A REVIEW OF DYNAMIC ASSIGNMENT AND TRAFFIC SIMULATION MODELS FOR ADIS/ATMS APPLICATIONS

Hani S. Mahmassani
Srinivas Peeta
The University of Texas at Austin

Gang-Len Chang
Thanavat Junchaya
The University of Maryland

TECHNICAL REPORT DTFH61-90-R-00074

CENTER FOR TRANSPORTATION RESEARCH
THE UNIVERSITY OF TEXAS AT AUSTIN

April 1991

TASK A REPORT FOR DOT/FHWA PROJECT DTFH61-90-R-00074
"TRAFFIC MODELING TO SUPPORT ADVANCED DRIVER INFORMATION SYSTEMS (ADIS)"

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

OVERVIEW

This report summarizes the findings of Task A of project DTFH61-90-R-00074, "Traffic Modelling to Support Advanced Driver Information Systems (ADIS)". The objective of this task is to provide a review of dynamic assignment and traffic simulation models that could be used to support the functional operating core of ADIS and ATMS (Advanced Traffic Management Systems). In addition, this task provides recommendations on the most profitable direction for further development of the required dynamic assignment and traffic modelling capabilities.

Two principal functions are addressed by the assignment-simulation models in an ADIS/ATMS context. The first consists of determining, in real-time, the network paths to which the drivers should be directed in going toward their destination, so as to achieve system-level objectives. The second is the prediction or description of the time-varying link flow patterns that result from the path choices made by motorists in response to supplied route guidance or other forms of information.

Three main areas have been reviewed under this task: (1) traffic simulation models, (2) dynamic traffic assignment models, and (3) parallel computing environments. The review of traffic simulation models has confirmed that none of the existing models adequately meet the functional requirements identified for the ADIS/ ATMS context. One of the primary limitations of these models is their inability to recognize and process travel paths through the network, and the absence of the basic data structures that would be necessary to support such network path processing.

The review of dynamic assignment models has revealed that existing formulations do not adequately capture the essential elements of the problem faced in the ADIS/ATMS context. No formulations are generally accepted for either the so-called user optimal or system optimal assignment problems. Similarly, no algorithmic procedures are known for solving those formulations for general networks. In addition, virtually all mathematical programming-based formulations do not adequately model traffic congestion formation and dissipation in the network. Some procedures combining simulation and assignment capabilities have been proposed to remedy these concerns. However, existing procedures do not satisfactorily meet all the functional requirements of the ADIS/ATMS environment.

A review of parallel processing computers and architectures has also been conducted, revealing a wide array of capabilities, with different combinations of price, performance and program development effort. Recognizing and exploiting parallelism is essential in the development of solution algorithms and their implementation. Two

prototype development environments have been identified as the most promising and comprehensive for the purpose of this study.

In light of the above deficiencies in both traffic assignment and traffic simulation model capabilities, the study team recommends the following: (1) it is necessary to develop formulations of the system-optimal dynamic assignment problems under different assumptions regarding information availability to the central controller; (2) in solving the above formulations, the values of the performance measures (objective functions) should preferably be calculated using a traffic simulation model, thereby requiring an integrated assignment-simulation capability; (3) several solution approaches, algorithms and heuristics should be developed and tested for the dynamic assignment problem; (4) the development of an integrated simulation-assignment capability will most profitably proceed on the basis of a fundamental approach to the best way of providing for the functional requirements of the problem, rather than the modification of an existing computer program; (5) the simulation-assignment modelling framework developed at the University of Texas at Austin appears to provide a sound basis for incorporating the various functional capabilities necessary in an ADIS/ATMS context; and (6) prototype development of codes for the assignment-simulation capabilities is recommended in two different computing environments—a CRAY Y-MP supercomputer with eight parallel vector processors (asynchronous MIMD parallel vector supercomputer) and a massively parallel SIMD Connection Machine.

CHAPTER 1
INTRODUCTION

1. Motivation

Applications of advanced technologies in telecommunications, information technology, microprocessors and automation to intelligent vehicle-highway systems provide new opportunities to improve the performance of traffic networks under both recurrent and non-recurrent congestion. For instance, Advanced Driver Information Systems (ADIS) and Advanced Traffic Management Systems (ATMS) will provide drivers the capability to communicate with the network control center on a real-time basis.

However, the sophistication in technological and hardware capabilities needs to be matched by more powerful methodological and algorithmic constructs than presently available, especially for real-time control in large-scale traffic systems. This need is perhaps nowhere more evident than in the development of dynamic route assignment and associated real-time network traffic simulation capabilities.

2. Problem Definition

The problem addressed in this study consists of the specification and development of dynamic network assignment capabilities and associated traffic performance simulation capabilities that will be necessary to achieve the potential of in-vehicle route guidance Advanced Driver Information Systems (ADIS) for improving the productivity and efficiency of traffic networks under recurrent and non-recurrent congestion. The dynamic assignment capabilities required must serve the following principal functions:

1. Allow a central controller, with partial or complete information about time-dependent origin destination (O-D) trip desires as well as current link status conditions (loadings, prevailing link travel times, capacity reducing incidents), to route all trips from their current position (including initial origins and intermediate locations) to their respective destinations and thereby perform an assignment either to the network's links or along paths on the network, so as to achieve systemwide objectives, subject to certain constraints. This information would form the basis of route guidance instructions to be provided to suitably equipped vehicles on a real-time basis. This capability would be used primarily on-line for the above purpose, or off-line to determine initial assignments and routing schemes for the routine and historically known trip patterns, which would subsequently be updated on-line.

2. Allow the controller or analyst to determine, for known O-D trip desires, the time varying link flow patterns that result from the path choice decisions made by

motorists, in response to real-time information supplied by the ADIS controller. This on board information might consist of specific route guidance instructions, or of prevailing and/or predicted link trip times, subjected to varying degrees of on-board and/or central processing. This descriptive assignment capability is needed off-line to evaluate alternative traffic control schemes, information supply strategies, and/or normative routing and assignment approaches, as well as on-line in connection with a model system to determine what information to provide to motorists.

Both types of assignment capabilities require a network traffic simulation capability to determine the principal figures of merit that describe the performance of the system, particularly the link trip times, for a given dynamic assignment pattern (i.e., time-dependent link flow patterns), for both on-line and off-line use.

In addition to the conceptual and algorithmic aspects of the above models, the computational issues associated with their implementation for real-time operation constitute an integral element of the problem. In particular, the development of algorithmic procedures must consider the issue of computational efficiency in novel computing architectures with varying forms and degrees of parallelisms.

3. Objectives and Structure of the Report

This report presents a review of existing traffic simulation models and dynamic network assignment formulations and algorithms in view of their ability to meet the functional requirements for operation in an ADIS/ATMS context. In addition, the report summarizes existing opportunities in terms of parallel computer architectures, as well as implications for the development of the requisite traffic modelling capabilities. Recommendations regarding the strategies for the development of simulation-assignment models are also given in light of the deficiencies identified in connection with existing models.

The next chapter presents a review of dynamic assignment models. Chapter 3 discusses existing traffic simulation models and their deficiencies with regard to ADIS/ATMS needs, as well as simulation-assignment models. Chapter 4 summarizes the review of parallel computing concepts and architectures. Chapter 5 presents the recommendations for future development of the assignment-simulation modelling capabilities.

CHAPTER 2
DYNAMIC ASSIGNMENT MODELS

1. Introduction

1.1 Advanced Driver Information Systems

Advances in communication technologies, automation, electronics, and information processing systems will increasingly be used to alleviate congestion and enhance the performance of traffic networks. These approaches, commonly known as Intelligent Vehicle Highway Systems (IVHS), envisage communication between a central controller and users on a real time basis. Under Advanced Driver Information Systems (ADIS), one of the IVHS program concepts, vehicles are provided with information on a real-time basis, about existing traffic conditions, and/or instructions on route selection from the vehicle's current location to its destination.

Hence, one of the tasks in ADIS is the assignment of O-D trip desires to the various paths on a network based on some criteria and operational constraints. Consequently, there is a need for assignment models. The bulk of the assignment models in the literature are static assignment models which assume link flows and link trip times to be time invariant. Such an assumption may be appropriate in the context of static equilibrium analysis for long-term planning applications, where the dynamic aspects of traffic behavior may not be of primary concern. However, the assumptions of uniform demand and constant trip time characteristics do not provide a reasonable approximation to the dynamic traffic behavior that occurs during the peak period. Hence, the successful implementation of ADIS, where real-time dynamics of traffic is of crucial importance, requires dynamic traffic assignment capabilities.

