_{p \in P_{(i,k)}} h_p = r_{(i,k)} \quad \text{with } (i,k) \in I \tag{38}$$

and non-negativity constraints:

$$h_p \; ; \; r_{(i,k)} \; \$ \; 0 \tag{39}$$

where

M = the set of management variables under the control of the traffic manager

I = the set of all O-D pairs (i,k)

$P_{(i,\,k)}$ = the set of all simple paths between O-D pair (i, k)

F = the set of all link flows

f_j = the flow on link j

$r_{(i,\,k)}$ = the rate of trip interchange (demand in vehicles) between origin node i and destination node k, with (i, k) 0 I

h_p = the flow on path p, with p 0 $P_{(i,\,k)}$

Z_1 = 3$_j$ f_j w_j (f_j, M)

Z_2 = 3$_j$ $\int_0^{f_j} c_j(x)\,dx$

c_j (f_j) = the average user-perceived travel cost function on link j

w_j (f_j, M) = a more general performance function reflecting the multiplicity of objectives pursued by the traffic manager in the public's interest

The formulation consists of a primary optimization program (the system-optimization), min Z_1 (A), and a secondary optimization program (the user-optimization), argmin Z_2 (A). The "argmin" of a mathematical program is the optimal solution of the program. Examples of computational procedures for solving the whole problem were also given.

The above framework can be extended to the dynamic case by changing the O-D demand and control actions to being time-dependent. However, there is no mathematical methodology that can solve the compound program simultaneously. If the system dynamics are slow and the process is relatively stable and predictable, the steady state models may be applied in consecutive time steps. If the dynamics of the system are more stochastic and thus less predictable, the steady-state approach is not applicable any more.

2.5 Real-Time Traffic-Responsive Control System

The application of advanced technologies in ITS makes it possible for on-line traffic control systems to respond to real-time traffic information. Three published research studies about real time traffic control systems, which take dynamic traffic demand into account, are optimization policies for adaptive control, non-linear program of dynamic traffic-responsive signal control system, and variational inequality model for dynamic traffic control. These are discussed in the following sections.

2.5.1 Optimization Policies for Adaptive Control

Gartner [24] used a rolling horizon approach, which is used by operations research analysts in production-inventory control, to develop a demand-responsive strategy for traffic signal control. The basic steps in the process are as follows:

Step 0: Determine the stage length, which is the project horizon consisting of k intervals, and the roll period r, which specifies the traffic condition is updated; the process is repeated every r intervals.

Step 1: Obtain flow data for the first r intervals from detectors and calculate flow data for the next k-r intervals from a model.

Step2: For a given switching sequence, the total delay on each approach was formulated as:

$$d(t_1, t_2, t_3) = \sum_{i=1}^{k} (Q_o + A_i - D_i) \qquad (40)$$

where Q_o = initial queue

A_i = arrivals during interval i

D_i = departures during interval i

t_1, t_2, t_3 = possible switching times during this stage

Calculate the optimal switching policy for the entire stage by an optimal sequential constrained search method in which the total delay is evaluated sequentially for all feasible switching sequences.

Step 3: Implement the switching policy for the roll period only.

Step 4: Shift the projection horizon by r units to obtain a new stage; repeat steps 1-4.

The rolling horizon strategy provides an effective method for solving the dynamic traffic control problem in real time. However, one argument about the "rolling horizon" technology in traffic control is that this method minimizes the total delay at each roll period r using the predicted and thus unreliable information about traffic flow for the future finite time period (the remaining horizon length) [13].

2.5.2 Non-Linear Program of Dynamic Traffic-Responsive Signal Control System

Chen and Hsueh [12] formulated a non-linear program for the dynamic traffic-responsive signal control system using Webster's delay formula. Webster's model uses an analytical method to replicate the delay time, which includes the time a vehicle is stopped while waiting to pass through the intersection and the time lost during acceleration and deceleration from/to a stop. Chen and Hsueh proposed a non-linear program as illustrated in Eq. 41 to minimize the total system delay. First, the approach delay is obtained by multiplying the traffic flow on an approach by the average delay per vehicle. The total system delay is then calculated by summing up the approach delay of each intersection in the network for every time interval. The proposed model is as follows:

$$\min_{g} Z = \sum_{t}\sum_{a}\sum_{m} v_a^m(t)\frac{9}{10}\left\{ \frac{C_I(t)[1-g_I^m(t)/C_I(t)]^2}{2(1-v_a^m(t)/S_a^m)} + \frac{[H_a^m(t)]^2}{2v_a^m(t)[1-H_a^m(t)]} \right\} \qquad (41)$$

subject to $H_a^m(t) = [v_a^m(t) / S_a^m]\,[C_I(t) / g_I^m]$ œI, a, m, t (42)

$H_a^m(t) < 1$ œa, m, t (43)

$g_I^m(t)$ $ min g_I^m œI, m, t (44)

3$_m$ $[g_I^m(t) + l_I^m] = C_I(t)$ œI, t (45)

where

a = link designation

$C_I(t)$ = cycle length for intersection I during interval t

$g_I^m(t)$ = green time associated with phase m at intersection I during interval t

$H_a^m(t)$ = degree of saturation over link "a" associated with phase m during interval t

l_I^m = lost time associated with phase m at intersection I

S_a^m = saturation flow on link "a" associated with phase m

$v_a^m(t)$ = exit flow from link "a" in phase m during interval t

The Frank-Wolfe method was used to solve the model since the objective function was proved to be convex and all constraints are linear. The framework of the above non-linear program is adopted in this research because of the intuitive interpretation of the mathematical model, the promise of the optimization model for finding the global minimum, and the amenability to computation with the application of computer programming. Although the concept of the above framework can ideally describe the environment of the real-time traffic control system, the delay function is the major key to locating the optimal signal timing settings. The delay function used in the above research over simplified the traffic congestion situation but not taking the spill-back queue problem into account.

2.5.3 Variational Inequality Model for Dynamic Traffic Control

The characteristics of the VI formula have been extensively studied in economics. A mathematical program often may be more intuitive for representing a real situation. However, the VI formulation is much broader than a mathematical program.

Chen and Ben-Akiva [14] defined a dynamic system optimum principle, which can be expressed in Eq. 46 - 48 for dynamic traffic control, as: for each intersection, any phase with a positive green time must have an equal and minimal marginal delay.

$$g_i^m(t)\,(c_i^m(t) - B_i(t)) = 0 \qquad \forall\, i, m \tag{46}$$

$$c_i^m(t) - B_i(t) \geq 0 \qquad \forall\, i, m \tag{47}$$

$$g_i^m(t) \geq 0 \qquad \forall\, i, m \tag{48}$$

where

$g_i^m(t)$ = green time for phase m of intersection i at time t

$c_i^m(t)$ = marginal cost for phase m of intersection i at time t

$B_i(t)$ = minimal marginal phase delay for intersection i at time t

The equivalent formulation for Eqs. 46-48 was then stated as follows:

$$\int_0^T \sum_i \sum_m c_i^{m*}(t)(g_i^m(t) - g_i^{m*}(t))\, dt \geq 0 \qquad \forall g_i^m(t) \in \mathcal{G} \tag{49}$$

where $g_i^{m*}(t)$ is the dynamic system optimal setting, $c_i^{m*}(t)$ is the marginal phase delay when the timing is $g_i^{m*}(t)$ and $\mathcal{M}$ is the feasible set for green time splits.

The VI formula of the real-time traffic-responsive signal control system is broader than the mathematical program. However, no solution algorithm was developed in the above research although the equivalent mathematical formulation for the VI model was provided.

2.6 Dynamic Traffic Assignment

Static traffic assignment distributes on O-D traffic flow based on the assumption that the traffic flow on a network is static, whereas dynamic traffic assignment models can present traffic situations on a network in infinitesimal short time periods. Therefore, static traffic assignment is sufficiently effective to predict the traffic flow for a long-term period but dynamic traffic assignment is powerful enough to analyze the traffic state over a specific time such as a peak hour or shorter period.

Merchant and Nemhauser [25], who proposed a discrete time, nonlinear and system-optimal model for the case of single-destination networks, were among the first researchers to address the dynamic traffic assignment system. The techniques available for DTA have progressed since then. Two kinds of formulations published in existing research studies about DTA are VI formulations and route-based algorithms. These are discussed in the following sections.

2.6.1 Variational Inequality (VI) Formulation for DTA

Drissi-Kaitouni [4] proved three theorems about the VI formulation for DTA, which is equivalent to the equilibrium DTA condition:

$$S_k(\mathbf{h}) = u^t_{pq} \quad \text{if } h_k > 0 \quad \text{œ } p, q, t, k \tag{50}$$

or

$$S_k(\mathbf{h}) \$ u^t_{pq} \quad \text{if } h_k = 0 \quad \text{œ } p, q, t, k \tag{51}$$

where $S_k(h)$ is the travel cost on a path k and u^t_{pq} is the travel cost on the shortest path from origin node p to the destination node q at period t, given traffic flow **h** on the network. These three theorems are summarized as follows:

Theorem 1. Let S_h be the set of feasible path flows. Then the above equilibrium condition, Eq. 50, may be rewritten as a variational inequality:

Find h 0 S_h such that $\Sigma_p \Sigma_q \Sigma_t \Sigma_k (S_k(\mathbf{h}) - u^t_{pq}) (h_k' - h_k) \geq 0 \quad \forall \mathbf{h}' \in S_h$.

Theorem 2. The variational inequality in Theorem 1 is equivalent to the following variational inequality:

Find h 0 S_h such that $\Sigma_p \Sigma_q \Sigma_t \Sigma_k S_k(\mathbf{h}) (h_k' - h_k) \geq 0 \quad \forall \mathbf{h}' \in S_h$.

Theorem 3. The variational inequality in Theorem 2 has a link variational inequality formulation:

Find f 0 S_f such that $\Sigma_a S_a(\mathbf{f}) (f_a' - f_a) \geq 0 \quad \forall f' \in S_f$.

where S_f is the set of feasible link traffic patterns over the network.

He also pointed out that according to the symmetry principle, the above link-based VI program is equivalent to the UE mathematical programming model in Eq. 1 if $S_a(\mathbf{f})$ is equal to $S_a(f_a)$.

Chen and Hsueh [12] proposed a nested diagonalization algorithm to solve the following dynamic user-optimal route choice model which was formulated as a VI model.

$$\mathbf{c}^* [\mathbf{u} - \mathbf{u}^*] \geq 0, \forall u \in S \tag{52}$$

where S denotes the feasible region that is delineated by the following constraints:

Flow conservation constraints:

$$\Sigma_p h_p^{rs}(k) = q^{rs}(k) \quad \forall r, s, k \tag{53}$$

Flow propagation constraints:

$$u_{apk}^{rs}(t) = \Sigma_{rs} \Sigma_p \Sigma_k h_p^{rs}(k) \delta_{apk}^{rs}(t) \quad \forall a, t \tag{54}$$

$$u_{apk}^{rs}(t) = u_{bpk}^{rs}(t-\Delta_b(t)) \delta_{apk}^{rs}(t) \quad \forall r, s, p, k, t, a \in p, b \in p, a \in B(j), b \in A(j) \tag{55}$$

Nonnegativity constraints:

$$h_p^{rs}(k) \geq 0 \quad \forall r, s, p, k \tag{56}$$

where

$A(j)$ = set of links whose tail node is j

$B(j)$ = set of links whose head node is j

$h_p^{rs}(k)$ = flows from origin r departing during interval k over route p toward destination s

$q^{rs}(k)$ = departure flows between origin-destination rs during interval k

$u_a(t)$ = inflow entering link "a" during interval t

$u_{apk}^{rs}(t)$ = portion of inflows entering link "a" during interval t which departs origin r during interval k over route p toward destination s

$J_a(t)$ = travel time on link "a" during interval t

$*_{apk}^{rs}(t)$ = 1, if flows departing origin r over route p during interval k entering link "a" during interval t

= 0, otherwise

Since the Jacobian matrix of the travel time function was asymmetric, two relaxation (or diagonalization) techniques were needed to transform the VI model to a non-linear optimization problem. One is to estimate the link travel time and the other is to temporarily fix the flow on each link other than on the subject time-space link. The solution algorithm which combines diagonalization and the Frank-Wolfe method can be summarized as follows:

Step 0: Initialization.

Step 1: "First Loop" Operation. Update the estimated link travel times based on the initial traffic flow conditions. Construct the corresponding feasible time-space network based on the estimated link travel times.

Step 2: "Second Loop" Operation. Modify the initial feasible solution based on the time-space network constructed by the estimated link travel time from Step 1. Fix the flows on all links to transform the VI model to a non-linear optimization model.

Step 3: Solve the optimization program in Step 2 using the FW method.

Step 4: Convergence check for the "Second Loop" operation.

Step 5: Convergence check for the "First Loop" operation.

The Variational Inequality is a useful technique to formulate the DTA system. Applying the relaxation algorithm to relax the link interactions to find the equivalent optimization model is the key action to solve the VI model. However, Chen and Hsueh [12] could not define a good estimate of link travel times to relax the link interactions in the DTA model when they derived the above solution algorithm. The VI formulation and the relaxation algorithm are applied in this research to solve the DTA system. Greater effort is devoted to finding reasonable estimates of the link travel times.

2.6.2 Route-Based Algorithms for DTA

To solve the dynamic user-optimal route choice problem, Chen et al. [26] compared three route-based algorithms with the link-based algorithm using the Frank-Wolfe method. The three route-based algorithms were the desegregate simplicial decomposition of Larsson et al., desegregate simplicial decomposition of von Hohenbalken, and a gradient projection method. They found that the link-based algorithm is inferior to the route-based algorithms in terms of execution time but superior in terms of memory requirements because the results from the route-based algorithms for the DTA system include not only the link traffic volumes but also the information related to turning-movement volumes.

The traffic volume for each turning movement at every intersection is essential to feed into a traffic control model. With this information, a traffic control model can determine the optimal green time for each signal phase which allows only specific traffic movements to move. However, CORSIM's assignment feature using a link-based algorithm to solve the assignment problem also provides turning movement information using extra

memory to record the route information at each iteration. Therefore, the link-based algorithm is adopted in this research with the application of CORSIM's assignment feature.

2.7 Combining a Dynamic Traffic Signal Timing Plan and Traffic Assignment

Most research addressing the interaction between traffic control and assignment only considers the static traffic situation. Two approach techniques for integrating DTA with real-time traffic-responsive control systems, which were proposed by recent studies, are game theory and bilevel programming. These are summarized in the following sections.

2.7.1 Game Theory Approach

Game theory provides a framework for modeling a decision-making process in which more than one player is involved and each individual's actions determine the outcome jointly. Fisk [27] analyzed the characteristics of a Stackelberg game in which one player knows how the other players will respond to any decision he or she may make, and the characteristics of a Nash noncooperative game in which each player is trying to minimize his or her performance function without prior knowledge of the other players' functions. The user equilibrium condition can be stated identically as a Nash noncooperative game with each traveler considered as a player. The signal optimization model can be formulated as a Stackelberg game in which the traffic authority tries to minimize the network performance function and motorists choose static user-optimal routes.

Chen and Ben-Akiva [14] combined the dynamic traffic control system and the dynamic traffic assignment system as three game theories. First the combined control-assignment system is formulated as a Cournot game, in which the players, i.e., the traffic authority and the users, choose their strategies simultaneously. In this game, each player makes his or her move independently without knowing the strategy of the other. Second, the

combined control-assignment system is formulated as a Stackelberg game in which the authority sets the signal timing first by anticipating the traffic flow and motorists then choose their best routes accordingly. In the end, the combined control-assignment system is formulated as a monopoly game in which both control and assignment solutions are system-optimal since the traffic authority controls both the signal setting and traffic assignment. However, only frameworks, not solution algorithms, were discussed for these game theories.

2.7.2 Bilevel Programming

Bilevel programming involves two levels of mathematical programming that can be viewed as a particular case of Stackelberg games. At the upper level, decision makers are bound by the decisions of the lower levels and maximize their own profit accordingly, taking into account the reactions of the lower levels.

Marcotte [28] presented the static network design problem as a bilevel programming model. Two of the four heuristic procedures analyzed to solve the bilevel model were iterative optimization assignment procedures with user-optimized and system-optimized equilibrium models. The numerical experiments showed that the iterative optimization assignment method with the user-optimal equilibrium model yielded a near-optimal solution.

Chen and Hsueh [12] combined the dynamic traffic control and the traffic assignment systems into a bilevel model. A heuristic procedure, involving an iterative optimization and assignment method, was used to solve the model for a near-optimal solution.

Since it does not appear possible to find a polynomial algorithm to solve the combined model because of the non-linear programming structures in both the traffic control and assignment models, the concept of bilevel programming and the iterative optimization and assignment method are pursued in this research.

2.8 Generalized Delay Model

A generalized delay model was proposed by Fambro and Rouphail [29], and validated by Rouphail et al. [30]. The model is a much improved version over the previous HCM model in estimating delay at vehicle-actuated traffic signals. Moreover, it includes a term to account for the effects of queues not being cleared during a signal cycle, a condition referred to as having oversaturated queues during variable demand conditions. This generalized delay model for Chapter 16 of the HCM is as follows:

$$d = \frac{0.5C[1 - (g/C)]^2}{[1 - (g/C)\min(X, 1.0)]} PF + 900T[(X-1)+\sqrt{(X-1)^2+8kIX/Tc}] + d_3 \qquad (57)$$

There are three cases for estimating d_3:

(1) If no oversaturated queue exists at the start of the analysis period, $d_3 = 0$.

(2) If an oversaturated queue exists at the start, but not at the end of the analysis period, $d_3 = (3600N_i/c)\ [0.5N_i/Tc(1-X)]$.