1.2 Why Dynamic Assignment ?

Static equilibrium assignment models assume link flows and link trip times to be constant over the duration of the peak period. Hence, they are inappropriate for real-time traffic control in congested networks where constant steady-state conditions hardly occur, especially during peak periods. In typical downtown area queueing situations, where one bottleneck highway segment can create queues stretching into other segments, static traffic assignments cannot indicate the locations and extents of queues or the delays associated with them. Because queueing can be of major importance in peak period expressway operations in downtown areas, the assignments can be grossly inaccurate in predicting peak

period operating speeds. In addition, static models have two crucial shortcomings with regard to their applicability to time-dependent problems :

1. They cannot depict the users' response to supplied real-time information under in-vehicle guidance systems. Hence, they are not helpful in gauging the behavioral tendencies of users, which is very important in the ADIS/ATMS modelling context.

2. They cannot adequately model congestion. Traffic phenomena in static models are represented using analytic link performance functions, which give the average trip time as a function of the prevailing average link flow. These violate the reality of traffic behavior, especially at high flow levels. Dynamic aspects such as congestion build-up and dissipation, queue formation and discharge, cannot be adequately represented using these link performance functions.

Hence, under real-time situations, static assignment models are inadequate. This points to the need for some models which can represent the dynamics of the system, namely, dynamic assignment models for those situations.

The state-of-the-art of dynamic assignment in a network context is in its developmental stages, for both the descriptive user equilibrium and the normative system optimal problems. Various kinds of formulations have been proposed, none of which is entirely satisfactory in terms of the underlying assumptions. Also, these formulations encounter problems with regard to their tractability and solvability in a realistic network setting. In addition, the stability and convexity requirements of these formulations pose problems indicating the complexity involved. Due to the time-dependent and large scale nature of these problems, computational difficulties arise in developing efficient network-level algorithms.

Dynamic Assignment refers to a variety of problem formulations, each of which may use a different set of decision variables and have a different behavioral basis, offering varying capabilities in terms of representing the traffic system or prescribing control actions. All of these depart in some manner from the standard static equilibrium assignment assumptions.

Dynamic assignment models can be separated into descriptive and normative models. The descriptive formulation attempts to capture how the users behave given a set of traffic conditions (user optimal). Normative models seek to determine how the system should behave in order to optimize some system-wide criteria (system optimal).

1.3 Overview

Because the field of dynamic traffic assignment is still in infancy, there are no computer packages available for solving this problem on general networks. For this reason, the review in this chapter is not of the "checklist" variety, where the features of available models are checked against the list of functional requirements identified in the Statement of Work. These have been discussed at length in the study proposal, and the discussion will not be repeated here. The starting point is therefore that the desired capabilities are not available in any existing model. The remainder of the chapter discusses the fundamental aspects of the problem, focusing on existing formulations and their limitations, and extracting some important conclusions relative to the central role of the dynamic assignment capabilities in the ADIS/ATMS context.

In the next section, the growing literature on dynamic assignment formulations is reviewed. This is followed by a description of the elements of most of these formulations. Rather than dwell on the details of all the formulations to date, a particularly representative model developed by Carey (1986) is discussed in some detail. A synthesis of issues arising in connection with this problem is given in Section 5. The discussion is further specialized to the ADIS/ATMS context in Section 6, which is followed by concluding comments in Section 7.

2. Literature Review

Dynamic assignment has attracted considerable attention in the past few years due to the nature of the problems being currently focussed upon in the transportation field. Until about five years ago, very few studies were available in this area, which contrasts with the abundance of material on static assignment. Of late, a number of problems are being approached under the label "dynamic assignment", each using different sets of decision variables and varying behavioral assumptions, thereby offering varying capabilities in terms of traffic representation and control. These models share a common feature in that they all differ in some way from the standard static assignment formulations, though none of them presently provides a tractable and robust solution in a realistic network setting. This review of the existing dynamic assignment models discusses the problems encountered.

2.1 Relevant Static Assignment Concepts

Static assignment models assume that link flows and link trip times remain constant through the peak period. Studies have shown that these formulations fail to capture the essential features of traffic congestion (Hendrickson and Planck, 1984; Mahmassani et al.,

1989; Tittemore et al., 1972; Lisco, 1983). Early approaches ignored congestion and the equilibration issue, making shortest path assignments on an all-or-nothing basis. Capacity restrained approaches resulted from a recognition of the need to incorporate congestion effects. These are heuristic approaches without theoretical backing.

The first mathematical programming formulation for the static user equilibrium (UE) problem with fixed demand as an equivalent optimization problem is due to Beckmann et al. (1956). This formulation allows the derivation of existence and uniqueness (in terms of link flows) properties of the solution, satisfying the Wardropian UE condition, namely,that no user can improve his travel time by unilaterally changing routes. Sheffi (1985) gives a comprehensive treatment of the static UE problem. More recently, the considerably more difficult problem with asymmetric link interactions has been addressed by Fisk and Boyce (1983) and Dafermos (1980, 1982), as have other variants of the basic formulation.

While UE formulations are descriptive in nature, the other major class of formulations is normative in that they seek a System Optimal (SO) link flow pattern that achieves some systemwide objective. Solution procedures for SO are identical to those for UE except that they differ in the specification of link performance functions (marginal cost functions in SO as opposed to average cost functions in UE). The difficulties involved in using static assignment to analyze route guidance systems were discussed by Boyce (1989). Ben-Akiva (1985) enumerates the shortcomings of using static models in modeling congestion.

2.2 Dynamic Assignment Models

The presence of time-dependent congestion makes the dynamic assignment problem much more difficult than the static one. Hence, initial developments have been either heuristic approaches or simulation-based. Yagar (1976) presents heuristics for achieving equilibrium flows in the dynamic assignment problem. Robillard (1974), D'Ans and Gazis (1976), Hendrickson and Kocur (1981), Midler (1969), and Hurdle (1981) all address some aspects of the dynamic problem, but in most cases limitations are introduced so as to make the problem tractable.

The bulk of the contributions to the system optimal dynamic assignment problem have addressed the situation where time-dependent flows are assigned from multiple origins to a *single destination* through the links of a network so as to minimize total system cost. The first mathematical programming approach to this problem is due to Merchant and Nemhauser (1978). Their model was formulated as a discrete-time, non-linear, non-convex mathematical program and the corresponding algorithm solved a piecewise linear version of

it. Congestion was treated explicitly using conventional link performance functions. Since the formulation was non-convex, global minimization through a one-pass Simplex algorithm was achieved through an assumption on the cost function. Ho (1980) resolved some issues involved with the implementation of the algorithm.

Carey (1987) reformulated the Merchant-Nemhauser problem as a well-behaved convex nonlinear program, which could offer mathematical and algorithmic advantages over the original formulation. In addition, Carey resolved some issues that arise from certain constraints of that formulation. Extensions were also made to handle multiple destinations and multiple commodities, though many of them remain problematic because of non-convexity issues. Also, multiple destinations require the models to satisfy a "first-in, first-out" requirement from a traffic viewpoint. This requirement, which will be discussed in depth later, creates additional constraints which complicate the formulation, and destroy many of its nice properties. In particular, such constraints lead to a non-convex constraint set (Carey, 1989).

A more recent line of work has been the use of constrained optimal control theory, which leads to continuous time formulations. The O-D trip rates are assumed to be known continuous functions of time, and the link flows are also sought as continuous functions of time. Ran and Shimazaki (1989) used this approach to develop a general model of dynamic system optimal traffic assignment for an urban transportation network with many origins and many destinations. Ran and Boyce (1990) use an optimal control theory approach to formulate a dynamic user optimal traffic assignment model. They use exit flows as a set of control variables rather than as functions, so as to overcome difficulties posed by the non-linearity of the exit flow function for multiple origin-destination networks.

Friesz et al. (1989) also discuss optimal control formulations for both system optimal and user equilibrium problems. They present the first dynamic generalization of Beckmann's equivalent optimization problem for static user optimized traffic assignment in the form of an optimal control problem. Wie (1990) extended the above traffic assignment model to include elastic time-varying travel demand which leads to the implicit consideration of departure time choices. Wie also enumerates several limitations of this approach.

Though an attractive approach, the optimal control type of formulation still suffers from many limitations, such as the lack of an explicit first-in, first-out requirement, the unrealistic modeling of traffic congestion, and more crucially, the lack of a solution procedure for general networks.

Another direction of work with feedback regulation was introduced by Papageorgiou et al. (1990). Their formulation consisted of a macroscopic modeling

framework for nonelastic but time-varying traffic demands and used a feedback methodology to establish dynamic traffic assignment conditions. A multivariable feedback regulator with integral parts and a simple bang-bang controller was developed and tested for a particular network traffic model. While the feedback feature is attractive from a control standpoint, especially for route guidance systems, the formulation does not establish the underlying mathematical basis with regard to the properties. Also, first-in, first-out requirements are not considered in the formulation. The problem is neatly stated from the control standpoint, though it leaves unanswered how some of the parameters necessary for the formulation can be estimated.

The stability of a dynamic model can be shown using a method of Lyapunov provided an assumption is made on the cost-flow function. If the cost function is monotone and smooth, this dynamic model converges to the set of Wardrop equilibria as the time passes. Smith (1984) goes through the proof by using a mathematical model and using relevant theorems.

3. Dynamic Traffic Network Model

Most dynamic network traffic assignment models include the following interacting elements:

1. Node modeling equations
2. Link models for traffic flow
3. Link model for composition rates

These are discussed in turn hereafter.

1. While considering a dynamic assignment problem, some network level constraints have to be satisfied. These include node flow conservation equations and node-link transfer balance equations for an assignment of flows to links.

2. Within a link, a traffic model is needed to transform input variables (flow into the link) into output variables (flow exiting a link). Several alternative link models can be applied for this purpose based on the physical and operational characteristics of the corresponding network link. Several authors ave used the same type of link performance functions used for static assignment. However, more realistic dynamic models are required.

3. Most of the models to date consider dynamic assignment from multiple origins to a single destination to simplify the formulation. However, for multiple destinations, there is a need to obtain the fractions of vehicles from nodes as well as links going to different destinations. This can be accompanied using composition rates which represent fractions of vehicles on a link that go to different destinations.

In the next section, the formulation due to Carey is discussed in some detail. Since the formulation considers only a single destination, the third of the above three elements does not come into effect. The second element is represented by means of exit functions that attempt to capture the flow through links. The same basic elements can be observed in the optimal control formulations.

4. A Representative Dynamic Model Formulation

Carey's formulation for the dynamic assignment of traffic is illustrated in detail. The aim of this section is to show how a typical mathematical formulation is constructed for a dynamic assignment problem, which includes an illustration of assumptions made regarding different variables, so as to obtain tractable models. It also gives an idea of the interacting modules involved in the model formulation in light of the discussion of the previous section. This model is by no means the "best" model nor is its approach the only approach to the problem (as discussed in the literature review, optimal control formulations as well as other formulations also exist). However, one area in which it is representative of the gamut of available models is with regard to the problems encountered in the formulation process. As such, this section sets the stage for the next one, which contains a detailed discussion of the issues involved in formulating such problems.

4.1 Definition of Variables

The Merchant-Nemhauser model is a discrete time, non-linear and non-convex program. As discussed earlier, the non-convexity causes analytical and computational problems. Carey develops a convex programming model for a system optimal problem. The model is described below.

Let the network be represented by a set of nodes N joined by a set of directed arcs A. Assume that there is only one destination. Let N' be the set of nodes excluding the destination node. Time is subdivided into periods of equal length, t=0, ... , T. Let,

x_{tj} = the number of vehicles on arc j at the beginning of period t.

d_{tj} = the inflow into arc j during period t.

b_{tj} = the actual outflow from arc j in period t, achieved by introducing flow controls on arc j to restrict potential outflow , which is,

$g_j(x_{tj})$ = the maximum outflow associated with arc j in period t, when the number of vehicles on the arc is x_{tj} and when arc characteristics, which may include existing flow controls, are taken as given.

4.2 The Exit Function

The function $g_j(x)$ is also referred to in the literature as exit function. It can be thought of as the flow rate when road characteristics, speed limits, traffic signals, and so forth are taken as given. However, from an operational standpoint, the concept of exit function is unclear. Exit functions are yet to be represented qualitatively or quantitatively, which is one of the disadvantages of Carey's formulation, and in fact of many other models using the exit flow concept. From the definition of the exit function, $g_j(0) = 0$ and $g_j(x) \geq 0$ for all $x \geq 0$. For the purpose of analysis, it is assumed that $g_j(x)$ is continuous for $x \geq 0$. It is also assumed that the time periods $t = 0, 1, 2, \ldots, T$ are each sufficiently short as to ensure that the outflow in a (unit) period cannot exceed the number of vehicles on the arc in that period (i.e., $x \geq g_j(x)$), though this condition need not be met, since if periods are longer than the time taken to traverse the arc then both the inflow and outflow per period can exceed the number of vehicles on the arc during that period. But, in general, one can control the time period so as to satisfy the requirement.

Additional definitions include:

$h_{tj}(x_{tj})$ = the travel cost incurred by the volume x_{tj} on the arc j in period t. The function $h_{tj}(.)$ can be assumed to be continuous, convex, non-decreasing, and nonnegative in the formulation. These are reasonable assumptions for cost functions for general traffic situations, though this reasonableness is not evident for a dynamic model.

F_{tk} = the exogenous demand (generation flow) at node k in period t.

E_j = the initial volume on arc j, i.e., $x_{0j} = E_j \geq 0$.

From the definitions, we can observe that,

$$0 \leq b_{tj} \leq g_j(x_{tj}) \text{ for all t, j.} \qquad (1)$$

This acts as a flow control or congestion control constraint. Flow controls can be used to keep the actual outflow b_{tj} below the natural or unrestricted capacity level $g_j(x_{tj})$. On the other hand, it will not be normally possible to increase the outflows above the exit levels without additional investment in arc capacity. Such investment would usually change the form or parameters of the function $g_j(x)$. The difference $s_{tj} = (g_j(x_{tj}) - b_{tj})$ is the flow control for arc j in period t.

Even if the traffic controls are not necessary nor applicable, there are substantial advantages, for analytical and programming reasons, in having the congestion control constraint as an inequality rather than as an equality.

4.3 Conservation Equations

The usual nodal flow conservation equations can be stated as

$$\sum_{j \in A(k)} d_{tj} = F_{tk} + \sum_{j \in B(k)} b_{tj} \qquad (2)$$

for each node k E N' and each period t = 0, ... , T-1; in this expression A(k) is the set of arcs pointing out of node k and B(k) is the set of arcs pointing into node k. The number of vehicles on an arc in any period is equal to the number of vehicles on the arc in the previous period plus the net inflow during the period, thus

$$x_{t+1,j} = x_{tj} - b_{tj} + d_{tj} \qquad (3)$$

for all periods t and arcs j.

The system cost minimizing, dynamic flow problem for a congested network can now be stated as follows.

4.4 The Mathematical Formulation

$$\text{Minimize} \quad \sum_{t=1}^{T} \sum_{j \in A} h_{tj}(x_{tj}) \qquad (4a)$$

subject to, for periods t = 0, ... ,T-1,

$$g_j(x_{tj}) \geq b_{tj} \qquad \text{for all } j \in A, \quad (4b)$$

$$b_{tj} = x_{tj} - x_{t+1,j} + d_{tj} \qquad \text{for all } j \in A, \quad (4c)$$

$$\sum_{j \in A(k)} d_{tj} = F_{tk} + \sum_{j \in B(k)} b_{tj} \qquad \text{for all } k \in N', \quad (4d)$$

$$x_{0j} = E_j \qquad \text{for all } j \in A, \quad (4e)$$

$$(b_{tj}, d_{tj}, x_{tj}) \geq 0 \qquad \text{for all } j \in A. \quad (4f)$$

If $g_j(x)$ has the property $g_j(x) \leq x$ for all j and all $x \geq 0$, then the constraints $x_{tj} \geq 0$ are redundant for all t and j, since the constraints $g_j(x) \leq x$, (4b, 4c, 4d) and the rest of (4e) ensure that $x_{tj} \geq 0$ for all t, j.

In this program, the constraint set (4b) is convex, since $g_j(x)$ is concave. All the other constraint functions are linear, hence the constraint set of the model is convex. Also, the objective function is convex, since $h_{tj}(x)$ is convex. Hence, any local optimum of the model is the global optimum, and if $h_{tj}(x)$ is strictly convex, then any local optimum will be the unique global optimum.

Since the model is a convex program, any of the several well-known non-linear programming algorithms can be used to solve it. Alternatively, one could take a piecewise linear approximation, by piecewise linearizing the functions $g_j(x)$ and $h_{tj}(x)$. The resulting formulation would then be solved by the simplex algorithm. However, the author did not provide numerical illustration nor implement any actual solution algorithm.

5. Issues Involved in Dynamic Assignment Formulations

A number of issues arise while formulating dynamic assignment models with regard to the form of the model and assumptions made. These are discussed in detail below.

5.1 Multiple Destinations and the Non-Convexity Issue

One of the constraints from the traffic viewpoint is a "first-in, first-out" (FIFO) requirement. This requirement creates a central difficulty for models of dynamic traffic flows on networks.

The FIFO requirement states that the traffic which embarks on a road or other facility in period t exits from that facility, *on average*, before traffic which enters in any later time periods. This represents the physical behavior of traffic.