(3) If an oversaturated queue exists at both the start and the end of the analysis period, $d_3 = (3600N_i/c) - 1800T[1 - \min(X, 1.0)]$.

where

C = average cycle length

g = average effective green time

X = degree of saturation for a subject lane group

PF = progression adjustment factor

T = analysis period

k = parameter for given arrival and service distributions

I = parameter for variance-to-mean ratio of arrivals from upstream signal

c = capacity of the lane group

N_i = initial queue at the start of the analysis period

This generalized delay model for signalized intersections can provide good estimates of intersection delays in a network for allocating appropriate green times at intersections because it takes the effects of oversaturated queues into account.

2.9 An Evaluation Tool: CORSIM 5.1

There are several simulation software packages that are currently utilized by transportation professionals to evaluate different traffic scenarios. For example, the TRansportation ANalysis SIMulation System (TRANSIMS) is designed to simulate the detailed interaction between individuals' activity plans and congestion on the transportation system [31]. The program is capable of simulating the movements of individuals across the network, including mode selection, on a second-by-second basis. However, intensive data collection and extreme computer execution time are currently obstacles to large-scaled applications of TRANSIMS. CORSIM 5.1 [32], on the other hand, is a comprehensive microscopic traffic simulation computer program using commonly accepted vehicle and driver behavior models. Therefore, it is usually used as an evaluation tool for traffic signal timings although it also provides a user-equilibrium platform to perform a traffic assignment. CORSIM's traffic assignment feature as well as its evaluation function are adopted in this research although there are not many reports evaluating CORSIM's effectiveness for traffic assignment. It is applied because the routing logic in CORSIM 5.1 will convert the O-D tables into turning percentages for each intersection, which can then be fed into the traffic control model. In each iteration of CORSIM's assignment procedure, an intermediate solution is obtained using link travel times produced by the previous iteration (direction finding). An iterative line search is then applied to the range between the current intermediate solution and the previous iteration solution to obtain an optimal solution for

each iteration (optimal moving size). The assignment process will terminate when the change in the objective function between two successive iterations is less than a threshold value (convergence test). The Bureau of Public Roads (BPR) and the modified Davidson link impedance functions are available to evaluate the travel time on a link. The BPR formula is as follows:

$$T = T_0 \, [1 + a(V/C)^b] \tag{58}$$

The modified Davidson impedance function is as follows:

$$T = T_0 \, [1 + aV/(S-V)] \qquad \text{if } V \# bS \tag{59}$$

or

$$T = T_0 \, [1 + ab/(1-b)] + aT_0 \, (V-bS)/[S(1-b)^2] \qquad \text{if } V > bS \tag{60}$$

where

T = travel time on link

T_0 = free-flow travel time on link

a, b = parameters to be estimated for each class of roadway

V = volume on link

C = capacity on link

S = saturation rate on link

The BPR formula increases travel times even after a link's traffic flow is higher than its capacity. The modified Davidson function, using a linear extension at a volume close to capacity, has been shown to reduce the error in traffic assignment results relative to actual volume counts. However, there is very limited estimation experience reported in the literature for the Davidson function. Instead, the BPR formula is frequently adopted in practice and the model parameters which appeared in the original publication are usually employed.

To estimate the capacity used in the impedance functions, the discharge rates for turns are held constant, and are estimated initially for free-flow conditions. These estimates could be calibrated after the assignment of turn movements, then applied to the next assignment process, if requested. The following equation is used for capacity calibration:

$$C_n = [rC_c + (100-r)C_p]$$

where

C_n = new estimate of capacity (for the next assignment iteration)

r = capacity smoothing (in a percentage)

C_c = calculated capacity using previously assigned volumes

C_p = previous estimate of capacity

CHAPTER 3
OPTIMIZATION MODEL FOR DYNAMIC TRAFFIC CONTROL

3.1 Introduction

This chapter presents the techniques applied for developing a mathematical model and the solution algorithm for optimizing dynamic traffic controls. In the mathematical model, the total intersection delay in a network is minimized by allocating optimal time splits of each traffic signal. The HCM generalized delay model is used to estimate the average delay for vehicles arriving during a discrete time interval. However, the objective function of the optimization model , Eq. 61, cannot be proven to be convex. An approach which can deal with this situation is presented. The last section in this chapter presents a solution algorithm for solving the optimization model.

3.2 Optimization Model

As described in Section 2.8, three cases corresponding to the oversaturation situations presented in the HCM delay model are formulated for the dynamic traffic control system by adding time as an additional dimension:

1. No oversaturated queue exists: the degree of the saturation at the previous time period, $X^p_m(t-1)$, is less than one.

2. An oversaturated queue exists at the start, but not at the end of the analysis period: the degree of the saturation at previous time period, $X^p_m(t-1)$, is greater than or equal to one but the degree of the saturation at current time period, $X^p_m(t)$, is less than one.

3. An oversaturated queue exists at both the start and the end of the analysis period: both of the degrees of the saturation, at previous time period, $X^p_m(t-1)$, and at current time period, $X^p_m(t)$, are greater than or equal to one.

The model for the dynamic traffic control system is expressed as follows:

$$\min\ Z(g) = \sum_{t}\sum_{I}\sum_{p}\sum_{m\in B(I)} v_m^p(t)\ d_m^p(t) \tag{61}$$

$$X_m^p(t) = \frac{v_m^p(t)}{s_m}\ \frac{C_I(t)}{g_I^p(t)} \qquad \forall I,\ m,\ p,\ t \tag{62}$$

If $X^p_m(t-1) < 1$ and $X^p_m(t) < 1$, then

$$d_m^p(t) = \frac{0.5C_I(t)[1-g_I^p(t)/C_I(t)]^2}{1-\frac{v_m^p(t)}{s_m}}PF$$

$$+\ 900T[(X_m^p(t)-1)+\sqrt{(X_m^p(t)-1)^2+8kRX_m^p(t)/(T\frac{s_m g_I^P(t)}{C_I(t)})}\ ] \tag{63}$$

If $X^p_m(t-1)$ $ 1 and $X^p_m(t) < 1$, then

$$d_m^p(t) = \frac{0.5C_I(t)[1-g_I^p(t)/C_I(t)]^2}{1-\frac{v_m^p(t)}{s_m}}PF\ +\ 900T[(X_m^p(t)-1)+\sqrt{(X_m^p(t)-1)^2+8kRX_m^p(t)/(T\frac{s_m g_I^P(t)}{C_I(t)})}\ ]$$

$$+\ [3600(v_m^p(t-1)-s_m\frac{g_I^p(t-1)}{C_I(t-1)})/(\frac{s_m\ g_I^p(t)}{C_I(t)})]$$

$$*\ [0.5(v_m^p(t-1)-s_m\frac{g_I^p(t-1)}{C_I(t-1)})/T(\frac{s_m\ g_I^p(t)}{C_I(t)})(1-X_m^p(t))] \tag{64}$$

If $X^p_m(t-1)$ $ 1 and $X^p_m(t)$ $ 1, then

$$d_m^p(t) = \frac{0.5C_I(t)[1-g_I^p(t)/C_I(t)]^2}{1-\frac{g_I^p(t)}{C_I(t)}}PF + 900T[(X_m^p(t)-1)+\sqrt{(X_m^p(t)-1)^2+8kRX_m^p(t)/(T\frac{s_m g_I^p(t)}{C_I(t)})}\ ]$$

$$+ [3600(v_m^p(t-1)-s_m\frac{g_I^p(t-1)}{C_I(t-1)})/(\frac{s_m\ g_I^p(t)}{C_I(t)})] \qquad (65)$$

subject to

$g_I^p(t)$ $ $\bar{g}_I^p$ œ I, p, t (66)

3 $_P$ $[g_I^p(t) + l_I^p] = C_I(t)$ œ I ,t (67)

where

m = link index

B(I) = set of links entering intersection I

I = intersection index

p = phase index

$v_m^p(t)$ = vehicle flow on link m during phase p at time t

$d_m^p(t)$ = vehicle delay on link m during phase p at time t

s_m = saturation flow on link m

$g_I^p(t)$ = green time associated with phase p at intersection I during interval t

$C_I(t)$ = cycle length for intersection I at interval t

$X_m^p(t)$ = degree of saturation associated with phase p at intersection I during interval t

$\bar{g}_I^p$ = minimum green time associated with phase p at intersection I

l_I^p = lost time associated with phase p at intersection I

PF = progression adjustment factor

T = length of time interval

k = parameter for given arrival and service distributions

R = parameter for variance-to-mean ratio of arrivals from upstream signal

The objective function minimizes the total intersection delay, which is represented by the sum of the products of the traffic flow and the average vehicle delay for each signal phase at each intersection during each time period. The average delay model, $d_m^p(t)$, is essentially the HCM generalized delay model which has either two or three terms depending on the degree of saturation during the previous time period. There are two constraints. The first one restricts the green time assigned to each phase to be not less than the minimum green time which is usually determined based on the green time needed for pedestrians to cross streets safely. The second constraint restricts the sum of green times and lost times for all signal phases at each intersection to be equal to the cycle length.

3.3 Convexity

The above optimization model is a nonlinear programming model with linear constraints for conserving cycle length and minimum green time. Since the constraints are all linear, this nonlinear programming model can be solved by the Frank-Wolfe method. The Frank-Wolfe method is a feasible direction method and its direction finding procedure is a descent method, meaning that the objective function value decreases at every iteration. This method converges to a local minimum, which would naturally be a global minimum for convex objective functions.

The objective function of the above optimization model is the sum of nonlinear functions associated with vehicle delay at each intersection. Without considering the

incremental delay due to oversaturated queues, the objective function, Z(g), can be proven to be a convex function by showing that the matrix of the second derivatives of Z(**g**) with respect to **g** (the Hessian) is positive definite.

Assuming there are I intersections in the analysis network and each intersection has p phases, then $\mathbf{g}' = [g^1_1(1), g^1_1(2),..., g^1_1(t), g^2_1(1),..., g^p_I(t)]$. Let $n = I \times p \times t$, and $\mathbf{g}' = [g_1, g_2,..., g_n]$. The Hessian is calculated by using a representative term of the matrix. The derivative of Z(**g**) is therefore taken with respect to green time on the mth and nth elements in $\mathbf{g}'$ so

$$\frac{\partial^2 Z(\mathbf{g})}{\partial g_m \partial g_n} = \begin{cases} \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_n} & \text{for } m = n \\ 0 & \text{otherwise} \end{cases} \tag{68}$$

This means that all of the off-diagonal elements of the Hessian, $\nabla^2 Z(\mathbf{g})$, are zero and all of the diagonal elements are given by $\partial^2 Z(\mathbf{g}) / \partial^2 g_n$. In other words,

$$\nabla^2 Z(\mathbf{g}) = \begin{bmatrix} \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_1} & 0 & 0 & \cdots \\ 0 & \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_2} & 0 & \cdots \\ 0 & 0 & \ddots & \\ \vdots & \vdots & & \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_n} \end{bmatrix} \tag{69}$$

If $\partial^2 Z(\mathbf{g}) / \partial^2 g_i \geq 0$, this matrix is positive definite because it is then a diagonal matrix with positive entries. However, for the oversaturated queue situation, the third term in the generalized delay model, which estimates delay due to oversaturated queues for time interval

t, is dependent on the green time at time interval t-1 because the initial queue at the start of interval t is a variable of the green time at interval t-1. Therefore,

$$\frac{\partial^2 Z(\mathbf{g})}{\partial g_m \partial g_n} = \begin{cases} \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_n} & \text{for } m = n \\ \dfrac{\partial^2 Z(\mathbf{g})}{\partial g_m \partial g_n} & \text{for } m = n-1 \text{ , } m = n+1 \\ 0 & \text{otherwise} \end{cases} \tag{70}$$

This means that not all of the off-diagonal elements of the Hessian are zero. In other words,

$$\nabla^2 Z(\mathbf{g}) = \begin{vmatrix} \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_1} & \dfrac{\partial^2 Z(\mathbf{g})}{\partial g_1 \partial g_2} & 0 & \cdots \\ \dfrac{\partial^2 Z(\mathbf{g})}{\partial g_2 \partial g_1} & \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_2} & \dfrac{\partial^2 Z(\mathbf{g})}{\partial g_2 \partial g_3} & \cdots \\ 0 & \dfrac{\partial^2 Z(\mathbf{g})}{\partial g_3 \partial g_2} & \ddots & \\ \vdots & \vdots & & \dfrac{\partial^2 Z(\mathbf{g})}{\partial^2 g_n} \end{vmatrix} \tag{71}$$

Even if $\partial^2 Z(\mathbf{g}) / \partial g_i \partial g_j \geq 0$, this matrix cannot be proven to be positive definite because it is not a diagonal matrix.

Although the objective function of the above model cannot be proven to be a convex function, there are some methods which can find a near-optimal local minimum. One of these methods is the multistart approach which chooses several starting points and then compares the local minimums determined from those starting points. Since a good starting point is important for finding the global optimum, the static network-wide optimal signal settings, which can be obtained from TRANSYT-7F, are used as the initial signal timings for a near-optimal minimum.

3.4 Solution Algorithm

A procedure using the Frank-Wolfe method is used to solve the real-time traffic control model. To initialize, the link-based traffic volume on each link, which is obtained from the traffic assignment model, is put into TRANSYT-7F to obtain the static network-wide optimal signal settings.

Since the Frank-Wolfe method is a feasible direction method and its direction finding procedure is a descent method, the descent direction at each iteration is then found by assigning a green time equal to the preset minimum green time for each phase (feasible direction) and then assigning the remaining green time to the phase with the highest vehicle delay (descent direction). Moreover, with the linear constraints for conserving cycle length and minimum green time (Eq. 66 and Eq. 67), the preset cycle length has to be greater than the sum of minimum green times and lost times for all signal phases at each intersection. An auxiliary green time setting, $\mathbf{g'}$, is obtained to determine the descent direction.

After the descent direction is decided, the optimal move size between the current solution $\mathbf{g}$ and the auxiliary solution $\mathbf{g'}$ can be found by solving the following linear program:

$$\min \quad z[\mathbf{g}^n + \text{"}(\mathbf{g'}^n - \mathbf{g}^n)] \tag{72}$$

$$\text{subject to} \quad 0 \# \text{"} \# 1 \tag{73}$$

Solving the above linear program is equivalent to finding the value of ", that satisfies $dZ(\text{"})/d\text{"} = 0$. The objective function is differentiable with respect to ". However, the derivative is complicated since the average delay model is quite complex itself. The derivative of the objective function is given as follows:

$$\frac{dZ[\alpha]}{d\alpha} = \sum_t \sum_I \sum_p \sum_{m \in B(I)} v_m^p(t) \frac{dd_m^p(t)}{d\alpha} \tag{74}$$

If $X^p_m(t-1) < 1$ and $X^p_m(t) < 1$, then

$$\frac{dd_m^p(t)}{d\alpha} = \frac{0.5C_I(t)PF}{1-\frac{v_m^p(t)}{s_m}} 2 \left[\frac{1-(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t))}{C_I(t)}\right]\left(\frac{g_I^p(t)-g'^p_I(t)}{C_I(t)}\right)$$

$$+ 900T\left(\frac{v_m^p(t)C_I(t)}{s_m}\frac{g_I^p(t)-g'^p_I(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}+\frac{1}{2}\left((X_m^p(t)-1)^2+\frac{8KRX_m^p(t)}{T\frac{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))}{C_I(t)}}\right)^{-1/2}\right.$$

$$*(2(X_m^p(t)-1)\frac{v_m^p(t)C_I(t)}{s_m}\frac{g_I^p(t)-g'^p_I(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}$$

$$+\frac{8KR}{T}\left(\frac{v_m^p(t)C_I(t)(g_I^p(t)-g'^p_I(t))}{s_m(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))^2}\frac{C_I(t)}{s_m(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))}\right.$$

$$\left.\left.-\frac{C_I(t)^2X_m^p(t)}{s_m^2(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))^2}\frac{s_m(g_I^p(t)-g'^p_I(t))}{C_I(t)}\right)\right)\right) \tag{75}$$

If $X^p_m(t-1)$ $ 1 and $X^p_m(t) < 1$, then

$$\frac{dd_m^p(t)}{d\alpha} = \frac{0.5C_I(t)PF}{1-\frac{v_m^p(t)}{s_m}} \; 2\; [\frac{1-(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t))}{C_I(t)}](\frac{g_I^p(t)-g'^p_I(t)}{C_I(t)})$$

$$+\; 900T(\frac{v_m^p(t)C_I(t)}{s_m}\frac{g_I^p(t)-g'^p_I(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}+\frac{1}{2}((X_m^p(t)-1)^2+\frac{8KRX_m^p(t)}{T\frac{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))}{C_I(t)}})^{-1/2}$$

$$*(2(X_m^p(t)-1)\frac{v_m^p(t)C_I(t)}{s_m}\frac{g_I^p(t)-g'^p_I(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}$$

$$+\frac{8KR}{T}(\frac{v_m^p(t)C_I(t)(g_I^p(t)-g'^p_I(t))}{s_m(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))^2}\frac{C_I(t)}{s_m(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))}$$

$$-\frac{C_I(t)^2X_m^p(t)}{s_m^2(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))^2}\frac{s_m(g_I^p(t)-g'^p_I(t))}{C_I(t)})))$$

$$+\; 3600(\frac{v_m^p(t-1)C_I(t)}{s_m(g_I(t)+\alpha(g'^p_I(t)-g_I^p(t)))} - \frac{C_I(t)}{C_I(t-1)}\;\frac{g_I(t-1)+\alpha(g'^p_I(t-1)-g_I^p(t-1))}{g_I(t)+\alpha(g'^p_I(t)-g_I^p(t))})$$

$$*(T(1-\frac{v_m^p(t)C_I(t)}{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))})^{-1}(\frac{v_m^p(t-1)C_I(t)(g_I^p(t)-g'^p_I(T))}{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}$$

$$-\frac{C_I(t)}{C_I(t-1)}(\frac{g'^p_I(t-1)-g_I^p(t-1)}{g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t))}-\frac{(g'^p_I(t)-g_I^p(t))(g_I^p(t-1)+\alpha(g'^p_I(t-1)-g_I^p(t-1)))}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}))$$

$$+\ 1800(\frac{v_m^p(t-1)C_I(t)}{s_m(g_I(t)+\alpha(g'^p_I(t)-g_I^p(t)))}-\frac{C_I(t)}{C_I(t-1)}\frac{g_I(t-1)+\alpha(g'^p_I(t-1)-g_I^p(t-1))}{g_I(t)+\alpha(g'^p_I(t)-g_I^p(t))})^2$$

$$*\frac{-1}{T(1-\frac{v_m^p(t)C_I(t)}{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))})^2}\frac{v_m^p(t)C_I(t)}{s_m}\frac{g'^p_I(t)-g_I^p(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2} \quad (76)$$

If $X^p_m(t-1)$ $ 1 and $X^p_m(t)$ $ 1, then

$$\frac{dd_m^p(t)}{d\alpha}=\frac{0.5C_I(t)PF\ 2\ (1-\frac{g_I^p(t)+\alpha(g_I^p(t)-g'^p_I(t))}{C_I(t)})\frac{g_I^p(t)-g'^p_I(t)}{C_I(t)}}{1-\frac{g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t))}{C_I(t)}}$$

$$+\ 0.5C_I(t)PF(1-\frac{g_I^p(t)+\alpha(g_I^p(t)-g'^p_I(t))}{C_I(t)})^2\frac{(-1)}{(1-\frac{g_I^p(t)+\alpha(g_I^p(t)-g'^p_I(t))}{C_I(t)})^2}(\frac{g_I^p(t)-g'^p_I(t)}{C_I(t)})$$

$$+\ 900T(\frac{v_m^p(t)C_I(t)}{s_m}\frac{g_I^p(t)-g'^p_I(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}+\frac{1}{2}((X_m^p(t)-1)^2+\frac{8KRX_m^p(t)}{T\frac{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))}{C_I(t)}})^{-1/2}$$

$$*(2(X_m^p(t)-1)\frac{v_m^p(t)C_I(t)}{s_m}\frac{g_I^p(t)-g'^p_I(t)}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}$$

$$+\frac{8KR}{T}(\frac{v_m^p(t)C_I(t)(g_I^p(t)-g'^p_I(t))}{s_m(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))^2}\frac{C_I(t)}{s_m(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))}-\frac{C_I(t)^2X_m^p(t)}{s_m^2(g_I^p(t)+\alpha(g'^p_I-g_I^p(t)))^2}\frac{s_m(g_I^p(t)-g'^p_I(t))}{C_I(t)})))$$

$$+\ 3600[\frac{v_m^p(t-1)C_I(t-1)(g_I^p(t)-g'^p_I(t))}{s_m(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}\ +\ \frac{C_I(t)}{C_I(t-1)}(\frac{g_I^p(t-1)-g'^p_I(t-1)}{g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t))}$$

$$-\ \frac{(g_I^p(t-1)+\alpha(g'^p_I(t-1)-g_I^p(t-1)))(g_I^p(t)-g'^p_I(t))}{(g_I^p(t)+\alpha(g'^p_I(t)-g_I^p(t)))^2}] \qquad (77)$$

The bisection method can be used to find an approximate value of " which sets the above function equal to zero. The optimal solutions are found when the above algorithm runs iteratively until the stopping criterion is met.

There are several stopping criteria that can be applied while implementing the Frank-Wolfe method. For instance, the convergence criterion can be based on the marginal contribution of successive iterations. Alternatively, the algorithm can be terminated if the elements of the gradient vector are close to zero. In some cases, criteria that are based on the change in the variables between successive iterations are used.

Because calculating the relative change of the objective function involves more computation and longer computer running time, the convergence criterion based on the change in the variables between successive iterations is used in this research, that is,

$$\max_i \{|g_i^n - g_i^{n-1}|\} \# g$$

The notation $\max_i \{.\}$ stands for the maximum, over all possible values of i, of the arguments in the braces. In other words, the iterations terminate if the maximum difference between the green times of each phase at the previous and current iterations for all intersections during all time periods is less than or equal to g.

The procedure for solving the real time traffic control model is summarized as follows:

Step 0: Obtain initial solutions from TRANSYT-7F.

Step 1: Find the descent direction.
For every intersection and each time period, assign the green time g' = min g to each phase. Assign the remaining green time to the phase with the highest vehicle delay.

Step 2: Determine the optimal move size.

$$\min Z\,(g + "(g' - g)) \tag{78}$$

$$\text{subject to} \quad 0 \# " \# 1 \tag{79}$$

Use the bisection method of iterative interval reduction to find the value of " which can satisfy $dZ(\text{"})/d\text{"} = 0$

Step 3: Update the green times.

$$g = g + \text{"}(g' - g) \tag{80}$$

Step 4: Test convergence.

If max $\{|g^n - g^{n+1}|\}$ # g , stop. Otherwise, go to Step 1.

A signal phase plan comprises a complete specification of phasing sequence, splits (green time for each phase) and the length of amber intervals. The choice of phasing sequence depends primarily on the treatment of left turn traffic. The length of a cycle and its splits are determined by a number of factors, the main ones are traffic demand patterns and delay to traffic imposed by signals.

While a series of intersections along an arterial streets are treated as a single system and their timing plans are developed together to provide good vehicle progression along the arterial, the reasonable 2-way progression between traffic signals generally requires the same cycle lengths and similar phase patterns at all signals in a network. Cycle offsets for progression between signals must usually be in half-cycles. Although the cycle length and signal phase plan can be varied at each intersection, they have to be specified before initialization and are not modified during the optimization. The same restriction is also applied to the number of time periods and the duration of each time period. The methodology developed in the study may allow constructing additional external looping to optimize cycle length and signal phase sequence. However, such optimization is out of the scope of this study and its performance has not been investigated. Figure 3-1 shows the flow chart of the above solution algorithm.

In Chapter 5, the optimization program for dynamic traffic-responsive signal control is combined with the DTA model described in Chapter 4. A Visual Basic program

developed to implement this flow chart to determine the optimal signal settings, which are then imported into the DTA model, is presented in Chapter 6.

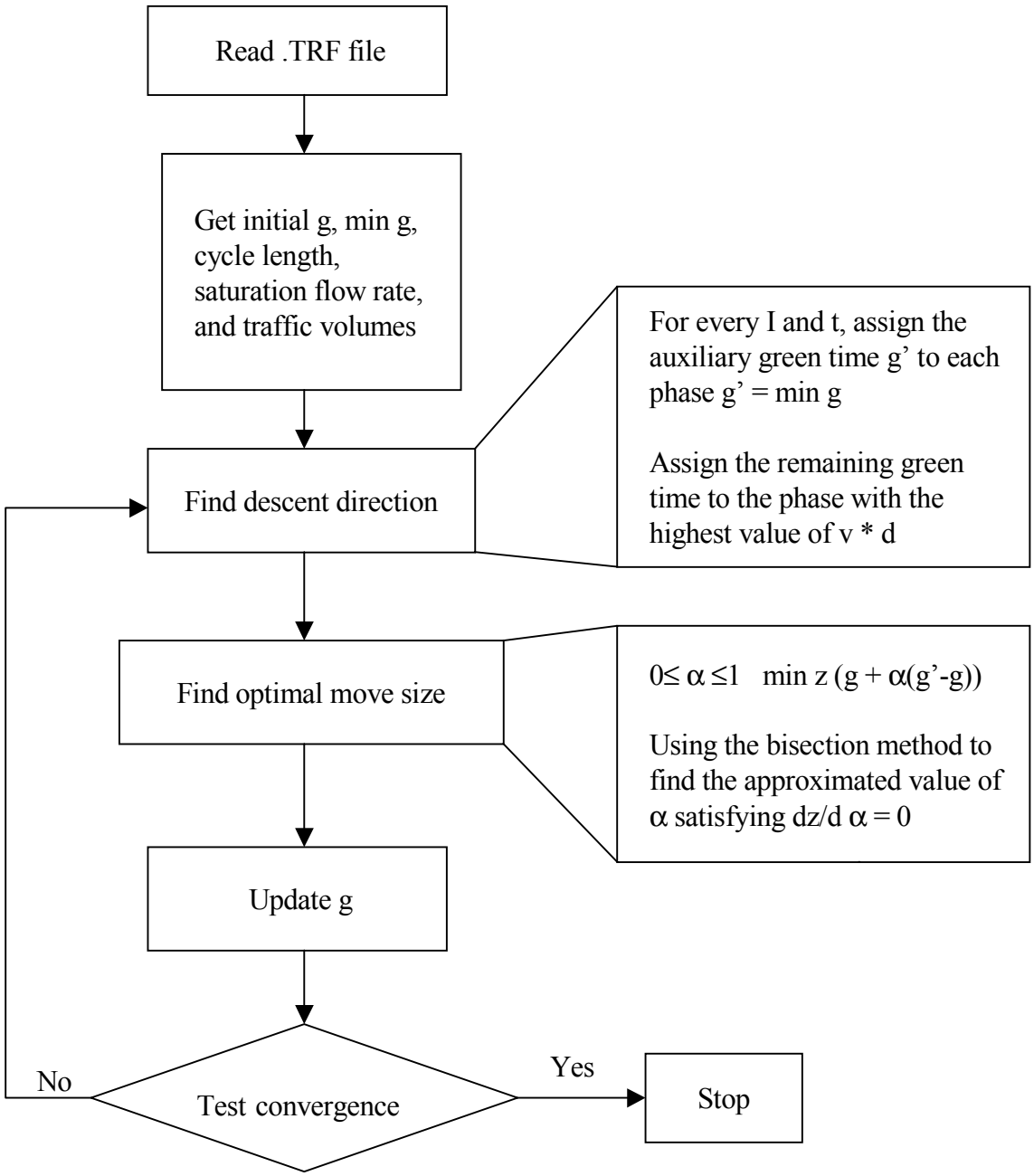

Figure 3-1 Flow Chart of the Solution Algorithm for the Real-Time Traffic-Responsive Signal Control Model.

CHAPTER 4
DYNAMIC USER-OPTIMAL ROUTE CHOICE MODEL

4.1 Introduction

This chapter presents the techniques applied for developing a variational inequality formulation for a dynamic user-optimal route choice model. A variational inequality is a broader formulation for equilibrium problems compared to a mathematical program. The static formulation and the equivalent optimization program are presented first. The formulation is then extended for the dynamic case.

Subsequently, the relaxation algorithm for relaxing the inseparable link cost functions caused by time-dependent link flow is discussed. A good travel time estimate is introduced to relax the link interaction. The equivalent optimization program is presented after the link interaction is relaxed. The last section presents the solution algorithm developed to solve the equivalent optimization program based on the improvement in the travel time estimation.

4.2 Variational Inequality Models

A variational inequality model is a general model formulation that encompasses a set of mathematical models, including nonlinear equations, optimization models and fixed point models [33]. Variational inequalities were originally developed as a tool for the study of certain classes of partial differential equations such as those that arise in mechanics. However, with its capability of formulating and analyzing more general models for

equilibrium conditions than the constrained optimization approach, the basic concept of VI theory has received increasing attention from transportation network modelers during the last decade [5]. The definition and proposition presented in the following section focus on variational inequality models suitable for the analysis of transportation network equilibrium models.

4.2.1 Static Transportation Network Equilibrium Model

Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ be a vector of link traffic volumes and $\mathbf{d}[\mathbf{x}] = [d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_m(\mathbf{x})]$ be a vector of disutility functions. The equivalent variational inequality formulation of a discrete link-based user-optimal route choice model can be summarized as follows from existing literature [4]: the variational inequality model is to determine a control vector $\mathbf{x}^* \in G$ such that

$$\mathbf{d}[\mathbf{x}^*] \cdot [\mathbf{x} - \mathbf{x}^*] \geq 0 \qquad \forall \mathbf{x} \in G \tag{81}$$

where G denotes the feasible region that is delineated by the following three constraints:

1. flow conservation constraints: $\sum_k f^{rs}_k = q_{rs} \qquad \forall r, s$ (82)
2. non negativity constraints: $f^{rs}_k \geq 0 \qquad \forall k, r, s$ (83)
3. flow propagation constraints: $x_m = \sum_{rs} \sum_k f^{rs}_k \delta^{rs}_{m,k} \qquad \forall m$ (84)

According to the Symmetry Principle [34], if the Jacobian matrix $\nabla d(x)$ is symmetric, then the variational inequality above has an equivalent optimization model. In this particular case, where the link travel disutility functions are separable, the above VI formulation is equivalent to the following mathematical programming model:

$$\min\ z(\mathbf{x}) = \sum_m \int_0^{x_m} d_m(\omega)d\omega \tag{85}$$

This program is exactly the same optimization program used to represent the UE condition discussed in Section 2.3. Therefore, the solution algorithm presented in Section 2.3 can be used to solve it. The solution algorithm with the application of the Frank-Wolfe method is summarized in the following:

Step 0: Find initial volumes.

Step 1: Update disutility estimates.

Step 2: Find descent direction. Perform an all-or-nothing traffic assignment.

Step 3: Perform line search for optimal move size.

Step 4: Calculate new solution volumes.

Step 5: Test convergence.

As discussed in Section 2.3, at some point in step 2 involving the procedure for finding the descent direction, the solution algorithm will call for minimizing the total travel time over a network without flow-dependent travel times on each link. The total travel time spent in the network will be minimized by assigning all motorists to the shortest travel time paths connecting their origins to their destinations. Such an assignment is performed by the all-or-nothing network loading procedure. The core of the all-or-nothing procedure is the determination of the shortest paths between all origins and destinations. An efficient method to find these paths between all network nodes can be easily found in existing literature [16]. However, since the traffic assignment feature built into CORSIM 5.1 is used to implement the above algorithm, the methods to locate the shortest path are not introduced here.

The procedure of step 3, performing a line search for optimal move size between the current solution and the auxiliary solution from step 2, involves the bisection method to find

the value of " which can set the derivative of the objective function with respect to " equal to zero. This procedure is also discussed in Section 2.3.

4.2.2 Dynamic Transportation Network Equilibrium Models

Unlike the static network system, the dynamic network system concerns a vector of control variables $\mathbf{x}(\mathbf{t}) = [x_1(\mathbf{t}), x_2(\mathbf{t}), \ldots, x_m(\mathbf{t})]$. The equivalent variational inequality formulation was defined as follows in existing literature [5]: the variational inequality model for the dynamic transportation network equilibrium model is to determine a control vector $\mathbf{x}^*(\mathbf{t})$ O G $(\mathbf{t})$ such that

$$\mathbf{d}\,[\mathbf{x}^*(\mathbf{t})] \text{ @ } [\mathbf{x}(\mathbf{t}) - \mathbf{x}^*(\mathbf{t})] \text{ \$ } 0 \qquad \text{œ } x(\mathbf{t}) \text{ O } G(\mathbf{t}) \tag{86}$$

where G $(\mathbf{t})$ denotes the feasible region that is delineated by the three sets of constraints, which are the same as for the static condition, and another set of flow propagation constraints:

$$\text{in-flow } x_{mk}^{rs}(t) = \text{exit-flow } x_{mk}^{rs}(t + \text{travel time on link m}) \tag{87}$$

The link interactions caused by the above flow propagation constraints result in an asymmetrical Jacobian matrix of the dynamic travel disutility functions such that the Symmetry Principle cannot be applied to the reformulated VI format as an optimization program. However, there is an iterative method (relaxation algorithm) that can solve the problem. Assuming that there exists a vector of smooth auxiliary functions $\mathbf{g}(\mathbf{x}, \mathbf{y})$ with suitable properties, Ran and Boyce [5] proved that at iteration n, solving the following variational inequality model:

$$g(x^{(n)}, \mathbf{x}^{(n-1)}) \text{ @ } (\mathbf{x} - \mathbf{x}^{(n)}) \text{ \$ } 0 \qquad \text{œ } x \text{ O } G \tag{88}$$

is equivalent to solving the corresponding mathematical programming model:

$$\min\ Z(\mathbf{x}, \mathbf{x}^{(n-1)}) \qquad \text{œ } x \text{ O } G \tag{89}$$

Chen and Hsueh [12] applied the above relaxation algorithm and suggested relaxing the link interactions by calculating the link travel times based on the traffic volumes from the assignment result at the previous iteration. This procedure resulted in a model with a positive-valued diagonal Jacobian of $\mathbf{d}[\mathbf{x}(\mathbf{t})]$. The following mathematical programming model is then equivalent to the relaxed variational inequality model:

$$\min\ z = \sum_{t}\sum_{m} \int_{0}^{x_m^{(n)}(t)} d_m(x_m^{(n-1)}(1),\ x_m^{(n-1)}(2),\ \ldots,\ x_m^{(n-1)}(t-1),\ \omega)d\omega \qquad (90)$$

A nested diagonalization algorithm applying the relaxation (diagonalization) method is then:

Step 0: Find an initial feasible solution.

Step 1: Estimate link travel times $T_m(t)$.

Step 2: Solve the optimization model (Eq. (90)) based on $T_m(t)$.

Step 2.1: Update the $x_m(t)$ based on $T_m(t)$.

Step 2.2: Solve the optimization model (Eq. (90)) by the FW method.

Step 2.3: Test convergence.

Step 3: If $d_m(t)$. $T_m(t)$, stop. Otherwise, set $T_m(t)^{n+1} = T_m(t)^{n} + (1/n)(d_m(t)^{n} - T_m(t)^{n})$, and go to step 2.

The optimization program in Eq. 90 has many similarities with the one representing the static case. Therefore, step 2 is the typical procedure for solving the static route choice model with asymmetric travel time functions which are dependent on the traffic flows from the previous iteration. In step 2.1, the initial solution is updated every time. In other words, the traffic patterns resulting from the previous iteration in step 2.2 cannot be carried over into the next iteration because different link travel times construct different subproblem

feasible regions. In step 3, the estimated link travel times are updated using the method of successive averages to predetermine the move size of 1/n.