The FIFO requirement does not cause a problem in the static traffic assignment. But in the dynamic case, it yields a non-convex constraint set, especially if multiple destinations or multiple commodities exist. In the single destination case, all vehicles on a link are going to the same destination. Hence, vehicles do not have advantage over one another in terms of path allocations, whereas vehicles travelling to different destinations face different levels of congestion based on the demand between the various O-D pairs. Carey (1989) considers various formulations, each of which yields a non-convex optimization problem which is at present computationally tractable only for relatively small-scale examples. The problem arises regardless of whether congestion occurs or not, or whether SO or UE models are being considered. On a road network, traffic of different types which enters the same arc at approximately the same time will usually travel at the same speed. Individual vehicles do travel at different speeds and do pass each other. But in modeling aggregate flow, the normal assumption is that vehicles which are bound for different destinations, and which enter an arc at the same time, will take (approximately) the same time to traverse the arc. Stated in other terms, traffic which enters an arc first will "on average" exit first, i.e., FIFO.

Why is FIFO problematic from a mathematical programming standpoint? It is easy to construct traffic network flow examples where total travel costs would be lower if some

traffic types could be temporarily held back on an arc, while allowing some other traffic types to proceed to later arcs. From an algorithmic and logical viewpoint for a minimization problem, this would be correct. However, in real world traffic conditions, it is generally not physically possible to do this, especially when congested conditions occur (during peak period of traffic flow or when incidents occur). When congestion occurs, this form of passing is generally not possible. In addition, holding back vehicles is not realistic.

An example scenario with multiple destinations arises when the path to one destination is less congested than the path to another destination, but the two paths share some common links. In such a case, the program might like to assign some vehicles following the less congested path but presently on a common link to the next link on their path, while holding back those vehicles on that same common link but following the congested path. If the vehicles bound for the destination on the less congested path are, on average, behind the held-back vehicles, the reality of traffic behavior dictates that they cannot move ahead. This points to the need for an explicit FIFO requirement in the formulation to preclude such situations.

Carey (1989) has explored this problem, proposing one additional mathematical equation to impose the FIFO requirement. However, this constraint makes the feasible set non-convex, thereby destroying much of the computational and analytical advantages of the formulation. Hence, he suggests first solving the network flow model without introducing any explicit FIFO restriction, to see how or where FIFO is violated and the seriousness of the violations. However, he does not propose an explicit procedure to do so.

Additional research is necessary in dealing with this problem. Especially important is the need to circumvent the non-convexity problem, so as to develop robust formulations.

5.2 Exit Functions

A majority of the existing dynamic assignment models make use of the so-called link exit functions to represent congestion and/or as a means of control for the flow exiting a link. Generally represented as $g_j(x_{tj})$, it is the flow which can exit a link j in a time period t given x_{tj} is the number of vehicles on link j at the beginning of period t. For a road network, $g_j(x)$ is the flow rate when the road characteristics, speed limits, traffic signals, and so forth are taken as given. In other words, $g_j(x)$ represents the existing capacity outflow of a link and, hence, captures congestion on the link. Thus,

$$g_j(0) = 0 \qquad , x = 0$$

$$\text{and} \quad g_j(x) \geq 0 \qquad , x \geq 0.$$

Assumptions are often made on the exit function to make the corresponding model tractable. One set of assumptions typically made is,

(1) $x_{tj} > g_j(x_{tj}) > 0$, for $x > 0$,

(2) $1 > g'_j(x_{tj}) \geq 0$, for $x > 0$,

(3) $g_j(0) = 0$

The property, $x > g_j(x)$ states that the exiting volume in a given period t should not exceed the number of vehicles on link j at the beginning of period t. This property can be ensured by assuming sufficiently small time periods, t = 0,...,T. This property, along with a concavity assumption for $g_j(x)$, ensures the satisfaction of the Kuhn-Tucker conditions (a constraint qualification is satisfied) and ensures global optimality of the solution for single-destination problems.

The slope $g'_j(x)$ is constrained between the values 0 and 1. The upper limit restates property (1) for a concavity assumption, though this is not a necessary assumption. The lower limit prevents oversaturation associated with a downward sloping g(x). Otherwise, with a downward sloping g(x), flow controls can be used to increase outflows g(x) by reducing x.

Though the exit function captures some congestion effects due to its capacity constraining nature, it fails to properly capture the dynamics of the process. The function $g_j(x)$ could additionally be used as a tool for flow control, if desired, by restricting the actual outflow from a link to be below the capacity flow or exit flow. Hence, it will not be possible to increase the outflow above $g_j(x)$ without additional investment in the link capacity—and such investment might change the form or parameters of $g_j(x)$.

The concept of link exit function, though attractive from an analytic viewpoint, suffers a number of weaknesses in the manner in which it is presently envisaged. Firstly, an exit function concept is unclear from an operational standpoint. No accepted models or functions have been formulated nor are there sufficiently specific suggestions in the literature on how they can be captured robustly. By definition, an exit function is necessary for each link. This raises computational issues as well as issues of consistency in obtaining the exit functions for the various links.

Secondly, a number of models, Carey (1986), Ran and Shimazaki (1989a), Friesz et al. (1989), make assumptions about the exit functions. General assumptions of non-negativity, differentiability, concavity and boundedness are made so as to achieve tractability of the models as well as to obtain attractive analytical properties with regard to the models. These assumptions act as constraints on the modeling of the exit function and may deviate from reality in some situations.

Serious difficulties arise with respect to defining meaningful exit functions for the case of multiple destinations (Carey, 1988). For non-linear exit flow functions, it is very difficult to establish a dynamic generalization of the static model for the network with

multiple O-D pairs. Linear exit function assumptions are necessary to keep the property of separability which is crucial in showing equilibration of instantaneous unit path costs. Hence, it appears that only linear exit flow functions will satisfy the first-in, first-out criterion at nodes (Wie, 1988). This difficulty led to models avoiding exit functions (Ran and Shimazaki, 1989b) as well as formulations (Ran, Boyce and LeBlanc, 1990) where exit flows from links have been assumed as control variables rather than as functions in order to establish dynamic equivalency of the static model using optimal control theory.

Papageorgiou et al. (1990) present a good discussion from the control viewpoint, using independent splitting rates of flows from nodes to links (i.e., flows entering links) rather than exit functions as control variables. This approach is especially interesting for the ADIS/ATMS context, where the splitting rates are tools, both for gauging user behavior and exercising control. However, this model too suffers from an operational point of view, as it is unclear how some of the variables can be obtained for a real system.

In summary, the construct of an exit function has been one of the primary limitations on the relevance of exiting dynamic assignment formulations to real problems.

5.3 Path-based Approach and Lack of Uniqueness

For the ADIS/ATMS context, path-based assignments are ideal because the central controller has information with regard to O-D trip desires and will assign various paths to different O-D pairs based on some criteria. However, almost all models in the literature for the dynamic problem are link-based. That is, the mathematical formulations, for both the SO and UE assignments use link flows as the variables being solved for. Though there exists computational difficulty and inconvenience due to utilization of a link-based approach, path enumeration destroys some of the nice properties of the formulation, thereby making unclear the tractability of the path-based approaches.

Flows on arcs and paths can be related to one another through equations known as the path-arc incidence relationships. In the static assignment case, these relationships are easy to establish because flows on a path are assumed to exist simultaneously all along the path. If x_a and t_a represent the flow and travel time, respectively, on link a, ($a \in A$), $t_a = t_a(x_a)$ where the link congestion function $t_a(.)$ represents the relationship between flow and travel time for link a. If f_k^{rs} and c_k^{rs} represent the flow and travel time, respectively, on path k connecting origin r and destination s ($k \in K_{rs}$), the travel time on a particular path is the sum of the travel time on the links comprising this path. Mathematically stated,

$$c_k^{rs} = \Sigma\, t_a\, \partial_{ak}^{rs}$$

where

$$\partial_{ak}^{rs} = \begin{cases} 1, \text{ if link a is part of path k connecting O-D pair r-s.} \\ 0, \text{ otherwise.} \end{cases}$$

Using the same indicator variable, link flow can be expressed as a function of the path flow, namely,

$$x_a = \Sigma\Sigma\Sigma\ f_k^{rs}\ \partial_{ak}^{rs}, \text{ for all } a \in A\ ,$$

the summation performed over all r, s and k. This means that the flow on each arc is the sum of the flows on all paths going through that arc. By contrast, the dynamic link-path incidence relationships are not trivial. In the dynamic problem, vehicles assigned to a path at a given time are not simultaneously present on all links forming that particular path. Therefore, link-path incidence relationships must recognize the time at which vehicles are actually loaded onto a particular path.

For the static equilibrium problem, the convexity of the objective function as well as of the constraint set for the link based formulation can be readily proved. This implies that the UE program has a unique minimum, thereby establishing the uniqueness of the link flow pattern that solves the program. The same cannot be said for the path flows. The program is not convex with respect to path flows and, therefore there may be many path flow patterns which are consistent with the unique equilibrium link flows. Hence, many existing dynamic formulations which assume convexity of cost functions and of the constraint sets in obtaining tractable models fail to establish uniqueness when path-based approaches are used. In fact, there can be a number of path flows satisfying the optimality criterion. The non-convexity causes analytical (lack of uniqueness) and computational problems, as well-known algorithms developed for convex programs no longer hold.

5.4 Link Interactions

Mathematical programming formulations generally assume that travel time on a given link depends only on flow through that link and not on the flow through any other link. This assumption fails in reality when heavy traffic occurs on two-way streets, unsignalized intersections, and left-turning movements in signalized intersections. In such cases, link interactions cannot be ignored.

Link interactions can be either symmetric or asymmetric. When interactions are symmetric, the marginal effect of one link flow, say x_a on the travel time on any other link, say t_b, is equal to the marginal effect of x_b on t_a. When link interactions are asymmetric,

there is no known equivalent minimization program that can be used to obtain the equilibrium flow pattern.

Almost all models to date on dynamic assignment problems avoid considering link interactions in their problem formulation as they lead to a much higher degree of complexity. Even in the static case, only in the past decade have approaches like variational inequality formulations for the asymmetric link interactions been discussed—Fisk and Boyce (1983). Computational aspects of this problem have been investigated by Nagurney (1984, 1986) and Mahmassani and Mouskos (1988, 1989).

6. The ADIS/ATMS Context

As discussed in Chapter 1, both descriptive and normative assignment capabilities are needed in the ADIS/ATMS context. This section assesses the formulations described in this chapter from the standpoint of the ADIS/ATMS needs.

6.1 System Optimal and User Equilibrium Formulations

Putting aside the obvious limitations of the formulations reviewed in this chapter with respect to the representation of traffic processes, the dynamic UE formulations are not of immediate relevance to the real-time assignment needs in the ADIS/ATMS context. Conceptually, the UE formulations could describe some long term equilibrium that might develop after real-time information has been available for some time. As such, dynamic UE formulations do not meet the descriptive needs of the ADIS/ATMS context, as their value for operations and control purposes is quite limited.

On the other hand, the system optimal formulations have more direct relevance, *conceptually*, to the ADIS/ATMS needs for a normative tool that a central controller could use to determine how vehicles should be routed through the network. This would be applicable in a centrally controlled network in which the controller has complete information as to the O-D trip desires and users comply with the controller's directives. However, to the extent that not all users may comply with the controller's directives, and/or not all are equipped with devices for real-time communication, a formulation combining SO objectives for a subset of the vehicles and UE for the others may be appropriate. Such a formulation would not be very meaningful for real-time assignment decisions by the central controller because the very assumption of long-term equilibrium is not likely to be applicable for operational control purposes. On the other hand, the SO formulation may be of direct relevance to the central controller's functions though only under relatively stringent assumptions. Nevertheless, it provides a starting point for alternative

formulations that would explicitly recognize the complexities of the problem and the different informational situations under which the central controller may be operating.

6.2 How to Incorporate Real-time Information ?

The various formulations discussed previously illustrate the variables likely to be involved in the dynamic assignment problem. In the ADIS/ATMS context, it will be necessary to identify at least two classes of users; those with information (equipped users) and those without. Ideally, a controller with complete information will know how many vehicles for a given O-D pair are equipped. On the other hand, the controller may have only partial information, in which case, probabilistic models might be used to obtain the fraction of vehicles equipped for given O-D trip desires in a given time period t.

Users with information are those who will receive guidance instructions from the controller. Modelling users with and without information constitutes an area of strong interface between the normative and descriptive assignment capabilities needed in the ADIS/ATMS context. Essentially, the system optimal problem must be solved in real-time to obtain assignment decisions for the guided vehicles, but the value of the objective function depends also on the decisions made by the unguided vehicles. The two classes are strongly interacting, since unguided or noncompliant vehicles are influenced in their path choices by perceived traffic conditions, which clearly depend on the actions of the guided vehicles as well. Recognizing this dynamic interdependence further complicates an already formidable problem.

Another important question is that of the time horizon over which the optimal assignment is made. Ideally, an infinite horizon over which future traffic conditions are predictable is desirable while making optimal assignments. However, this is neither computationally desirable nor consistent with the rationale for ADIS/ATMS, which are intended to provide superior performance through real-time knowledge of traffic conditions, on the premise that traffic conditions in the network are not entirely predictable due to the presence of incidents, variable demands and stochastic effects.

If "optimal" assignments are made in a given time period, changing traffic conditions may prevent these paths from being optimal at a later time. How do we ensure that the "optimality" is maintained for a "reasonable" time period? That is, how do we ensure a "look-ahead" feature?

One possible approach to this issue is the use of the well-known rolling horizon approach (Wagner, 1975). It enables us to utilize only available flow data, obtained in real time, while at the same time preserving the performance of the computational procedure. Essentially, it consists of a projection horizon of k time steps—the period for which traffic

flow information is necessary. A real-time set up provides actual data for a near term period of r steps at the 'head' of the stage. For the next (k-r) time steps, the 'tail' of the stage, flow data is estimated from a model or from the data collected during previous projection horizons. An optimal assignment is obtained for the entire stage, but only implemented for the head section. The projection horizon is then shifted (rolled) by r-units ahead. New data for the stage (head and tail) is obtained and the process is repeated. Gartner et al. (1983) use this approach to obtain a demand-responsive strategy for traffic signal control. Powell (1987) uses the same approach for the dynamic vehicle allocation problem.

6.3 Defining the Control Variables

Most of the existing formulations reviewed in the previous sections use the exit function concept to capture traffic congestion effects. As already discussed, this exit flow function can be used as a flow control by requiring the actual flow to be less than or equal to the exit flow on a link. Ran and Boyce (1990) use the exit flows as a set of control variables in their optimal control theory formulations.

Another approach (Papageorgiou et al.,1990) envisages the independent splitting rates as control variables. Here, the independent splitting rates refer to the way flows going to a particular destination are split at a node among the various links going out of that node along paths leading to the desired destination.

The independent splitting rate ß ($0 \leq \beta \leq 1$), at each node reflect the drivers' route decisions, in response to the supplied information. There are two interesting issues that arise in connection with using ß :

(a) Modelling : How should ß be calculated in the absence of any information to drivers so as to capture the user behavior?

(b) Control : What is the best choice for the value of ß, at each node, if it can be manipulated through provision of real-time information?

Lack of complete compliance can result in the actual ß being different from the splitting rate advised by the control system. Then, a parameter μ, $0 \leq \mu \leq 1$ reflecting the compliance rate can be used to obtain the actual splitting rate β_r.

6.4 System Optimal Dynamic Assignment Models for ADIS Context

Existing dynamic assignment models, like the mathematical programming models and optimal control formulations are deficient in several aspects with regard to the ADIS/ATMS functional requirements. Among the various shortcomings observed in these models are:

(1) They are mostly link-based formulations. The advantage of link flow representation is that it involves fewer variables, though the conservation conditions sometimes cause awkward mathematical expressions. In the ADIS framework, we have a central controller exercising a normative assignment capability by directing users in some manner and assigning them to optimal routes (or paths) in the network. The advantage of path flow representation is that the conservation conditions are automatically satisfied, though the number of paths may be enormous. In a path-based representation, the controller can assign users to various paths on the network and/or guide them on a real-time basis.

(2) Most either consider single origin, single destination scenarios or multiple-origin, single destination scenarios in order to attain certain desirable analytical properties, as well as to avoid physical constraints like first-in first-out (FIFO) requirements.

(3) They do not keep track of the entry time of vehicles into the system, and therefore cannot consider the experiences of individual vehicles as a result of the assignment made by the controller.

(4) They have objective functions dependent on link flows. If the objective is the minimization of total system travel time, the dynamic case can pose some tricky problems. In the static assignment problems, link performance functions are used to obtain link travel times. However, in the dynamic case, these link performance functions are of little use because the need is to capture dynamic phenomena like queue formation or congestion buildup, and link travel times are difficult to obtain as flows are not ergodic (i.e., time and space averages of flows no longer coincide), unlike the static case. Here the flow moments must be computed over repeated simulations of transients. Simply stated, the contribution of a particular O-D flow presently on some link to that link's present travel time is difficult to compute especially as flows are not identical over the length of the link and dynamic phenomena like queuing exist. In static assignment, this is not a problem because flows are assumed identical along the length of a link, as well as identically and simultaneously existing along links of a desired O-D path (because of the assumption that there can be no storage of vehicles on links). Hence, the link-based travel objective functions could be trivially obtained in the static case. Using similar objective functions in the dynamic case makes the dynamic problem much more difficult than if path-based objective functions are used, especially from the ADIS viewpoint.

6.5 Issues Related to Path-Based Dynamic Assignment Formulations

Ideally, a central controller would like to assign a user with a particular O-D desire to some path taking into account systemwide objectives. In the static situation, it is easy to

do this while keeping track of the system state (link flows, link travel times) because of assumptions that flows on a path are identical and exist simultaneously all along the path. Hence, there exists a static-link path incidence matrix, which shows the relationship between path of arc flows.