The estimated link travel times were defined using the following setting: $T_m(t) = NINT\,[d_m(t)]$. The notation *NINT* indicates the round-off arithmetic operation: set $T_m(t) = i$, if $i - 0.5 \# T_m(t) \# i + 0.5$. Although finding the link travel time is a critical issue for solving the above algorithm, Chen and Hsueh [12] simply used the above round-off operation to estimate it and claimed to leave more precise estimation to further research. Improvements for this issue are discussed in the following section.

4.3 Enhanced Solution Algorithm

One of the most important analytical tools of traffic engineering is computer simulation. Computer simulation is more practical than a field experiment because it is less costly and the results are obtained quickly. CORSIM 5.1, the standard simulation model, uses a microscopic stochastic simulation model to represent the movements of individual vehicles, which includes the influences of driver behavior. Therefore, CORSIM 5.1 is an efficient tool to simulate the utilization of transportation resources and develop precise measures to a transportation system's operational performance.

In this study, CORSIM 5.1 is used to simulate the traffic conditions at each iteration given the traffic volumes assigned in the previous iteration. The actual travel time on each link is then calculated from the simulation results.

Moreover, since CORSIM 5.1 provides a user-equilibrium platform to perform traffic assignments, and the program internally translates the origin-and-destination data into a form suitable for use by its simulation model, it is adopted not only to calculate the link travel times but also to perform the traffic assignment procedure. Adopting CORSIM 5.1 into the

solution algorithm improves both the computational efficiency and the quality of the solution.

Although CORSIM 5.1 can only deal with the static case, an important feature of CORSIM 5.1 is that characteristics that change over time, such as signal timings and traffic volumes, can be represented by dividing the simulation into a sequence of user-specified time periods during which the traffic flows, traffic controls and geometry are held constant. This feature is especially useful for solving the optimization program in Eq. 90 because the procedure for solving the static route choice model with asymmetric travel time functions has to be performed for each time period separately in each iteration.

The solution algorithm adopting CORSIM 5.1 is illustrated and explained as follows.

Step 0: Assign traffic initially by user-equilibrium assignment using CORSIM 5.1 to obtain traffic volumes $x_m^{(0)}$.

Step 1: Simulate the traffic conditions based on $x_m^{(n)}(t)$ and calculate the link travel times $T_m^{(n)}(t)$.

Step 2: Solve the optimization model in Eq. 90 based on $T_m^{(n)}(t)$.

Step 2.1: Calculate congested speeds based on $T_m^{(n)}(t)$.

Step 2.2: Assign traffic using CORSIM 5.1 to obtain new $x_m^{(n)}(t)$ for each time period sequentially.

Step 2.3: Compute $d_m^{(n)}(t)$ based on new $x_m^{(n)}(t)$

Step 3: Test Convergence. If max $\{|\mathbf{d}^n - \mathbf{T}^n|\}$ # g, stop. Otherwise, update $x_m^{(n)}(t)$ and then go to Step 1.

In step 0, the initial assignment is performed by user-equilibrium assignment using CORSIM 5.1. The analysis time interval is divided into user-specified time periods but each time period is assumed to have the same link volumes initially.

In step 1, CORSIM 5.1 is used to simulate the traffic conditions for all of the time periods sequentially given assigned traffic volumes either from step 0 (each time period has the same traffic volumes) or from step 3. CORSIM's simulation output can present the average speed of travel on each link. The average travel time on each link can be obtained by dividing the link length by the average speed.

In step 2.1, the congested speed on each link, based on the link travel times obtained from step 1, is updated. As discussed in the Section 4.2.2, the travel time function in the above optimization program is dependent on the traffic volumes assigned in the prior iteration. Therefore, while performing the static traffic assignment, the travel time function has to apply traffic volumes from the prior iteration.

CORSIM 5.1 provides two travel time functions, the BPR and the modified Davidson link impedance functions, which were explained in Section 2.9. Users are allowed to choose one of the impedance functions, but are not allowed to modify them. To deal with this shortcoming, the CORSIM 5.1 "free-flow speed" on each link has to be adjusted based on traffic volumes from the previous iteration to consider the transportation network as not being empty for the assignment performed in step 2.2.

Step 2.2 performs a traffic assignment using CORSIM 5.1 to obtain new traffic volumes for each time period sequentially. As mentioned above, minimizing the optimization program in Eq. 90 is to minimize the static route choice model with asymmetric travel time functions for each time period separately.

In step 2.3, the link travel times from the travel time function used in the traffic assignment model based on traffic volumes from step 2.2 are computed. The relaxation algorithm relaxes the link interactions by assuming the link travel times based on the traffic

volumes from the previous iteration are known from the first loop (outside steps 2.1- 2.3) of the solution procedure. Therefore, the link travel times calculated from the travel time function built into the assignment model in the second loop operation have to converge to the estimated link travel times in the first loop. This convergence is tested in step 3. Since CORSIM's assignment output includes the average link speed according to the travel time function, the average travel time based on the travel time function can be obtained also by dividing the link length by the average speed.

The remaining algorithmic step is to perform updating and convergence testing. The solution procedure is performed iteratively until the stopping criterion is met. Figure 4-1 shows the flow chart of the above solution algorithm.

In Chapter 5, the above DTA model is combined with the optimization program for the real-time traffic control model developed in Chapter 3. The Visual Basic program presented in Chapter 6 is developed to implement the process in this flow chart to determine the user-equilibrium traffic flows on each link for each time period given the signal settings from the real-time traffic control model.

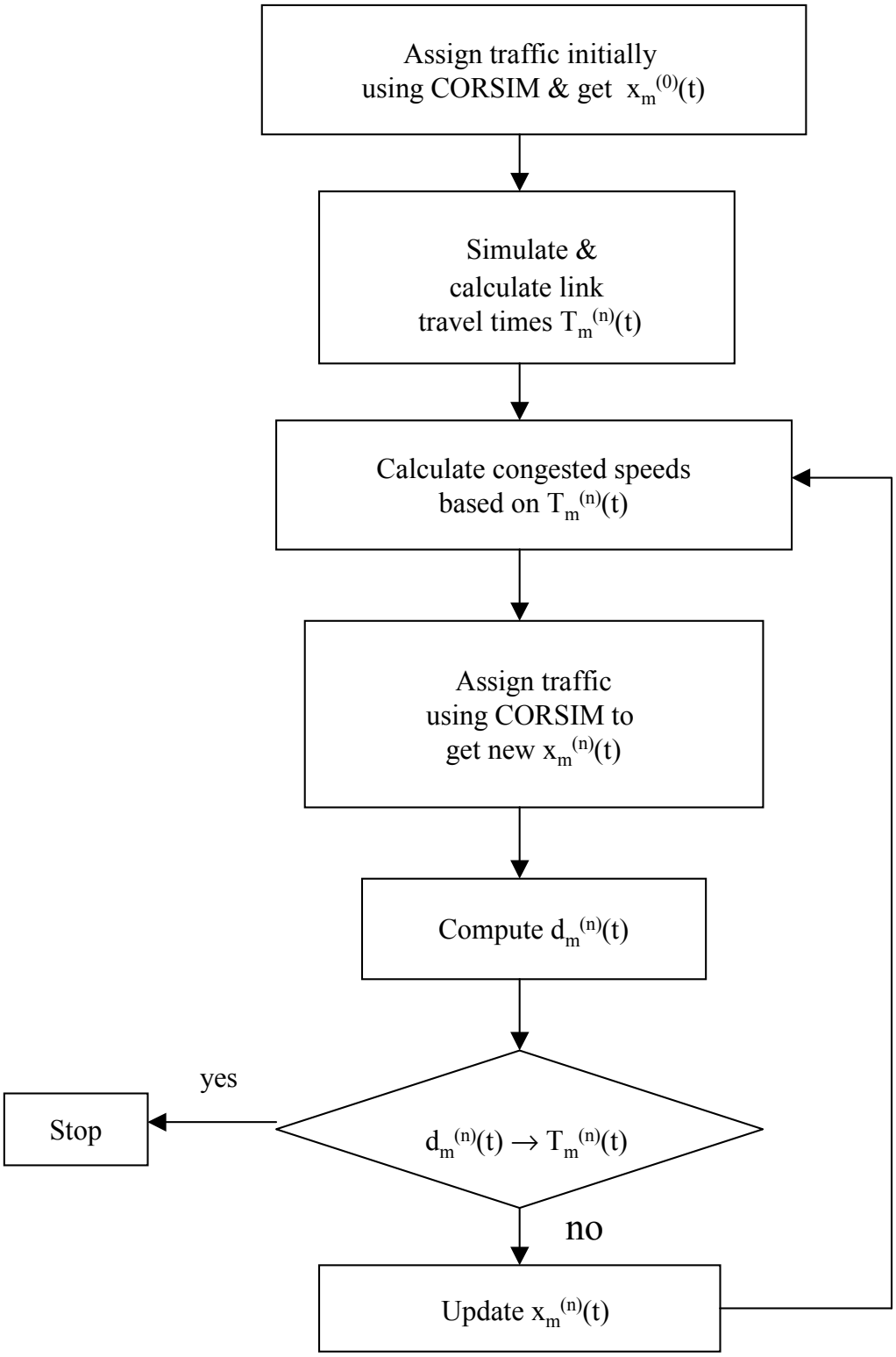

Figure 4-1 Flow Chart of the Solution Algorithm for the DTA Model.

CHAPTER 5
COMBINED MODEL

5.1 Introduction

This chapter presents a theory to deal with the traffic flow equilibrium which involves traffic-responsive signal control policies. This situation has been extensively investigated in two different ways [11]: global optimization models and the iterative optimization assignment (IOA) procedure. However, global optimization models have major difficulty in finding an efficient solution algorithm for calculating the optimal signal settings in traffic networks while anticipating user-optimum equilibrium flows.

Some authors [9, 12, 27] have used a bilevel programming method to combine the traffic control and the transportation assignment models. This presumes that decision makers at two levels act in a hierarchical manner. At the upper level, decision makers, bound by the decisions of the lower level, try to maximize the their own profit, taking into account the reactions of the lower level accordingly. The iterative optimization assignment procedure can be used to solve a bilevel program.

The procedure of the IOA is to update the signal settings for fixed flows and solve the traffic equilibrium assignment for fixed signal settings sequentially until the solutions of the two models are considered to be consistent. Yang [11] claimed that the IOA procedure has the advantages that the traditional traffic assignment and signal setting techniques can be employed to solve the problem and can be applied to a large network.

Marcotte [28] used the IOA procedure to solve for optimal signal settings based on capacity variables corresponding to user-optimized equilibrium traffic flows. Marcotte hoped that the convergence toward a 'good' solution would be obtained. He concluded that the numerical experiments tended to show the IOA procedure yielded near-optimal solutions.

In the following sections, the bilevel program is presented first. The iterative optimization assignment procedure is then addressed.

5.2 Bilevel Model

The bilevel programming model described formally and completely by Bard and Falk [35] can be formulated as:

$$\underset{g \in G}{\text{Min}} \quad Z_1(g,v) \tag{91}$$

subject to

$$v \in \left\{ \underset{z \in V(g)}{\text{argmin}} \; Z_2(g, z) \right\} \tag{92}$$

where g and v represent the decision vectors associated with the upper and lower levels, respectively. G is the feasible set of the g-variables and V(g) is the feasible set, possibly dependent on g, of the v-variables [28]. The "argmin" of a mathematical program is the optimal solution of the program; in other words, $Z_1(\cdot)$ has to be evaluated at the optimal solution of $Z_2(\cdot)$. Since the objective of traffic management systems is to calculate equilibrium flows which are consistent with a given traffic-responsive control policy, this framework can be applied to integrate the traffic control policy and the transportation assignment procedure. In the upper level, $Z_1(\cdot)$ represents the optimization model for dynamic traffic control in which total intersection delay in a network is minimized by allocating appropriate green times, g, given the user-optimal traffic flows, v, which are

obtained from the lower level model. In the lower level, $Z_2(\cdot)$ represents the dynamic traffic assignment model formulated as a variational inequality formula in which the link traffic flows, z, reach user equilibrium, v, given the link travel times gathered from the upper model. The mathematical formulation of the dynamic transportation management system, which combines the models developed in Chapter 3 and Chapter 4, is given as follows:

$$\min_{\mathbf{g}\in G} Z(\mathbf{g}, \mathbf{v}) = \sum_{t} \sum_{I} \sum_{p} \sum_{m\in B(I)} v_m^p(t)\ d_m^p(t) \tag{93}$$

$$\text{subject to} \quad \mathbf{T}[\mathbf{v}(\mathbf{t})] \cdot [\mathbf{z}(\mathbf{t}) - \mathbf{v}(\mathbf{t})] \geq 0 \qquad \forall\, \mathbf{z}(\mathbf{t}) \in \mathbf{V}(\mathbf{g}) \tag{94}$$

where G denotes the feasible region that is delineated by the two constraints for conserving the cycle length and the minimum green times shown earlier in Eqs. 66 and 67. The vehicle delay function, d, is dependent on the variables of green times, **g**, and traffic flows, **v**.

To calculate the total delay happening at each intersection, the traffic flows, **v**, in the upper level have to be specified as traffic volumes moving on each link during a specific phase, at a specific time, that is, $v_m^p(t)$. However, these traffic flows have to satisfy the VI formula in the lower level in which the total volumes on each link are represented in vector form, $\mathbf{v}(\mathbf{t})$. $\mathbf{T}[\mathbf{v}(\mathbf{t})]$ denotes the travel disutility functions which are dependent on link traffic flows whereas $\mathbf{V}(\mathbf{g})$ is the feasible traffic flow set delineated by the flow constraints presented in Eqs. 82-84 in Chapter 4. One of the flow constraints, the flow propagation constraint, is dependent on the link travel times which are affected by the green time splits.

5.3 Iterative Optimization and Assignment Procedure

Since there is not much hope to develop exact solution algorithms for large or even medium-size networks [28], the IOA procedure is used to solve the above bilevel program.

The iteration of the algorithm consists of solving sequentially an optimization model involving the signal timing variables, with the flow variables fixed, and a user-optimized equilibrium model corresponding to this new green time vector. The procedure for iterative optimization and assignment is illustrated in Figure 5-1.

Although there are limitations, Smith and Vuren [36] proved the convergence of the iterative optimization and assignment algorithm. Using numerical experiments, Marcotte [28] also showed the IOA heuristic can yield near-optimal solutions.

5.4 Detailed Procedure

The two submodels included in Figure 5-1, the adaptive traffic-responsive signal control submodel and the dynamic traffic assignment submodel, can be replaced by the flow charts shown in Figure 3-1 and Figure 4-1, respectively. To initialize the procedure, users need the information of the link-node network structure and the O-D demand matrix, and must decide on the number of time periods, the duration of each time period and signal settings. The detailed procedure for implementing the IOA method to optimize the dynamic transportation management system is described as follows:

Step 1: Initialize.

Step 1.1: *Prepare input data.* Prepare the link-node network connection structure, initial travel speed estimates, link length measurements, link capacities, the O-D trip matrix, the number of time periods, the duration of each time period, and for each intersection, the signal cycle length, cycle offset, and signal phase plan.

Step 1.2: *Perform static traffic assignment*. The traditional user-equilibrium traffic assignment method is used to assign the O-D trip matrix to the network using CORSIM 5.1 based on initial estimates of travel times.

Step 1.3: *Determine initial signal settings and intersection delays*. The static network-wide optimal signal timing settings and delays for the user-specified phase plans based on the traffic flows obtained from Step 1.2 are determined by TRANSYT-7F.

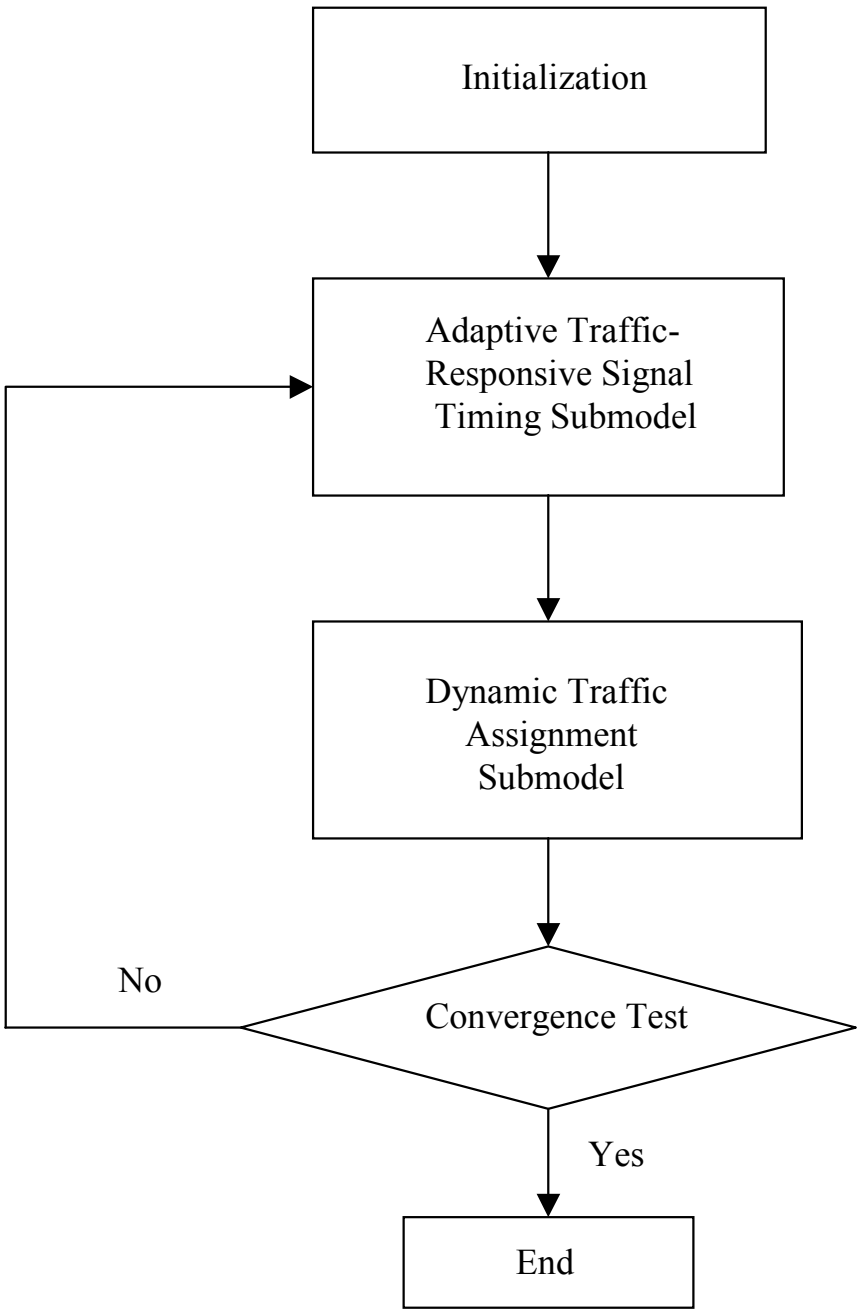

Figure 5-1 Flow Chart of the Iterative Optimization Assignment Procedure.