By contrast, the dynamic link-path incidence is not trivial. In the dynamic case, vehicles starting along a path at a given time are clearly not simultaneously present on all links of the path. Hence, the incidence matrix changes dynamically depending on the present location of the vehicles assigned to a given path at a particular time. To dynamically keep track of assigned vehicles is difficult especially in congested conditions. Cascetta et al. (1991) describe a dynamic network loading model which can be used to obtain a dynamic link-path incidence relationship. The advantage of having dynamic incidence matrices is that the actual travel time of individual (or groups of) vehicles in the network can then be easily obtained. This means that the actual or predicted travel times on various paths would be available to the controller, and thus be useful to make assignment decisions. Another advantage of dynamic incidence matrices is that keeping track of the position of vehicles, even on the links, prevents a violation of the FIFO requirement. This is done by keeping track of the entry time of vehicles into the system as well as the present locations of those vehicles.

6.6 Information Availability Scenarios for the Controller

For a normative assignment capability, the best scenario would be an 'ideal' fully informed system, where the central controller knows all there is to know about eery tripmaker in terms of origin, destination and timing of the trip, and use this information to develop a coordinated scheme that assigns to each vehicle a path to its destination, so as to achieve some desirable system-level objectives. These functional capabilities are envisioned to be available at the so-called "coordinated stage" of ADIS development (Mobility 2000, 1990a). As discussed below, many variations from this scenario are possible, and several may be more likely.

It is unlikely that a controller will have full information regarding O-D trip desires for the complete duration during which assignment is to be made. More likely, information may be available for a short duration into the future through detectors and advance information from drivers. This provides an ideal opportunity to use a rolling horizon approach with forecasted future O-D desires to come up with models for normative assignment.

An even more likely scenario is that the controller has O-D trip desires only for the present period. In this case as well as for the previous non-deterministic case, there must

be provision for a "look-ahead" feature, which could foresee future O-D desires and path assignments. Assignments based on present conditions alone may lead to worse outcomes than anticipated.

In the absence of information regarding future O-D trip desires, the controller can assume known probability distributions for these O-D desires and obtain stochastic formulation with "look ahead" features. Incidents can also be assumed to have known distributions based on the type, duration and locations of these incidents.

These questions will be address in future tasks of the present study.

7. Conclusions

The area of dynamic network traffic assignment is still in its early stages of development, for both descriptive user equilibrium and the normative system optimal problems. The state-of the-art is clearly fragmented into several lines of work, none of which is entirely satisfactory in terms of the underlying assumptions, or in terms of tractability and solvability in a realistic network setting. Key weaknesses remain in terms of incorporating user decisions, and in terms of representing the dynamic traffic phenomena which are of essence in congested networks. Nevertheless, there has been much progress in the past five years, and there are interesting ideas and directions that offer some promise.

The implications for the present study are that no formulations, models or algorithms presently satisfy the functional needs of the ADIS/ATMS context. In addition, a strong interface between the descriptive and normative capabilities is critical to the success of these systems. Because of the nature of the problem, realistic representation of traffic behavior is essential. For this reason, a simulation approach will be necessary to model network performance, and thereby to evaluate the objective functions. The use of a traffic simulation model for the performance side automatically insures consistency with the FIFO requirements discussed earlier in the section.

Another important conclusion is that the development of formulations for the dynamic SO assignment problem should be a high priority item as the study team addresses Task B of this study.

CHAPTER 3
REVIEW OF EXISTING TRAFFIC SIMULATION MODELS

1. Introduction

This chapter presents a review of existing traffic simulation models in view of their applicability in the ADIS/ATMS context. First, the functional requirements for this modelling capability are reviewed. This is followed by a summary review of the principal model packages relative to the functional requirements. The CONTRAM model, a simulation-assignment program, is discussed in more detail in Section 4, as an illustration of traditional attempts at heuristic simulation-assignment modelling. Concluding comments are given in Section 5.

2. Functional Requirements

In the ADIS/ATMS implementation, it is expected that an effective real-time traffic simulation model will be used for assessment of various route assignment strategies, and projection of future network traffic conditions. Hence, the traffic simulation program shall be capable of responding and completing the requested task in a sufficiently short time for real-time operation.

Furthermore, a real-time traffic simulation model developed for use in connection with ADIS/ATMS should have comprehensive capability to simulate an integrated network with both freeways and surface streets. More specifically the real-time simulation models should have the following functions:

A. *Functional Requirements for Surface Streets:*

A.1 It shall have a realistic representation of changes in traffic signal control.

A.2 It shall be capable of modelling various types of link geometric configurations.

A.3 It shall have a precise simulation of traffic related effects.

A.4 It shall have a concise but precise measurement of the system effectiveness on both a link specific and network-wide bases.

A.5 It shall be capable of simulating trip generation and attraction centers, and bus operations as well as related facilities.

B. *Functional Requirements for Freeway Systems:*

B.1 It shall have a realistic representation of traffic characteristics and geometric configurations.

B.2 It shall have a detailed simulation of ramp flow characteristics and traffic control strategies.

B.3 It shall provide a concise output statistics for measurement of the effectiveness on both a link specific and network- wide bases.

C. *Special Functional Requirements for Use in ADIS Systems:*

C.1 It should be able to simulate traffic flows at the individual vehicle/driver level.

C.2 It should be able to model the route choice behavior of drivers with and without the access to ADIS systems.

C.3 It should be capable of simulating different sections of the network at different levels of detail, ranging from microscopic to macroscopic.

C.4 It shall be capable of accepting data from both the surveillance and historical traffic information at a user specified time period.

C.5 It should be responsive to dynamic O-D information reported by the ADIS system.

C.6 It shall be able to track the route and location of each driver who accepts the route advice from the control center.

C.7 It shall be able to predict future impedance based on the assignment results, and feedback to the control center to determine what fraction of flows need to be reassigned.

3. Review

Based on the above functional requirements, we have reviewed both simulation models for freeways and composite networks and simulation/assignment models in the literature. The evaluation results of simulation models for freeways and composite networks are summarized in Table 1. Table 2 contains evaluation results for simulation assignment models.

TABLE 1: FUNCTIONAL REQUIREMENTS EVALUATION OF SIMULATION MODELS FOR FREEWAYS AND COMPOSITE NETWORKS

REQUIREMENTS	MACK-FREFLO-FRECON	INTRAS	FREQ	SCOT
A.1	No	No	No	Yes
A.2	No	No	No	Yes
A.3	No	No	No	Yes
A.4	No	No	No	Yes
A.5	No	No	No	No
B.1	Yes	Yes	Yes	Yes
B.2	Yes	Yes	Yes	Yes
B.3	Yes	Yes	Yes	Yes
C.1	No	Yes	No	No
C.2	No	No	No	No
C.3	No	No	No	Hybrid*
C.4	No	No	No	No
C.5	No	No	No	No
C.6	No	No	No	No
C.7	No	No	No	No

* hybrid denotes models that use "mesoscopic" approach or combine different approaches for different components of the traffic system.

TABLE 2: FUNCTIONAL REQUIREMENTS EVALUATION OF SIMULATION/ASSIGNMENT MODELS

REQUIREMENTS	CORQ	CONTRAM	SATURN	TRAF	DYNEV	INTEGRATION
A.1	No	No	Yes	Yes	No	Yes
A.2	No	Yes	Yes	Yes	No	Yes
A.3	No	No	Yes	Yes	No	No
A.4	No	Yes	Yes	Yes	No	Yes
A.5	No	No	Yes	Yes	No	No
B.1	No	No	No	Yes	No	Yes
B.2	Yes	No	No	Yes	No	Yes
B.3	Yes	No	No	Yes	No	Yes
C.1	No	Yes	No	Yes	No	Yes
C.2	No	No	Yes	No	No	Yes
C.3	No	No	Yes	Hybrid*	No	No
C.4	No	No	No	No	No	No
C.5	Yes	Yes	No	No	No	Yes
C.6	No	Yes	Yes	No	No	Yes
C.7	Yes	Yes	No	No	No	Yes

* hybrid denotes models that use "mesoscopic" approach or combine different approaches for different components of the traffic system.

4. CONTRAM

4.1 Model Features

CONTRAM (Leonard et al., 1989) is a traffic assignment model developed by the Transportation and Road Research Laboratory (in the U.K.) for use in design of traffic management schemes in urban areas. It predicts flows, queues and routes of vehicles as they travel through an urban network. It is a capacity-restrained model, which takes account of the interactive effects of traffic conditions. Some of the principal features of CONTRAM include:

1. The calculation of route trip times includes junction delays.
2. Three types of vehicles can be defined in CONTRAM: cars, buses and lorries.
3. Junction types include signal controlled, major/minor junction and roundabouts.
4. Time variation is modelled by subdividing the period being analyzed into consecutive time intervals.
5. The basic unit for assignment in CONTRAM is a packet. Each packet is assigned independently to its route through the network in an 'incremental' form of loading.
6. An iterative procedure is adapted for the convergence of CONTRAM.
7. Three types of signal operation can be modelled: fixed cycle/fixed split, fixed cycle/optimized splits and optimized cycle/optimized splits.
8. Fuel consumption estimates can be made for each class of vehicles and origin-to-destination speeds.

4.2 Structure of Model

The overall structure is shown graphically, in Fig. 1. There are three important parts: input, assignment and output. Input data includes traffic demand data (time varying), network data and control data. In the assignment mechanism, the minimum journey route is determined for assigning a given packet, which can then be used to determine flows and queues on links. The minimum journey route is then recalculated for the next packet.

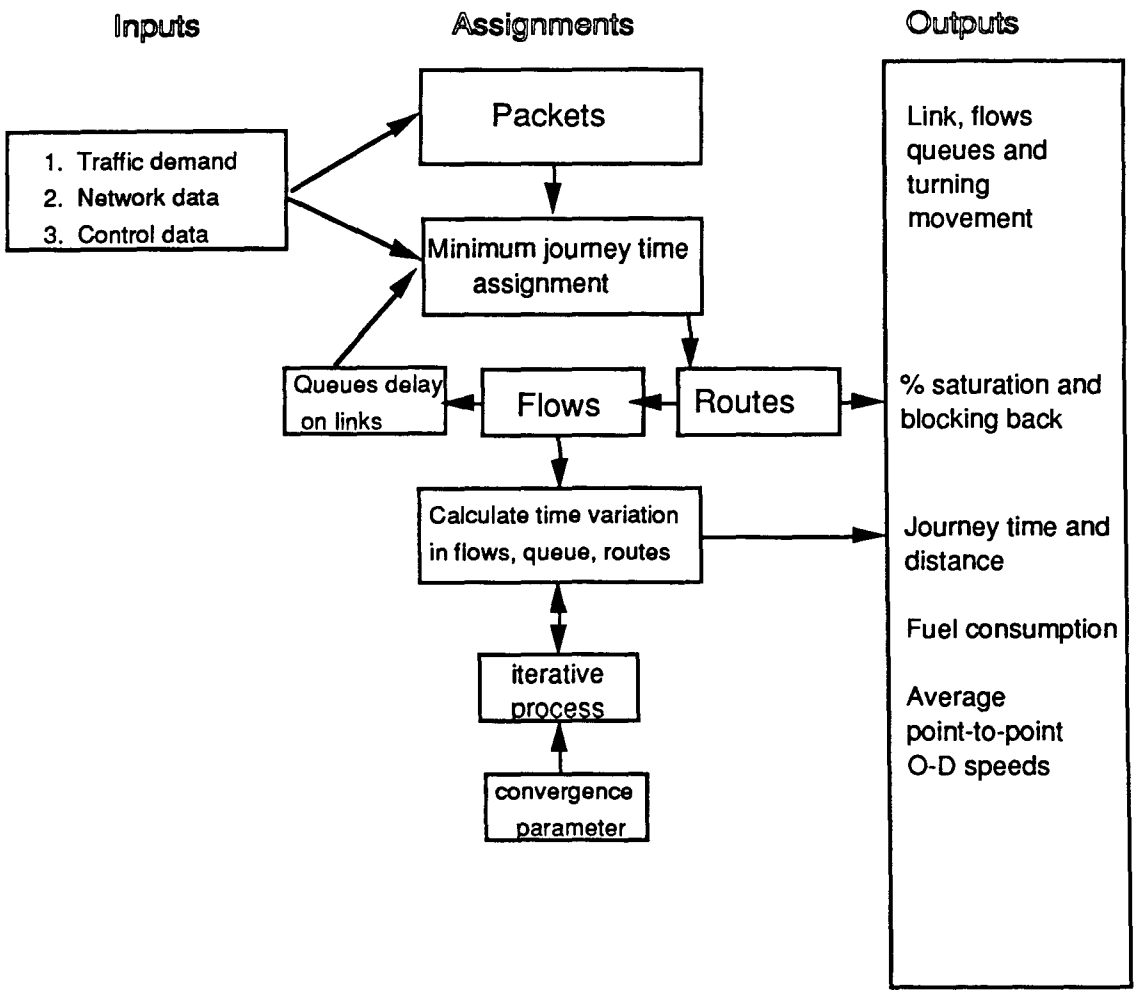

Fig. 1 The Overall Structure of CONTRAM

The period to be modelled is subdivided into a sequence of time intervals, which need not have the same length. Time variation in the input demand data is achieved by using different demand values in each time interval. Within each interval, vehicles for each O-D pair are handled in groups called 'packets'. The progression of packets through a network is approximated using statistical and queuing relations between the flow increments.

For the purpose of assignment, all of the vehicles in a given packet are assigned to the same minimum journey time route and each packet is assigned independently to its route.

The journey time is divided into the cruise time and the time delayed at the downstream stopline along the link. The cruise time is the free run time taken by a vehicle travelling at the average speed of traffic along the full length of the link. Therefore, the

time is the same for all vehicles of the same class. In CONTRAM 5, the cruise time is obtained by speed/flow relationships instead of fixed cruise time.

The delay time is the time taken to discharge the queue of vehicles encountered by the packet at the time that the packet reaches the downstream stopline after travelling the full length of the link. While computing the delay time, time dependent queuing formulae are used to estimate the queue at any point in a time interval. These formulae take into account: the type of link, the initial queue on a link, the vehicle arrival and departure rates, randomness in the arrival and departure patterns and the length of the time interval.

The packets are assigned to their minimum journey time routes by an iterative procedure. After the first packet loading, the sequence of operations for assigning each packet is:

1. Remove the increment of flow in the appropriate time interval.
2. Recalculate the queues on links affected by the previous route of the packet.
3. Assign the packet to its new minimum journey time route.
4. Add the flow due to the packet to links on the new route and recalculate the queues affected by the new route.
5. Take next packet and repeat steps 1 to 4.

The iterative procedure is continued until satisfactory convergence has been reached. Ideally, this occurs when all packets are reassigned to the same routes in consecutive iterations. However, practical criteria for convergence for a given network are typically met in about 5 to 10 repeated iterations. Note that a new version called CONTRAM.ROGUS is intended for ADIS and other IVHS applications.

While it goes further than other simulation or assignment models in combining simulation and assignment capabilities, CONTRAM still suffers from major shortcomings relative to the ADIS/ATMS context. It is only quasi dynamic in that it treats time-dependence only through different relatively long intervals. Its network path processing capabilities are very basic, and do not go beyond shortest path calculation. Only one path is considered for any particular packet. In addition, the behavioral content is very limited.

5. Conclusion

Clearly there is no existing simulation assignment model that fully meets the functional requirements for use in real-time traffic simulation. Furthermore, the basic concept for the proposed simulation model will emphasize the fast-execution feature for

real-time applications. Hence, the computational efficiency will be an integral factor in the design process. Recent developments in computer architectures and supporting software have offered significant opportunities for the fast solution of large-scale problems. The application of parallel computing hardware architectures, programming language, and optimal compilers should overcome traditional size limitations imposed by scalar mainframe computers. Some of the commercial parallel computers are reviewed in the next chapter.

CHAPTER 4
REVIEW OF ADVANCED PARALLEL COMPUTERS

1. Introduction

It has become apparent to most computer researchers that the most promising long term approach to achieving affordable, accessible supercomputing is parallel, or concurrent processing. A concurrent machine uses multiple, interacting processors to perform many operations at once. To be solved on such a machine, a problem must be broken down into a number of pieces that can be run concurrently on more than one processor. Fortunately, this fundamental structure is a common characteristic of a wide variety of applications, particularly in scientific computing.

Parallel computers can be classified into two general categories:

- Single-instruction, multiple-data (SIMD) stream architectures
- Multiple-instruction, multiple-data (MIMD) stream architectures

SIMD machines, representing the simpler of these two classifications, use either the array processor or the pipeline approach. MIMD machines, however, breakaway completely from the von Neumann architecture, representing the leading edge of research. Most existing dataflow architectures follow the MIMD approach.

2. SIMD Architectures

Single-instruction, multiple-data stream architectures use a single instruction to act on many sets of data simultaneously. This architecture, also called an "array processor," features a control unit, multiple processors, multiple memories, and an interconnection network. The control unit broadcasts instruction to the all processors but only active processors execute the same instruction at the same time. Each processor executes the instruction using data taken from its local memory. The interconnection network allows interprocessor communications. There are essentially four methods of interprocessor communications: shared variables, message passing, marker passing, and value passing. The Connection Machine, a SIMD computer, uses marker/message passing method.