Step 1.4: *Perform initial flow simulation*. The total analysis time interval is split into user-specified time periods. Each time period has the same signal timing settings obtained from Step 1.3. With the link traffic flows obtained from Step 1.2 at the initial time period, the CORSIM 5.1 simulation model simulates the traffic conditions and obtains the link traffic flows for each time period. The queue buildups from each time period will carry into the next time period. The signal timings and intersection delays can then be calculated for the revised traffic volumes.

Step 2: Execute the adaptive traffic-responsive signal control submodel.

Step 2.1: *Obtain input data.* For the first iteration, read the signal timing settings and delays from TRANSYT-7F (Step 1.3) and the link traffic flows from the CORSIM 5.1 simulation results (Step 1.4) for each time period. Starting from the second iteration, read the optimal signal timing settings and delays resulting from the previous iteration and read the link traffic flows for each time period from the dynamic traffic assignment submodel.

Step 2.2: *Find descent direction.* Assign the minimum green time to each phase at each intersection for each time period. Then assign the remaining green time from the cycle length of the intersection to the phase with the highest vehicle delay.

Step 2.3: *Optimize move size.* Perform a line search to find the optimal extent value to move from the last solution of green times toward the auxiliary solutions obtained form Step 2.2.

Step 2.4: *Test convergence.* If the convergence criteria for green times are not satisfied, return to Step 2.2; otherwise, proceed to Step 3.

Step 3 Execute the dynamic traffic assignment submodel.

Step 3.1: *Perform simulation.* The optimal signal timing settings for each time period are updated using the results from the adaptive traffic-responsive control submodel (Step 2). Using the link traffic flows obtained from Step 1.2 for the initial time period at the first iteration (or using the link traffic flows obtained from the previous iteration for the second iteration and beyond), the simulation model simulates traffic conditions based on dynamic traffic-responsive signal timings to obtain the link traffic flows for each time period.

Step 3.2: *Calculate link travel times.* Calculate the link travel times for each time period by dividing the link length by the average speed from the CORSIM 5.1 simulation results of Step 3.1.

Step 3.3: *Modify free-flow speeds.* Modify the free-flow speed on each link for each time period to transfer the effects of the traffic flows resulting from the

previous iteration to the travel time functions built into the static traffic assignment software (CORSIM 5.1).

Step 3.4: *Perform static traffic assignment for each time period separately.* The static traffic assignment is performed for each time period separately with congestion speeds represented as modified free-flow speeds.

Step 3.5: *Compute estimated link travel times based on the travel time functions.* The estimated link travel times based on the travel time functions built into the traffic assignment software are calculated.

Step 3.6: *Test convergence.* Compare the links travel times resulting from Step 3.2 and Step 3.5. If the convergence criteria are not satisfied, return to Step 3.1; otherwise, proceed to Step 4.

Step 4 Test convergence. Compare the traffic flows between two iterations. If the convergence criteria, max $\{|\mathbf{v}^{n} - \mathbf{v}^{n+1}|\}$ # g, are not satisfied, return to Step 2; otherwise, stop.

CHAPTER 6
IMPLEMENTATION

6.1 Introduction

Software implementation is a necessary step in optimizing the transportation management system. The objective of this chapter is to develop computer software to implement the detailed procedure presented in Section 5.4. A numerical example is used to illustrate the procedure for implementing the IOA method to optimize the dynamic transportation management system.

6.2 Numerical Example

Figure 6-1 shows the link-node structure of a traffic network illustrated by the ITRAF display. ITRAF is a preprocessor for CORSIM, which allows users to create traffic networks, enter traffic volumes and timing data on a map window and then transfer all information into input files for CORSIM. There are eight roadways with eleven intersections in the example network.

Most numerical examples used in the existing literature were small traffic networks because of the complicated computation and the limitation of computer memory to store large link-node structures. The computer program developed in this research was written in Visual Basic for Windows 2000. With their increasing power, microcomputers have become a suitable hardware platform for rather large network processing applications. However, there are size limitations for CORSIM 5.1 network characteristics.

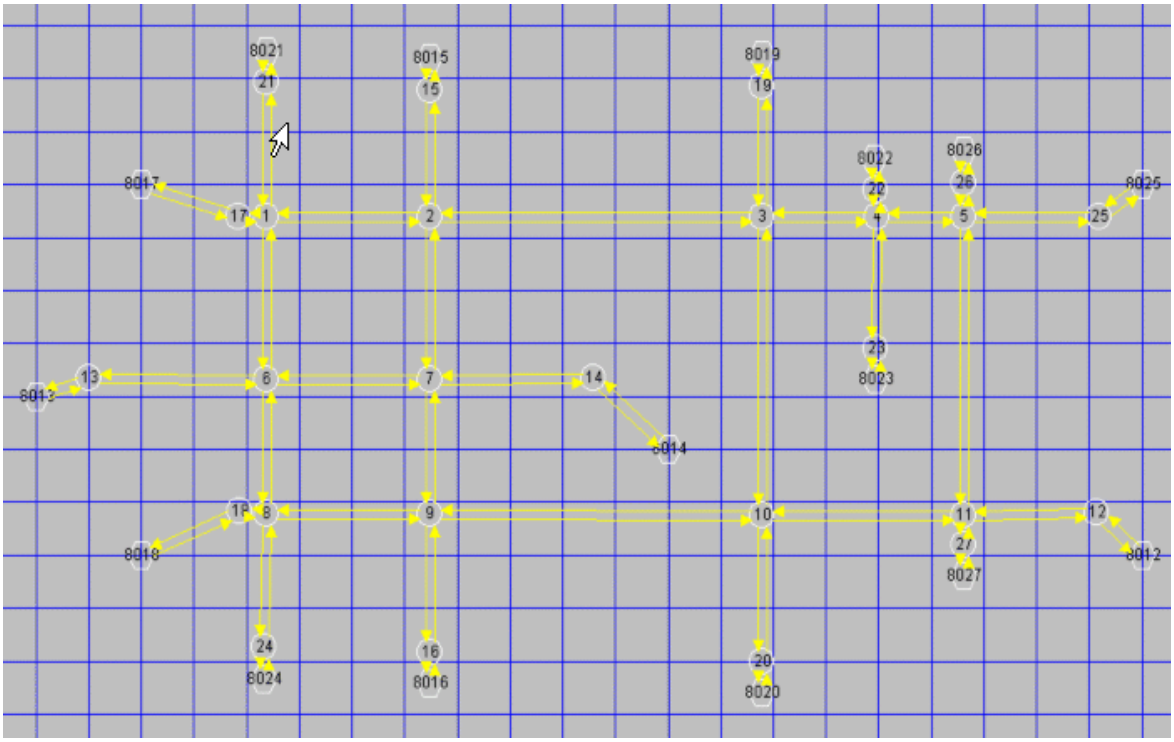

Figure 6-1 Link-Node Structure of the Example Network.

In CORSIM 5.1, the entry and/or exit nodes must be numbered between 8000 and 8999 while the internal nodes can be numbered from 1 to 6999. In other words, the latest version of CORSIM 5.1 allows a maximum number of 1,000 entry and/or exit nodes. Entry and/or exit nodes were used in this study to represent the centroids of traffic analysis zones (TAZ). The TAZ is the basic analysis unit for calculating trip productions and attractions. These trips are assumed to start or end at the centroid of a zone and are distributed throughout the network to create the O-D matrix. In Florida, the numbers of TAZs for Palm Beach County, Broward County and Miami-Dade County are 1,118, 892 and 1,466, respectively. Therefore, for urban areas with less than 1,000 TAZs, CORSIM 5.1 can accommodate the origin and destination zones without adjustment. However, for urban areas with more than 1,000 TAZs, some TAZs have to be merged to reduce the number of zones to less than 1,000.

The maximum number of internal nodes is 6,999 in CORSIM 5.1. Internal nodes are used to represent intersections of a given street network. The numbers of non-centroid nodes (representing intersections of streets) in the 1999 networks for Palm Beach County, Broward County and Miami-Dade County are 2,861, 3,378 and 5,969, respectively. For a street network with more intersections than 6,999, some minor streets have to be excluded to reduce the total number of intersection to be less than the maximum number of internal nodes allowed in CORSIM 5.1.

Table 6-1 shows the O-D demand matrix for the numerical example. The number of trips per hour for a specific trip purpose (e.g., working or shopping) shown in the cells were defined for demonstration purposes. For example, in Table 6-1, there were 100 shopping trips per hour from TAZ 8021 to TAZ 8019.

Table 6-1 O-D Demand Matrix for the Example Application

	Destination										
Origin	8021	8015	8019	8023	8025	8013	8014	8024	8016	8020	8012
8021	0	50	100	30	110	60	100	100	200	50	110
8015	20	0	120	150	50	50	50	50	50	60	110
8019	90	50	0	40	140	100	150	200	150	100	150
8023	50	100	120	0	110	110	120	40	50	30	40
8025	50	100	100	50	0	50	50	100	100	50	50
8013	80	100	100	130	70	0	100	60	110	60	50
8014	50	100	150	100	150	150	0	80	100	80	50
8024	50	100	50	20	200	110	50	0	200	40	50
8016	30	130	100	150	80	130	50	130	0	50	100
8020	20	120	150	130	150	120	50	120	90	0	100
8012	60	70	60	120	110	130	90	80	70	50	0

Note that nodes 8017, 8018, 8022, 8026 and 8027 shown in Figure 6-1 do not have volumes in the O-D matrix. These nodes were treated like dummy nodes and can be connected to new nodes in the future.

Figure 6-2 shows the pre-timed signal settings which were defined at each intersection. It consists of four phases for each signal cycle with a protected left-turn phase and a through movement phase for both north-south and east-west directions. As discussed in Section 3.4, the signal settings can be different at each intersection. However, to simplify the implementation procedure, the signal settings for all of the intersections in the example network were the same. In addition, the phase durations for the north-south and east-west directions were originally set to be equal.

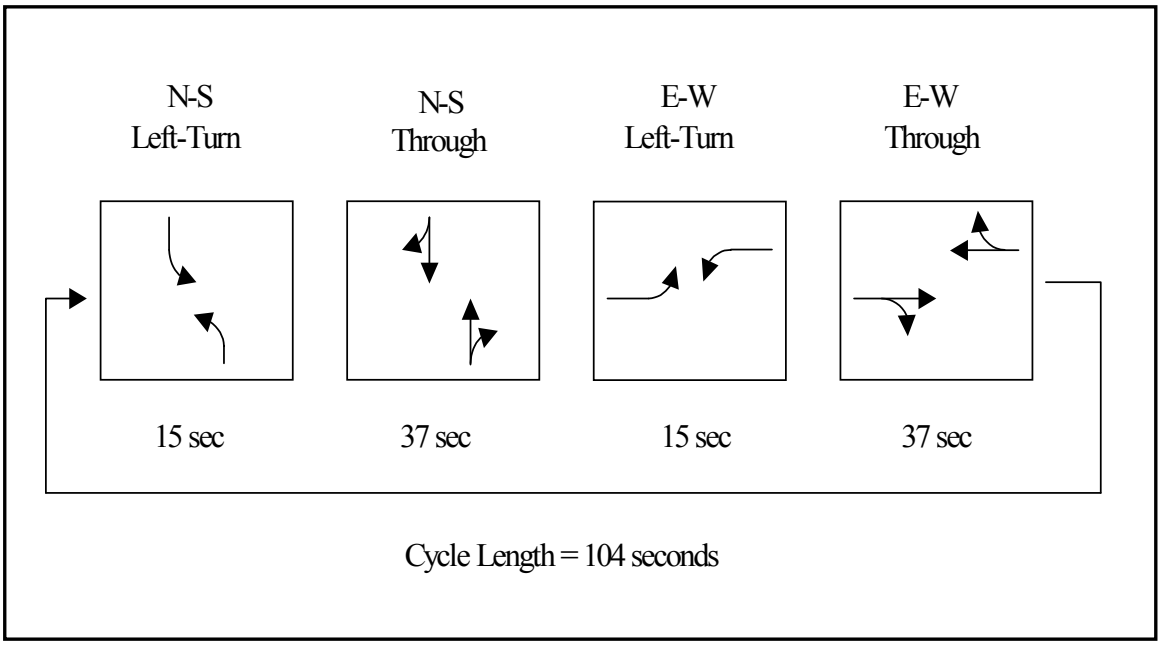

Figure 6-2 Initial Signal Settings for the Example Application.

6.3 Software Implementation

A comprehensive, menu-driven software program was developed in Visual Basic to implement the detailed procedure presented in the previous chapter. Figure 6-3 shows the program's main menu. A copy of the program can be downloaded from the following web site: http://www.fiu.edu/~chowl/. The application of the software is illustrated in the following procedure following the steps in Section 5.4:

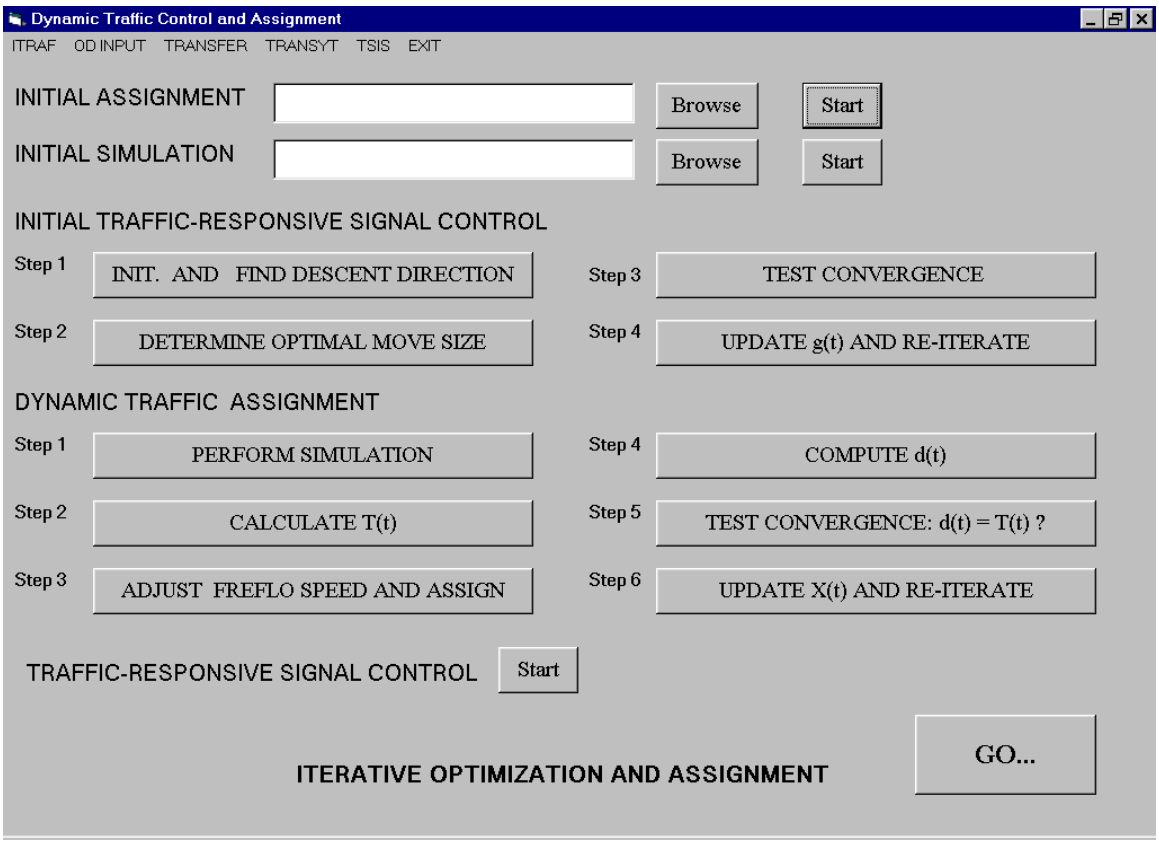

Figure 6-3 Main Menu Screen.

Step 1 Initialization:

Step 1.1: *Prepare input data*.

1. The **ITRAF** button on the menu bar launches the application of ITRAF and allows users to create traffic networks, enter traffic volumes and timing data on a map window and then transfer this information into input files for CORSIM 5.1. The required input information at this step includes:
   (1) link-node geometric data
   (2) length of each link
   (3) number of lanes on each link
   (4) mean value for number of vehicles in initial queues and start-up lost time
   (5) free-flow speed on each link
   (6) number of approaches and signal phase sequences for each node
   (7) durations of signal timing intervals for each node
   (8) cycle offset for each node

2. The **O-D Input** button on the menu bar serves as an O-D matrix file manager and allows users to create or modify O-D matrix files in Excel spreadsheets.

3. The **Transfer** button on the menu bar transfers the O-D demand matrix, as created or modified using the **O-D Input** option, into CORSIM 5.1 input files. Users can

specify the following assignment parameters through ITRAF if the default values provided by the software are not satisfactory:
(1) acceptable threshold of objective function (default value: 0.1%)
(2) maximum number of iterations (default value: 5)
(3) impedance function type (default type: BPR formula)
(4) parameters of impedance function
(5) capacity smoothing factor (default value: 0)
(6) line-search accuracy threshold (default value: 0.1%)

Step 1.2: *Perform static traffic assignment*.

The **Browse** button at the **Initial Assignment** window invokes the Open File dialog box. With this dialog box, users can select the desired CORSIM 5.1 input file, with the O-D information embedded, and then click the **Start** button to activate the assignment model provided by CORSIM 5.1. The static user-equilibrium traffic is applied to the network using the specified origin-destination information. For the first iteration, the link impedances are evaluated for free-flow speed conditions throughout the entire network. An intermediate solution for each iteration is obtained using link impedances produced by the previous iteration. To obtain an optimal solution for each iteration, an iterative line search is applied to the range between the current intermediate solution and the previous iteration solution. The search terminates when the contribution of the current iteration is less than the accuracy threshold value. The traffic assignment process terminates when the relative change of the objective function between two successive iterations is less or equal to the threshold value.

The Visual Basic (VB) program extracts and formats part of CORSIM's output file and shows the link turning volumes on the screen when the procedure is completed. Figure 6-4 shows an excerpt of initial assignment results for the example.

Traffic Assignment Results

Link	Left Turn	Through	Right Turn
(8025, 25)	0	700	0
(25, 5)	106	594	0
(1, 6)	230	460	141
(1, 2)	118	685	95
(1, 21)	0	500	0
(8021, 21)	0	910	0
(21, 1)	357	553	0
(8024, 24)	0	870	0
(24, 8)	0	493	377
(6, 8)	410	480	0
(6, 13)	0	1010	0
(6, 7)	185	317	281
(6, 1)	0	173	540
(8013, 13)	0	860	0
(13, 6)	288	352	221
(8014, 14)	0	1010	0
(14, 7)	289	423	298

Figure 6-4 Excerpt of the Initial Assignment Results for the Example Application.

Step 1.3: *Determine initial signal settings.*

The **TRANSYT** option on the menu bar launches the application of T7F9, an interface shell program for TRANSYT-7F, and allows users to input the initial input data specified in Step 1.1.1 and the assignment results from Step 1.2 into the input files of TRANSYT-7F. The T7F9 shell program can also be used to access TRANSYT-7F to get the static optimal green times. The timing data from TRANSYT-7F can then be imported into CORSIM's format by the **ITRAF** button.

Step 1.4: *Perform initial flow simulation.*

The **Browse** button at the **Initial Simulation** command invokes the Open File dialog box. With this dialog box, users can select the CORSIM 5.1 input file from Step 1.3. After clicking the **Start** button, users are asked to input the number of time periods and the duration of each time period before the simulation model is activated. In the example, an analysis time interval of one hour was selected and separated into six time periods with a duration of ten minutes for each time period. CORSIM 5.1 allows users to partition the simulation time into a series of time periods (up to 19 time periods) of varying duration. The VB program modifies the CORSIM 5.1 input file to specify the number of time periods as well as the duration of each period. With the link traffic flows obtained from Step 1.2 at the initial time period, the CORSIM 5.1 simulation model simulates the traffic conditions and obtains the link traffic flows for each time period.

Step 2 Execute the initial dynamic traffic-responsive signal control submodel.

Step 2.1 & Step 2.2: *Obtain input data and find descent direction.*

The **Init and Find Descent Direction** button activates the VB program to read the signal timing settings from Step 1.3 and the link traffic flows for each time period from Step 1.4 after users input the minimum green time for each intersection. The VB program then calculates delay for each phase at each intersection at each time period according Eqs. 63-65. The descent direction is obtained by assigning the minimum green time to each phase at each intersection for each time period and then assigning the remaining green time from the cycle length of the intersection to the phase with the highest vehicle delay. Figure 6-5 shows an excerpt of the results of the auxiliary green times.

Step 2.3: *Optimize move size.*

The **Determine Optimal Move Size** button activates the VB program to find the optimal move size from the last set of green times to the auxiliary set of green times. This is achieved by using the bisection method to find an approximate value of " which sets the complicated derivative of the objective function (Eq. 74) equal to zero.

Step 2.4: *Test convergence.*

The **Test Convergence** button executes a VB command to examine the criteria for assessing convergence. A message box on the screen shows whether the criteria are satisfied or not.

```
Time Period: 1
                 AUXILIARY GREEN TIME AT EACH PHASE

NODE         N-S LT      N-S TH     E-W LT     E-W TH
 1            22          21.6       33.6       42.8
 6            14.8        40.4       14.4       50.4
 8            16.4        19.2       16.8       67.6
 2            25.6        25.6       22         46.8
 7            14.4        56.8       17.6       31.2
 9            25.2        43.2       12.4       39.2
 3            20.8        41.6       20         37.6
 10           12.4        43.2       21.2       43.2
 4            14.4        37.4       49.8       18.4
 5            8           89.2       9.6        13.2
 11           11.2        83.6       8.8        16.4

Time Period: 2
NODE
 1            26.8        20.8       29.2       43.2
 6            21.6        42         20.8       35.6
 8            21.2        18         16.8       64
 2            22.4        28.4       34.4       34.8
 7            16.4        48.4       19.2       36
 9            21.6        43.6       15.2       39.6
```

Figure 6-5 Excerpt of the Auxiliary Green Time Results.

The **Update g(t) and Re-iterate** button automatically runs the traffic-responsive signal control submodel (Step 2.1 to Step 2.3) repeatedly until the criteria for convergence are satisfied. Figure 6-6 shows an excerpt of the results for the optimal green times at the initial iteration.

Step 3: Execute the dynamic traffic assignment submodel.

Step 3.1: *Perform simulation.*

The **Perform Simulation** button activates the VB program to update the signal timing settings for each intersection at each time period using the results from the dynamic traffic-responsive signal control submodel (Step 2). The simulation model (CORSIM 5.1) is then activated to simulate the traffic conditions for the network with the optimal signal settings and the traffic flows from the initial assignment (Step 1) or the last iteration.

Step 3.2: *Calculate link travel times.*

The Calculate T(t) button calculates the link travel times from the simulation results. The VB program calculates the link travel times for each time period by dividing the link lengths by the average speeds from the CORSIM 5.1 simulation results of Step 3.1. Figure 6-7 shows an excerpt of results of the link travel times for each time period.

Steps 3.3 and Step 3.4: *Modify free-flow speeds and perform static traffic assignment for each time period separately.*

```
Time Period: 1
                    GREEN TIME AT EACH PHASE

NODE         N-S LT      N-S TH     E-W LT     E-W TH
 1            26          24         25         45
 6            16          43         18         43
 8            22          26         20         52
 2            26          31         24         40
 7            17          49         20         35
 9            20          44         15         41
 3            22          34         22         43
 10           15          45         17         43
 4            18          49         32         21
 5            12          60         21         26
 11           14          58         12         36

Time Period: 2
NODE
 1            27          24         24         45
 6            17          44         19         40
 8            24          25         20         50
 2            25          32         25         37
 7            16          47         21         36
 9            19          44         16         41
```

Figure 6-6 Excerpt of the Optimal Green Time Results at the Initial Iteration.

```
TIME PERIOD  2

  Link          LENGTH (ft)    AVERAGE SPEED (MPH)   TRAVEL TIME
(sec)

(  25,   5)        1277              20                   43
(   1,   6)        1533              13                   80
(   1,  17)        272               24                   7
(   1,   2)        1552              10                   105
(   1,  21)        1269              25                   34
(  21,   1)        1269              10                   86
(  24,   8)        1256              11                   77
(   6,   8)        1268              11                   78
(   6,  13)        1689              26                   44
(   6,   7)        1552              10                   105
(   6,   1)        1533              10                   104
(  13,   6)        1689              10                   114
(  14,   7)        1552              10                   105
(   8,  24)        1256              26                   32
(   8,  18)        255               25                   6
(   8,   9)        1552              10                   105
(   8,   6)        1268              10                   86
(  18,   8)        255               10                   17
```

Figure 6-7 Excerpt of the Link Travel Time Results Based on the Simulation.

The **Adjust Free Flow Speed and Assign** button activates the VB program to replace the free-flow speeds (specified on Entry 25 of Record 11 in a CORSIM 5.1

input file) with the average speeds from the CORSIM 5.1 simulation results of Step 3.1 for each link at each time period. The VB program then creates six CORSIM 5.1 input files, each for a specific time period, with the effects of the traffic flows resulting from the initial assignment or previous iteration, and activates CORSIM 5.1 to perform the static user-equilibrium traffic assignment using these files as input separately.

Step 3.5: *Compute estimated link travel times based on the travel time functions.*

The **Compute d(t)** command activates the VB program to retrieve the link travel times estimated based on the travel time functions built into the static traffic assignment model in CORSIM 5.1 from the six output files generated at Step 3.4 for each time period. Figure 6-8 shows an excerpt of the results of the estimated link travel times from the BPR formula.

```
TRAVEL TIME, d(t), ON THE PATH-LINK USING FHWA IMPEDANCE FUNCTION
TIME PERIOD  1

  Link          LENGTH (ft)    AVERAGE SPEED (MPH)   TRAVEL TIME
(sec)

(  25,   5)         1277            24.6                  35
(   1,   6)         1533            10.9                  95
(   1,  17)          272            10                    18
(   1,   2)         1552            15.9                  66
(   1,  21)         1269            29.7                  29
(  21,   1)         1269            10                    86
(  24,   8)         1256            11.5                  74
(   6,   8)         1268            10                    86
(   6,  13)         1689            29.1                  39
(   6,   7)         1552            13.3                  79
(   6,   1)         1533            10                    104
(  13,   6)         1689            16.4                  70
(  14,   7)         1552            12.7                  83
(   8,  24)         1256            28.6                  29
(   8,  18)          255            10                    17
(   8,   9)         1552            13.9                  75
(   8,   6)         1268            13.8                  62
```

Figure 6-8 Excerpt of the Link Travel Time Results from the BPR Formula.

Step 3.6: *Test convergence.*

The **Test Convergence: d(t) = T(t) ?** button compares the link travel times based on the simulation results and the BPR formula. A message box on the screen shows whether the criteria are satisfied or not. Figure 6-9 shows an excerpt of the comparison of link travel times based on simulation results and the BPR Formula. If the convergence criteria are not satisfied, clicking the **Update X(t) and Re-iterate** button will automatically run the dynamic traffic assignment procedure iteratively until the criteria are satisfied. After the criteria are satisfied, clicking the **Traffic-Responsive Signal Control** button will run Steps 2.1 through 2.4 of the dynamic traffic-responsive signal control submodel but using the traffic flows resulting from the dynamic traffic assignment submodel.

LINK	TRAVEL TIME (sec) BPR	TRAVEL TIME (sec) SIMULATION
(25, 5)	71	43
(1, 6)	104	80
(1, 17)	18	7
(1, 2)	72	105
(1, 21)	34	34
(21, 1)	86	86
(24, 8)	85	77
(6, 8)	86	78
(6, 13)	45	44
(6, 7)	105	105
(6, 1)	104	104
(13, 6)	114	114
(14, 7)	105	105
(8, 24)	34	32
(8, 18)	17	6
(8, 9)	79	105
(8, 6)	86	86
(18, 8)	17	17
(12, 11)	85	65
(2, 7)	103	51

Figure 6-9 Excerpt of the Comparison of Link Travel Times Based on Simulation Results and the BPR Formula.

Step 4 Test convergence.

The **Iterative Optimization Assignment** button runs the dynamic traffic-responsive signal control submodel and the dynamic traffic assignment submodel sequentially until the solutions converge.

The final results for the traffic-responsive signal timings and dynamic traffic flows are presented in the next Chapter.

CHAPTER 7
RESULTS AND EVALUATION

7.1 Introduction

The objective of this chapter is to evaluate the time-dependent traffic flows and the signal settings produced by the optimal dynamic transportation management system developed in this research using the numerical example presented in Chapter 6. The results for both static and dynamic cases are presented and are followed by a comparison and assessment.

7.2 Results for Static Signal Settings and Static Traffic Flows

Static optimal traffic flows determined by the traditional user-equilibrium methodology for the sample network and the given O-D matrix are shown in Figure 7-1. From the traffic flows shown in Figure 7-1, TRANSYT-7F found that the traffic signal settings shown in Figure 7-2 should produce minimum delays during the analysis time interval. There are eleven intersections and each intersection has four phases for each signal cycle. In the static case, traffic flows and signal settings remain the same throughout the analysis time interval.

7.3 Results for Adaptive Signal Settings and Dynamic Traffic Assignment

In the dynamic case, traffic flows and signal setting are time-dependent. In the example, the analysis time interval, one hour, was separated into six time periods with a duration of ten minutes for each time period.

Link	LT	TH	RT	Link	LT	TH	RT
(8025, 25)	0	700	0	(9, 8)	509	0	223
(25, 5)	146	554	0	(9, 10)	317	562	273
(1, 6)	222	446	226	(9, 7)	322	367	240
(1, 2)	69	622	139	(8015, 15)	0	710	0
(1, 21)	0	500	0	(15, 2)	370	297	43
(8021, 21)	0	910	0	(8016, 16)	0	950	0
(21, 1)	362	548	0	(16, 9)	155	530	265
(8024, 24)	0	870	0	(3, 10)	194	247	345
(24, 8)	0	501	369	(3, 2)	396	570	432
(6, 8)	419	451	0	(3, 4)	0	1037	600
(6, 13)	0	1010	0	(3, 19)	0	1050	0
(6, 7)	299	276	264	(10, 20)	0	570	0
(6, 1)	0	176	468	(10, 9)	321	561	400
(8013, 13)	0	860	0	(10, 11)	277	644	0
(13, 6)	255	396	209	(10, 3)	317	403	341
(8014, 14)	0	1010	0	(8019, 19)	0	1170	0
(14, 7)	280	395	336	(19, 3)	364	399	407
(8, 24)	0	960	0	(8020, 20)	0	1050	0
(8, 9)	0	540	248	(20, 10)	342	543	165
(8, 6)	210	293	221	(4, 23)	0	920	0
(8012, 12)	0	840	0	(4, 3)	218	674	230
(12, 11)	0	591	249	(4, 5)	0	946	276
(2, 7)	294	369	168	(8023, 23)	0	770	0
(2, 1)	346	0	324	(23, 4)	586	0	184
(2, 3)	417	932	169	(5, 11)	166	0	255
(2, 15)	0	920	0	(5, 4)	320	536	0
(7, 9)	347	550	16	(5, 25)	0	1170	0
(7, 6)	215	574	96	(11, 10)	50	595	201
(7, 14)	0	810	0	(11, 12)	0	810	0
(7, 2)	58	418	526	(11, 5)	302	0	224
(9, 16)	0	1120	0				

Figure 7-1 Static Traffic Assignment Results.

Intersection	Green Time for Each Phase				Cycle
	N-S LT	N-S TH	E-W LT	E-W TH	Length
1	23	21	19	41	104
2	22	28	20	34	104
3	18	28	18	40	104
4	15	47	24	18	104
5	9	50	20	25	104
6	12	40	15	37	104
7	13	43	16	32	104
8	19	23	17	45	104
9	15	40	12	37	104
10	12	41	12	39	104
11	11	48	9	36	104

Figure 7-2 Optimal Signal Settings for the Static Case.

Figure 7-3 shows the optimal time-dependent signal settings for those six time periods. Unlike the static signal settings shown in Figure 7-2, the dynamic traffic signal settings shown in Figure 7-3 are different in different time periods. In the dynamic case, travelers are assumed to receive real-time traffic information based on delays associated with the signal settings, and change their paths to minimize their travel times. The transportation management system then modifies the signal settings according the updated traffic flows at each time period to minimize the system-wide delay. The optimal dynamic traffic flows are shown in Figure 7-4.

Time Period: 1

	N-S LT	N-S TH	E-W LT	E-W TH
1	22	20	20	41
6	12	40	14	38
8	18	22	17	47
2	22	27	20	35
7	13	44	16	31
9	15	40	12	38
3	19	28	18	40
10	12	41	13	39
4	14	45	27	18
5	8	54	18	23
11	11	52	8	33

Time Period: 2

	N-S LT	N-S TH	E-W LT	E-W TH
1	23	21	19	41
6	13	40	15	36
8	19	22	17	47
2	22	28	21	34
7	13	43	16	32
9	15	40	12	37
3	19	29	17	39
10	12	41	12	39
4	15	46	25	19
5	8	54	19	23
11	10	52	8	33

Time Period: 3

	N-S LT	N-S TH	E-W LT	E-W TH
1	23	20	20	41
6	13	40	15	37
8	18	22	17	47
2	23	27	20	34
7	13	43	16	32
9	15	40	12	37
3	18	28	18	40
10	12	39	13	40
4	14	46	26	18
5	8	54	19	23
11	10	52	8	34

Time Period: 4

	N-S LT	N-S TH	E-W LT	E-W TH
1	23	21	20	41
6	12	41	15	37
8	18	21	18	46
2	22	27	21	34
7	13	43	16	32
9	16	38	13	36
3	18	29	18	40
10	12	41	12	39
4	14	46	26	18
5	8	54	19	23
11	10	51	8	34

Time Period: 5

	N-S LT	N-S TH	E-W LT	E-W TH
1	23	22	19	41
6	12	40	15	37
8	19	24	17	45
2	23	29	19	33
7	13	43	16	32
9	14	41	11	38
3	17	29	19	38
10	12	39	13	40
4	15	46	24	19
5	8	54	19	23
11	13	50	8	33

Time Period: 6

	N-S LT	N-S TH	E-W LT	E-W TH
1	24	20	18	42
6	12	39	15	37
8	18	23	18	45
2	22	27	20	34
7	13	44	16	32
9	14	41	11	38
3	18	29	18	40
10	12	41	12	39
4	15	46	25	18
5	8	53	19	24
11	13	50	8	33

Figure 7-3 Optimal Time-Dependent Signal Timing Settings for the Example Network.

For the example network, the run-time on a computer with 900 MHz CPU was about 6 minutes for each iteration. It took the program less than an hour to reach the optimal

results. Although this cannot be used to proclaim the feasibility in terms of computer run-time for applications with real networks, which are usually in large in scale, the program should be suitable for rather large network processing applications.

	TP=1			TP=2			TP=3		
LINK	L	T	R	L	T	R	L	T	R
(8025, 25)	0	117	0	0	117	0	0	117	0
(25, 5)	14	103	0	14	102	0	22	95	0
(1, 6)	42	73	40	42	73	48	42	66	48
(1, 2)	21	125	13	33	100	11	19	131	13
(1, 21)	0	83	0	0	83	0	0	83	0
(8021, 21)	0	152	0	0	152	0	0	152	0
(21, 1)	74	78	0	74	77	0	78	74	0
(8024, 24)	0	145	0	0	145	0	0	145	0
(24, 8)	0	84	62	0	86	59	0	79	67
(6, 8)	56	84	0	69	82	0	61	72	0
(6, 13)	0	168	0	0	168	0	0	168	0
(6, 7)	38	60	44	52	53	37	34	54	45
(6, 1)	0	48	84	0	43	69	0	44	84
(8013, 13)	0	143	0	0	143	0	0	143	0
(13, 6)	33	71	39	26	70	48	34	70	40
(8014, 14)	0	168	0	0	168	0	0	168	0
(14, 7)	50	79	39	56	64	48	56	72	40
(8, 24)	0	160	0	0	160	0	0	160	0
(8, 9)	6	75	36	0	80	48	14	73	40
(8, 6)	29	56	28	35	51	30	44	48	21
(8012, 12)	0	140	0	0	140	0	0	140	0
(12, 11)	0	99	41	0	94	47	0	94	46
(2, 7)	47	53	40	45	43	29	44	46	29
(2, 1)	78	0	35	86	0	40	83	0	40
(2, 3)	70	155	40	58	147	43	65	152	49
(2, 15)	0	153	0	0	153	0	0	153	0
(7, 9)	48	99	1	55	80	1	53	89	5
(7, 6)	27	99	44	30	85	37	32	77	46
(7, 14)	0	135	0	0	135	0	0	135	0
(7, 2)	2	69	65	5	58	74	5	68	62
(9, 16)	0	187	0	0	187	0	0	187	0
(9, 8)	76	0	29	78	0	30	83	0	34
(9, 10)	57	83	40	60	97	40	69	76	34
(9, 7)	51	59	29	59	36	38	54	60	37
(8015, 15)	0	118	0	0	118	0	0	118	0
(15, 2)	75	34	10	75	28	16	74	31	14
(8016, 16)	0	158	0	0	158	0	0	158	0
(16, 9)	27	75	57	24	73	62	26	80	53
(3, 10)	39	47	44	49	45	63	48	51	57
(3, 2)	94	101	64	78	105	62	75	104	67
(3, 4)	0	184	102	0	165	89	0	172	90
(3, 19)	0	175	0	0	175	0	0	175	0
(10, 20)	0	95	0	0	95	0	0	95	0
(10, 9)	52	78	58	58	84	60	58	87	57
(10, 11)	46	104	0	71	108	0	57	103	0
(10, 3)	51	67	65	45	69	55	48	70	61

Figure 7-4 Optimal Time-Dependent Traffic Flows for the Six Time Periods.

LINK	TP=1 L	TP=1 T	TP=1 R	TP=2 L	TP=2 T	TP=2 R	TP=3 L	TP=3 T	TP=3 R
(8019, 19)	0	195	0	0	195	0	0	195	0
(19, 3)	65	64	66	53	80	62	49	84	62
(8020, 20)	0	175	0	0	175	0	0	175	0
(20, 10)	53	94	28	54	84	36	53	85	36
(4, 23)	0	153	0	0	153	0	0	153	0
(4, 3)	26	142	38	30	138	49	23	136	40
(4, 5)	0	160	50	0	155	37	0	159	41
(8023, 23)	0	128	0	0	128	0	0	128	0
(23, 4)	102	0	26	101	0	27	101	0	28
(5, 11)	31	0	32	27	0	25	32	0	31
(5, 4)	52	104	0	64	116	0	64	98	0
(5, 25)	0	195	0	0	195	0	0	195	0
(11, 10)	8	91	31	10	84	24	10	90	25
(11, 12)	0	135	0	0	135	0	0	135	0
(11, 5)	52	0	35	78	0	40	67	0	36

LINK	TP=4 L	TP=4 T	TP=4 R	TP=5 L	TP=5 T	TP=5 R	TP=6 L	TP=6 T	TP=6 R
(8025, 25)	0	117	0	0	117	0	0	117	0
(25, 5)	27	89	0	26	90	0	21	96	0
(1, 6)	51	68	46	49	66	46	47	69	42
(1, 2)	28	114	19	22	125	12	25	111	9
(1, 21)	0	83	0	0	83	0	0	83	0
(8021, 21)	0	152	0	0	152	0	0	152	0
(21, 1)	79	73	0	80	72	0	74	77	0
(8024, 24)	0	145	0	0	145	0	0	145	0
(24, 8)	0	89	56	0	78	67	0	81	64
(6, 8)	56	79	0	53	81	0	50	80	0
(6, 13)	0	168	0	0	168	0	0	168	0
(6, 7)	47	59	46	34	61	45	45	60	55
(6, 1)	0	46	82	0	48	79	0	47	71
(8013, 13)	0	143	0	0	143	0	0	143	0
(13, 6)	26	81	37	47	60	37	31	81	31
(8014, 14)	0	168	0	0	168	0	0	168	0
(14, 7)	52	71	45	57	62	50	50	76	43
(8, 24)	0	160	0	0	160	0	0	160	0
(8, 9)	0	72	40	5	72	44	0	73	40
(8, 6)	43	59	20	50	35	31	40	42	32
(8012, 12)	0	140	0	0	140	0	0	140	0
(12, 11)	0	98	42	0	93	47	0	95	46
(2, 7)	42	48	32	49	42	35	43	49	34
(2, 1)	91	0	38	89	0	36	81	0	36
(2, 3)	58	166	40	69	153	42	58	151	44
(2, 15)	0	153	0	0	153	0	0	153	0
(7, 9)	53	86	8	56	83	6	58	89	8
(7, 6)	30	80	43	30	73	45	30	86	46
(7, 14)	0	135	0	0	135	0	0	135	0
(7, 2)	4	65	72	6	69	65	0	71	71
(9, 16)	0	187	0	0	187	0	0	187	0
(9, 8)	81	0	33	79	0	39	80	0	33
(9, 10)	52	91	37	53	87	41	56	95	41
(9, 7)	50	49	34	51	55	26	51	54	32
(8015, 15)	0	118	0	0	118	0	0	118	0

Figure 7-4. Continued

	TP=4			TP=5			TP=6		
LINK	L	T	R	L	T	R	L	T	R
(15, 2)	78	26	14	74	32	12	71	39	9
(8016, 16)	0	158	0	0	158	0	0	158	0
(16, 9)	21	81	56	25	80	54	19	77	62
(3, 10)	41	47	53	46	46	53	48	44	67
(3, 2)	77	111	61	82	107	62	79	108	57
(3, 4)	0	169	98	0	161	100	0	155	95
(3, 19)	0	175	0	0	175	0	0	175	0
(10, 20)	0	95	0	0	95	0	0	95	0
(10, 9)	61	86	52	60	88	48	57	86	59
(10, 11)	63	104	0	59	110	0	67	111	0
(10, 3)	47	65	56	49	61	58	45	68	55
(8019, 19)	0	195	0	0	195	0	0	195	0
(19, 3)	46	83	67	50	80	65	45	85	65
(8020, 20)	0	175	0	0	175	0	0	175	0
(20, 10)	51	90	34	60	79	36	52	89	35
(4, 23)	0	153	0	0	153	0	0	153	0
(4, 3)	18	135	53	22	154	45	29	134	49
(4, 5)	0	160	37	0	155	34	0	159	26
(8023, 23)	0	128	0	0	128	0	0	128	0
(23, 4)	101	0	27	101	0	27	99	0	29
(5, 11)	32	0	32	25	0	35	24	0	23
(5, 4)	55	104	0	54	103	0	58	113	0
(5, 25)	0	195	0	0	195	0	0	195	0
(11, 10)	11	94	25	8	83	37	10	84	24

Figure 7-4. Continued

7.4 Comparison of Static and Dynamic Transportation Management Systems

CORSIM 5.1 was used to simulate both static and dynamic cases. Again, because CORSIM 5.1 simulates traffic conditions of a network with the feature that traffic characteristics can change with time, the time-varying signal timing plans (shown in Figure 7-3) and traffic flows (shown in Figure 7-4) in the dynamic case were specified in a sequence of time periods in a CORSIM 5.1 input file for simulation and animation. For the static case, cumulative results were reported by CORSIM 5.1 every ten minutes. For the dynamic case, CORSIM 5.1 produced its cumulative simulation results at the completion of each time period. Figures 7-5 and 7-6 show the excerpts of the cumulative simulation results, e.g.,vehicle miles and vehicle delay time, at the elapsed simulation time of 40 minutes for the static and dynamic cases, respectively.

To measure quantitatively and compare the quality of traffic service (i.e., frequency, expediency, smoothness and safety) for both static and dynamic transportation management systems, a Measure of Effectiveness (MOE) has to be used. The available MOEs include delay, average vehicle speed, degree of saturation, number of stops, fuel consumption, etc. Among these useful MOEs, delay time, which includes increased travel time from reduced speed and time added due to traffic signal control, is a primary MOE used to evaluate the performance of transportation systems.

Table 7-1 summarizes the cumulative network-wide vehicle delays at the end of each 10-minute time period for both the static and dynamic cases. The table shows the traffic conditions were improved at the fourth time period for the dynamic case. At end of the one-hour analysis period, for the sample network and the given traffic demands, the network-wide delay was 1,102 vehicle-hours for the static case. For the dynamic case, CORSIM 5.1 simulated the six 10-minute time periods sequentially and the network-wide delay during the one hour analysis period was 925 vehicle-hours. The fifth column in Figures 7-5 and 7-6 shows the vehicle delay for each link at the end of the fourth period for both cases. The results show that the traffic conditions on some links were improved in the dynamic case but some became worse. For example, for Link (12, 11), the total delay was 1,453 vehicle-minutes in the static case and was improved to 338 vehicle-minutes in the dynamic case. For Link (6,8), opposite results were observed. However, by summing up the vehicle delay on each link, the network-wide delay in the dynamic case was improved by 43 vehicle-hours at the end of the fourth time period and 150 vehicle-hours for the complete one-hour analysis period.

Table 7-1 Cumulative Network-Wide Vehicle Delays at 10-Minute Time Intervals for Static and Dynamic Cases

Elapsed Simulation Time	Total Delay (vehicle-hours)	
	Static Case	Dynamic Case
10 min.	98.93	101.22
20 min.	235.88	241.67
30 min.	409.56	407.22
40 min.	617.55	574.50
50 min.	846.67	763.45
60 min.	1,101.98	952.07

```
CUMULATIVE NETSIM RESULTS AT TIME 12:40: 0
ELAPSED TIME IS  0:40: 0 ( 2400 SECONDS)

                                    VEHICLE MINUTES
                  VEHICLE        MOVE   DELAY   TOTAL
LINK            MILES TRIPS      TIME    TIME    TIME
----            -----------      --------------------
(  25,   5)  112.70   466   150.3   229.8   380.1
(   1,   6)  161.43   556   322.9   652.0   974.9
(   1,   2)  142.69   489   190.3   354.7   544.9
(   1,  21)   98.80   415   197.6    30.1   227.7
(  21,   1)  136.27   567   272.5  1375.6  1648.1
(  24,   8)  112.28   472   224.6  2418.8  2643.3
(   6,   8)  106.84   447   213.7   443.6   657.3
(   6,  13)  157.08   493   314.2    49.5   363.6
(   6,   7)  176.50   603   353.0  1196.6  1549.6
(   6,   1)  128.86   446   257.7  1272.0  1529.7
(  13,   6)  184.25   576   368.5   773.6  1142.2
(  14,   7)  182.54   621   365.1  1634.9  2000.0
(   8,  24)  106.57   448   213.1    46.6   259.7
(   8,   9)  105.38   361   210.8   382.0   592.8
(   8,   6)  115.46   484   230.9   389.6   620.5
(  12,  11)  123.99   520   248.0  1452.5  1700.5
(   2,   7)  152.29   529   227.1   791.7  1018.9
(   2,   1)  136.36   464   181.8   192.2   374.0
(   2,   3)  500.44   840   667.3  1467.5  2134.8
(   2,  15)   65.96   296    98.4    17.2   115.6
(   7,   9)  136.31   566   203.3   921.5  1124.8
(   7,   6)  135.46   462   270.9   837.1  1108.0
(   7,  14)  182.56   623   365.1    61.2   426.3
(   7,   2)  122.48   426   182.7  1020.2  1202.9
```

Figure 7-5 Excerpt of the Cumulative Simulation Results at Elapsed Time of 40 Minutes for the Static Case.

```
CUMULATIVE NETSIM RESULTS AT TIME 12:40: 0
ELAPSED TIME IS  0:40: 0 ( 2400 SECONDS)

                                    VEHICLE MINUTES
                  VEHICLE        MOVE   DELAY   TOTAL
LINK            MILES TRIPS      TIME    TIME    TIME
----            -----------      --------------------
(  25,   5)  111.01   459   148.0   352.5   500.5
(   1,   6)  156.78   540   313.6  1045.9  1359.4
(   1,   2)  148.76   510   198.4   357.9   556.3
(   1,  21)   66.76   281   133.5    14.4   147.9
(  21,   1)  143.00   595   286.0   937.7  1223.8
(  24,   8)  117.27   493   234.5  1956.2  2190.7
(   6,   8)  116.30   487   232.6  1046.7  1279.3
(   6,  13)  153.26   481   306.5    51.6   358.1
(   6,   7)  154.91   529   309.8   769.1  1079.0
(   6,   1)  106.06   366   212.1   319.7   531.8
(  13,   6)  183.93   575   367.9   754.8  1122.7
(  14,   7)  155.79   530   311.6  2122.0  2433.5
(   8,  24)  103.00   433   206.0    42.1   248.1
(   8,   9)  109.11   374   218.2   582.7   800.9
(   8,   6)   91.69   383   183.4   304.0   487.4
(  12,  11)  114.22   479   228.4   337.6   566.0
(   2,   7)  140.03   487   208.9   626.7   835.6
(   2,   1)  118.53   404   158.0   194.7   352.7
(   2,   3)  474.88   797   633.2   776.3  1409.4
(   2,  15)  100.15   450   149.4    29.5   178.8
(   7,   9)  120.32   500   179.5   859.1  1038.5
(   7,   6)  111.97   382   223.9   327.9   551.9
(   7,  14)  127.90   436   255.8    40.0   295.8
(   7,   2)  116.05   404   173.1  2201.1  2374.2
```

Figure 7-6 Excerpt of the Cumulative Simulation Results at Elapsed Time of 40 Minutes for the Dynamic Case.

Figures 7-7 and 7-8 show the animations for a portion of the sample network at the end of the one-hour period for the static and dynamic cases, respectively. Figure 7-7 shows Link (6,7) in the static case had more serious spillback problems. Based on the results from the numerical example, the dynamic traffic management system performed more efficiently, by modifying the traffic conditions based on the signal settings and then adjusting the signal timing settings according to the updated traffic flows six times during an hour than the static traffic management system did for the same traffic facilities.

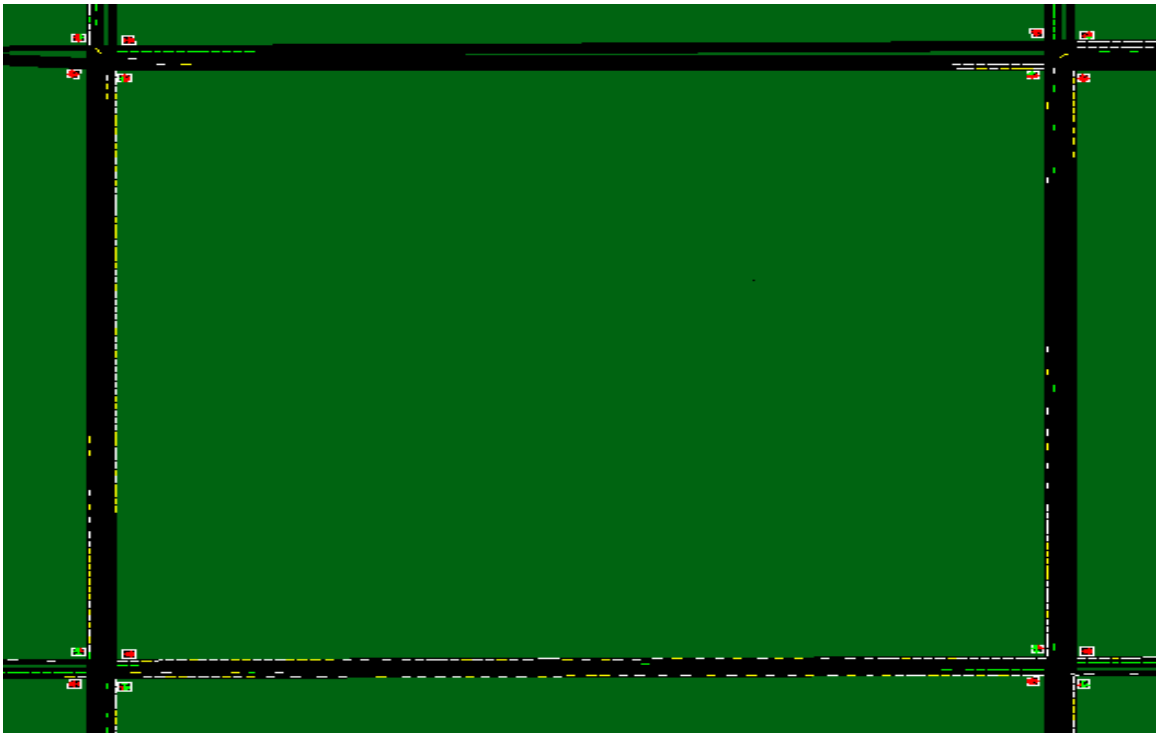

Figure 7-7 Example of Animation Result for the Static Case.

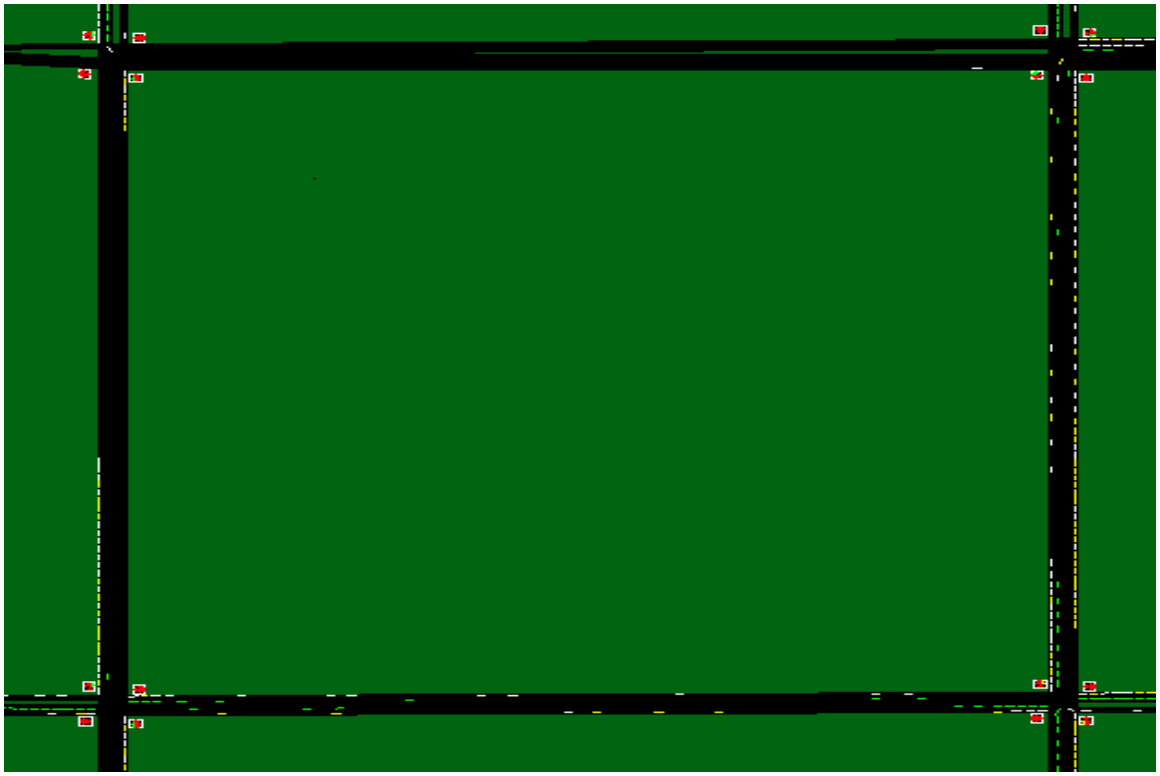

Figure 7-8 Example of Animation Result for the Dynamic Case.

CHAPTER 8
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

8.1 Summary and Conclusions

Intelligent Transportation Systems (ITS), which have been a topic of substantial research during the past decade, are being developed to improve the efficiency and productivity of existing transportation facilities. The success of ITS depends on two important sub-systems, Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS).

Real-time traffic control and dynamic traffic assignment are two major support technologies of ATMS because the traditional static network equilibrium models which were developed for long-term transportation planning, and the traditional traffic control management systems which were developed to set traffic signals by assuming fixed flow patterns, are not suitable for analyzing and solving transportation problems in real time.

A combined model has been developed to integrate the real-time traffic control model and the dynamic traffic assignment model. For the real-time traffic control model, the framework of the traditional traffic control model, which minimizes total delay, was extended to the dynamic case by adding time as an additional dimension. However, the conventional delay model, Webster's delay formula, which was originally derived through theoretical queuing analysis for isolated intersections, predicts infinite values of delay when flows approach capacity. Yet, in realistic situations, a queue will not grow infinitely because

drivers will change routes to avoid large queues. The generalized delay model for signalized intersections in the 2000 Highway Capacity Manual was used in the real-time traffic control model to take the oversaturated queue problem into account. A solution algorithm applying the Frank-Wolfe method has been developed to implement the real-time traffic control model. It uses TRANSYT-7F's static optimal signal settings as the initial solution to increase the chance for the non-convex objective function to converge to a near-optimal solution.

The dynamic traffic assignment model is formulated as a Variational Inequality (VI) formula. Using the simulation model to calculate link travel times, a relaxation algorithm is used to relax the asymmetric link travel time function resulting from the flow propagation constraints. A solution algorithm has been developed to implement the equivalent optimization model of the relaxed VI formula.

The iterative optimization assignment (IOA) procedure consisting of solving sequentially an optimization model involving the signal timing variables, with the flow variables fixed, and a user-optimized equilibrium model corresponding to the new green time settings, is used to solve the combined model.

Applying the IOA procedure, a solution algorithm with operational capability to solve dynamic network management models corresponding to the above technological concepts in ITS has been developed. A comprehensive computer program has been prepared to implement and test all algorithms associated with the dynamic transportation management model.

CORSIM 5.1, the simulation software which allows traffic characteristics to be time-varying, was used to demonstrate the performances of both static and dynamic cases using

a sample network and a given O-D demand matrix. Because of the limitations on resources available to this study, a sample network instead of the real-world data was used to test the procedure.

It is a common practice that multiple simulation runs, each specified with different sets of random seeds, are performed when CORSIM or other simulation computer packages are used. However, since the variation in the simulation results obtained by changing random number seeds are generally much less than 13.6%, only one CORSIM simulation run was performed in this study. The test results indicate that dynamic traffic assignment with adaptive traffic-responsive signal settings reduced the network-wide delays about 13.6% by periodically modifying the traffic flows based on the traffic conditions and then adjusting the signal timing settings according to time-dependent traffic flows.

This study applied mathematical programming methodologies to model traffic assignment and traffic signal control systems as well as the dynamic interactions between them. The dynamic transportation management system was modeled more realistically than those under a static state. In addition, the mathematical solutions are proven to be optimal and satisfy the equilibrium conditions in comparison with those developed solely based upon simulation techniques. Incorporating time-dependent optimal signal settings into an existing traffic control system can increase system efficiency by better assessing network conditions and achieving better traffic control.

8.2 Recommendations

The travel time function is very important. The same traffic assignment policy considered under different disutility assumptions may possess completely different theoretical properties and thus may be expected to produce completely different traffic flow

results. The BPR formula is known to have the limitation of an ambiguous definition of capacity in the function because the BPR formula increases travel times even after a link's traffic flow is higher than its capacity. Further investigation into enhancements of the BPR formula is recommended.

The dynamic queue-responsive traffic signal control with dynamic traffic assignment technique developed in this research consists of a number of parameters, including the cycle length, signal phase plan and O-D demand, that require input from users and were not modified during the optimization. Unlike the simulation-based, real-time computer systems which can simulate the traveler's behavior of changing departure time in response to congestion conditions, the O-D demand is presumed fixed during the analysis period in the mathematical programming model developed in this study. In other words, travelers would not be able to change their departure time because of traffic conditions. This assumption is practical for peak hour traffic since most roadway users do not have the privilege of arbitrarily altering the time they begin work or complete their jobs for the day. Further research is recommended to consider varied cycle lengths and signal phase sequences by programming additional external looping into the optimization process.

In addition, the traffic persisting for an hour under the static state was separated into six discrete 10-min time intervals for the example. The duration of each time period may change the performance of a dynamic transportation management system. Intuitively, the smaller the duration of each time period, the better the performance of the transportation system since the optimal signal settings and traffic flows can be updated more frequently. However, the marginal benefits for dividing hourly demand into shorter time periods may be limited since it takes time for road users to respond to the traveler information they are

given and for new signal timing plans to become effective. Further research is thus recommended to determine the robustness of the duration of discrete time interval.

While it is desirable to reduce the burden of input from users, it is also desirable to have some control over the IOA procedure through some parameter settings such as the convergence criteria and the maximum number of iterations. This is because the impact of the values of these parameters involve some trade-offs between computer running time and the quality of the solution. It might be beneficial to investigate the possibility of calibrating the optimal values for these parameters.

Although the IOA procedure has been demonstrated empirically in previous research as producing a near optimal solution, the approach is not theoretically proven to minimize total travel time. It might be more practical to change different parameters and run the procedure again if the result leads to a decline in network performance rather than an improvement. A criterion can be added at the convergence test to monitor if there is a decline in network performance.

To simply the process, the Visual Basic program was designed to only handle intersections without diagonal traffic, and with pre-timed signals with protected left-turning movements and no permitted left-turning movements during other signal phases. Ideally, the program should handle any signal control type allowed in CORSIM 5.1 because the program reads the CORSIM 5.1 input files. Future work should include developing modules to process different signal control types and more complex geometries.

REFERENCES

1. Federal Highway Administration, *A Roadmap for the Research, Development and Deployment of Traffic Estimation and Prediction Systems for Real-Time and Off-Line Applications (Treps, Treps-P)*, Office of Operations Research and Development, July, 2001.

2. Mahmassani, H.S., H.A. Sbayti, and Y.-C. Chiu, *DynaSMART-P User's Guide*, Version 0.930.0, U.S. Department of Transportation, Federal Highway Administration, October 2002.

3. Sheffi, Y., *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1985.

4. Drissi-Kaïtouni, O., "A Variational Inequality Formulation of the Dynamic Traffic Assignment Problem," *European Journal of Operational Research*, Vol. 71, pp. 188-204, 1993.

5. Ran, B. and D. Boyce, *Modeling Dynamic Transportation Networks: an Intelligent Transportation System Oriented Approach*, Springer, Berlin, 1996.

6. Gartner, N.H., C. Stamatiadis, and P.J. Tarnoff, "Development of Advanced Traffic Signal Control Strategies for ITS: A Multi-Level Design," *Transportation Research Record 1494*, Transportation Research Board, National Research Council, Washington, DC, pp. 98-105, 1995.

7. Allsop, R.E., "Some Possibilities for Using Traffic Control to Influence Trip Distribution and Route Choice," in Proceedings of the 6th International Symposium on Transportation and Traffic Theory, Elsevier, Amsterdam, 1974, pp. 345-374.

8. Smith, M.J., "Properties of a Traffic Control Policy Which Ensure the Existence of Traffic Equilibrium Consistent with the Policy," *Transportation Research*, Vol. 15B, No. 6, pp.453-462, 1981.

9. Sheffi, Y. and W.B. Powell, "Optimal Signal Settings over Transportation Networks," *Journal of Transportation Engineering*, Vol. 109, No. 6, pp. 824-839, 1983.

10. Smith, M.J. and T.V. Vuren, "Traffic Equilibrium with Responsive Traffic Control," *Transportation Science*, Vol. 27, No. 2, pp. 118-132, 1993.

11. Yang, H. and S. Yagar, "Traffic Assignment and Traffic Control in a Saturated Road Network," *Transportation Research*, Vol. 29A, No. 2, pp. 125-139, 1995.

12. Chen, H.-K. and C.-F. Hsueh, "Combining Signal Timing Plan and Dynamic Traffic Assignment," a paper presented at the 76th Annual Transportation Research Board Meeting, Washington, D.C., 1997.

13. Gartner, N.H. and C. Stamatiadis, "Integration of Dynamic Traffic Assignment with Real Time Traffic Adaptive Control," a paper presented at the 76th Annual Transportation Research Board Meeting, Washington, D.C., 1997.

14. Chen, O.J. and M.E. Ben-Akiva, "Dynamic Traffic Control and Assignment: A Game Theoretic Approach," a paper presented at the 77th Annual Transportation Research Board Meeting, Washington, D.C., 1998.

15. Transportation Research Board, *Highway Capacity Manual*, National Research Council, Washington, D.C., 2000.

16. Wallace, C.E., K.G. Courage, M.A. Hadi, and A. Gan,"Methodology for Optimizing Signal Timing: M|O|S|T Volume 4, TRANSYT-7F Users Guide," FHWA, Office of Traffic Operations and Intelligent Vehicle/Highway Systems, 1997.

17. Beckmann, M.J., C.B. McGuire, and C.B. Winston, *Studies in the Economics of Transportation*, Yale University Press, New Haven, Conn, 1956.

18. Leblanc, L.J., E.K. Morlok, and W. Pierskalla, "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem," *Transportation Research*, Vol. 9B, No. 5, pp. 309-318, 1975.

19. Potts, R.B., and R.M. Oliver, *Flows in Transportation Networks*, Academic Press, New York, 1972.

20. Sheffi, Y., and W.B. Powell, "An Algorithm for the Equilibrium Assignment Problem with Random Link Times", *Networks*, Vol. 12, No. 2, pp. 191-207, 1982.

21. Dial, R.B., "A Probabilistic Multipath Traffic Assignment Algorithm Which Obviates Path Enumeration," *Transportation Research*, Vol. 5A, No. 2, pp. 83-111, 1971.

22. Sheffi, Y., and W.B. Powell, "A Comparison of Stochastic and Deterministic Traffic Assignment Over Congested Networks," *Transportation Research*, Vol. 15B, No. 1, pp. 53-64, 1981.

23. Gartner, N.H. and G. Improta, *Urban Traffic Networks*, Springer Verlag, Berlin, 1995.

24. Gartner, N.H., "OPAC: A Demand-Responsive Strategy for Traffic Signal Control," *Transportation Research Record 906*, Transportation Research Board, National Research Council, Washington, DC, pp. 75-81, 1983.

25. Merchant, D.K. and G.L. Nemhauser, "A Model and an Algorithm for the Dynamic Traffic Assignment Problem," *Transportation Science*, Vol. 12, pp.183-199, 1978.

26. Chen, H.-K, C.-W. Chang, M.-S. Chang, and C.-Y. Wang, "A Comparison of Link-Based versus Route-Based Algorithms with the Dynamic User-Optimal Route Choice Problem," a paper presented at the 77th Annual Transportation Research Board Meeting, Washington, D.C., 1998.

27. Fisk, C.S., "Game Theory and Transportation Systems Modeling," *Transportation Research*, Vol. 18B, No. 4/5, pp. 301-313, 1984.

28. Marcotte, P., "Network Design Problem with Congestion Effects: A Case of Bilevel Programming," *Mathematical Programming*, Vol. 34, No. 1, pp.142-162, 1986.

29. Fambro, D.B. and N.M. Rouphail, "Generalized Delay Model for Signalized Intersections and Arterial Streets," *Transportation Research Record 1572*, Transportation Research Board, National Research Council, Washington, DC, pp. 112-121, 1996.

30. Rouphail, N.M., M. Anwar, D.B. Fambro, P. Sloup, and C.E. Perez, "Validation of Generalized Delay Model for Vehicle-Actuated Traffic Signals," *Transportation Research Record 1572*, Transportation Research Board, National Research Council, Washington, DC, pp. 105-111, 1996.

31. *TRANSIMS-2.1 User Documents*, Los Alamos National Laboratory, July 2001.

32. ITT Systems & Sciences Corporation, *CORSIM User's Manual*, Version 5.1, Colorado Springs, CO, January 2002.

33. Nagurney, A., *Network Economics: A Variational Inequality Approach*, Boston, Massachusetts, Kluwer Academic Publishers, 1993.

34. Ortega, J.M. and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, New York, Academic Press, 1970.

35. Bard, J.F. and J.E. Falk, "An Explicit Solution to the Multilevel Programming Problem," *Computers and Operations Research*, Vol. 9, pp. 77-100, 1982.

36. Smith, M.J. and T.V. Vuren, "Traffic Equilibrium with Responsive Traffic Control," *Transportation Science*, Vol. 27, No. 2, pp. 118-132, 1993.

BIOGRAPHICAL SKETCH

Lee-Fang Chow was born in Fung-Sang, Taiwan, on July, 4, 1966. She received her B.B.A. degree in transportation engineering and management from the National Chiao-Tung University at Taiwan in 1988. Before coming to the United States in 1992, she had worked as a traffic engineer in two major consulting companies in Taiwan for four years. She received her M.S. degree in civil engineering from the University of Florida in 1994 and continued to pursue doctoral studies in 1996.

Ms. Chow worked as a graduate research assistant with the University of Florida Transportation Research Center from 1993 to 1994 and completed a sponsored research project with Dr. Mohammed Hadi and Dr. Joseph Wattleworth. She worked as a graduate teaching assistant with Dr. Gary Long and Dr. Albert Gan for the courses of transportation engineering and civil engineering systems from 1996 to 2000. She was also involved in the test of EVIPAS and other traffic computer packages developed under the sponsorship of the Federal Highway Administration. Ms. Chow has worked as a research associate with the Florida International University Transportation Research Center since 2000.

She won the Best Student Paper Award of District 10 Institute of Transportation Engineers in 1997. Ms. Chow is married to Min-Tang Li.