3. MIMD Architectures and Dataflow

The multiple-instruction, multiple-data stream architectures consists of multiple processors and multiple memories, where each processor can follow an independent instruction stream. While offering advantages of flexibility, synchronization is a design challenge using MIMD machines. Most advanced research projects involve various MIMD

approaches. Many tightly coupled multiprocessors use shared memory as a major means of communication between processors such as the Encore Multimax multiprocessor. The iPSC and nCUBE which are loosely coupled, distributed memory multicomputers, use message passing communication mechanism.

The dataflow concept was developed in 1967 at MIT. Essentially, dataflow architectures permit asynchronous operations to fire whenever the required dataflow becomes available. In this decentralized architecture, instructions and data are sent together to many equivalent processing elements. The principle feature of dataflow hardware is the use of finegrain instruction-level parallelism. This parallelism is implemented by message passing, and the execution of instructions is data-driven. Instructions by a processing unit are executed only when requisite "tokens" tagged onto arriving data indicate that all required data have been received

4. Comparison of SIMD and MIMD Computer Architectures

The two main variations of computer architectures are in the number of processors and the speed of each processor. On one end there are computers like CRAY Y-MP which have a few very powerful processors (coarse-grain), while other machines such as the Connection Machine may have a large number of simple processors (fine-grain).

Most coarse-grain machines available in the market are MIMD computers. In contrast, most fine-grain machines are SIMD machines. Nearest-neighbor type aerophysics problems are ideally-suited to SIMD computers, as are image-processing problems. On the other hand SIMD model restricts programmers by requiring that all processors execute identical programs in a lock-step fashion.

One common problem encountered on a coarse-grain computer is the load-balancing assignment, because the execution time will correspond to the time required by the slowest processor. On coarse-grain computers, algorithm improvements may not be significant because the focus is on control parallelism. For most engineering applications, the number of independent subunits that can be broken down is typically measured in tens. However, they do offer flexibility and are more general-purpose computers.

There are several massively parallel MIMD machines under development such as IBM TF-1 which is planned to have up to 32K processors and be capable of 1.0 TeraFLOPS. However one of the main problems with massively parallel MIMD computers is that with current operating system and compiler technology they are extremely hard to program effectively. Hence major breakthroughs in software technology will be required before the full power of massively parallel MIMD computers can be realized.

Currently available commercial parallel computers can be grouped into following three classes:

I. Distributed memory MIMD multicomputer	 INTEL iPSC family nCUBE 2 family
II: Shared memory MIMD multiprocessor	 Alliant FX family BBN TC2000 Encore 90 Series FPS Computing System 500
III: Distributed memory SIMD	 Thinking Machines CM-2

5. Intel Corporation, Supercomputer Systems Division

The iPSC/860 is a distributed memory MIMD computer with message-passing architecture. The iPSC/860 performs its numeric computations on specialized processing nodes. Each numeric node has its own CPU, kernel operating system, local memory, and I/O facilities. The CPU is based on a 40 MHz Intel i860 microprocessor, each contains a 64-bit RISC processor, a floating point adder, a floating-point multiplier, and 8 and 16 Kbyte caches for instruction and data. The iPSC/860 are scalable from 8 processors to 128 processors, with peak performance ranging from 480 MFLOPS for an 8-processor system to 7.6 GFLOPS for a system with 128 processors. Memory is scalable ranging from 64 MBytes to 8 GBytes.

The iPSC/860 nodes communicate via Intel's proprietary, high-speed Direct-ConnectTM internal network, passing messages that contain instructions or data among the nodes. It provides high speed data pathways between all nodes of an iPSC/860 system and supports hundreds of simultaneous processor-to-processor communications with uniform

performance between all nodes. Each channel is a bi-directional pathway that delivers data at 5.6 MBytes/second.

The iPSC/860's UNIX workstation-based cross-development environment includes performance-optimizing compilers for FORTRAN and C languages. Also available are a parallel-oriented CASE development environment, interactive parallel debugger, performance analyzer, an iPSC/860 simulator, mathematical libraries, and matrix solvers. The iPSC/860 FORTRAN and C cross-compilers both use several optimization techniques to exploit the superscalar architecture of the i860 microprocessor. Among supported optimization are: vectorization, procedure in-lining, cache utilization strategies, software pipelining, i860 dual instruction mode, i860 dual operation mode, global optimization, local optimization, and loop optimization.

Intel's applications toolset for the iPSC/860 provides both broad functionality and high performance for numerically intensive applications. The toolset includes:

- The ProSolver family of linear equation solvers, a comprehensive set of parallel matrix solvers.
- The Basic Linear Algebra Sub-programs (BLAS) level 1, 2, and 3 standard routines for low-level numeric computation.
- The Numerical Algorithms Group library of single-processor algorithms, an extensive set of routines for solving numerical and statistical problems.

6. nCUBE Corporation

The nCUBE 2 parallel processing computer implements a MIMD parallel architecture approach with hypercube data network. The system is available from 8 to 8,192 computing elements and up to 512 billion bytes of memory, delivering speeds in excess of 27,000 MFLOPS and 60,000 MIPS. The nCUBE 2 processor is a full-custom VLSI design featuring 64-bit CPU, a 64-bit IEEE standard floating point unit, an error-correcting memory management interface with a 39-bit data path, a proprietary message routing unit, and 28 unidirectional direct memory access (DMA) channels. This chip, coupled with DRAM memory, constitutes a computing element. Users select local DRAM memory ranging from 1 to 64 Mbytes per computing element. Their systems include the nCUBE 2 Model 1, 1(), 2(), 40, and 80. The nCUBE 2 Model 1 is the entry level system in the nCUBE product family. It can house from 8 to 64 processors with up to 512 Mbytes of memory. Its potential performance is 470 MIPS and 210 MFLOPS. The nCUBE 2 Model 10 is a fully expandable entry level system. It is a general 64-bit parallel

supercomputer. The Model 10 can house from 64 to 1,024 processors and up to 16,384 Mbytes of memory. It potential performance is 7,500 MIPS and 3,375 MFLOPS.

The nCUBE 2 Model 20 is a midrange system which can house form 1,024 to 2,048 processors with up to 32 Gbytes of memory. Its potential performance rating is 15,000 MIPS and 6.75 GFLOPS.

The nCUBE 2 Model 40 is a high-end system within the nCUBE product family. The number of processors that can house in Model 40 range from 2,048 to 4,096, its memory can be up to 64 Gbytes, and its potential performance is 30,000 MIPS and 13.5 GFLOPS.

The nCUBE 2 Model 80 is the most powerful system in the nCUBE product family. It can house from 4,096 processors to 8,192 processors. Its memory capacity is 32 Gbytes, its potential performance is 60,000 MIPS and 27 GFLOPS.

Every nCUBE 2 is provided with compatible software development tools, including C and FORTRAN compilers, parallel debuggers, performance monitors, and subroutine libraries adapted to the nCUBE environment. nCUBE's applications toolset provides both broad functionality and high performance for numerically intensive applications. The toolset includes:

- FORGE™, a set of interactive programming tools designed to help in the development and maintenance of large bodies of FORTRAN code.

- Express™, a Parallel Operating Environment for the nCUBE 2 which provides users with a comprehensive set of tools for parallelizing C and FORTRAN programs. It offers users a range of low-level primitives and high-level utilities that facilitate the development of portable, high-performance, parallel software.

- MIMDizer™ offers nCUBE 2 users a state-of-the-art tool for the parallelization of FORTRAN programs. It works with the user to produce a highly parallelized representation of the target program. It begin by building an extensive knowledge base of the program being parallelized. Next it works interactively with the user to dissect and re-assemble a representation of the program in parallel form. MIMDizer makes it easier for users to design a completely new parallel program or redesign an existing serial program for parallel execution.

- PM, the nCUBE 2's Parallel Performance Monitor, monitors program status through an easy to use, windowed, graphical interface. It gives users the capability to analyze an algorithm's performance or to evaluate the effectiveness of various parallelization schemes.

7. Alliant Computer Systems Corporation

Alliant Computer Systems Corporation offers parallel-vector mini-supercomputers and visualization systems for computationally intense applications in science and engilleering. The FX/40 and FX/80 are the traditional products, while the FX/2800 is the most recent product announcement.

The Alliant FX/40 uses two processor types - advanced computational elements (ACEs) and interactive processors (IPs). The FX/40 ACEs are general purpose 64-bit vector processors based on 20,000 gate array CMOS technology. With up to four vector processors, the FX/40 delivers 94.4 MFLOPS of peak performance. The IPs are general-purpose processors that independently execute user jobs and operating system tasks and enable the ACEs to concentrate on computation. Up to six IPs can be configured on the FX/40. The FX/40 provides up to 128 MBytes of physical memory and 2 GBytes of virtual memory and 1.1 GBytes of disk.

The FX/80 can be configured up to 8 ACEs and 12 IPs to provide 188.8 MFLOPS of peak performance. The FX/80 provides up to 256 MBytes of physical memory and 2 GBytes of virtual memory.

The FX/2800 family of parallel supercomputers can be configured with up to 28 64-bit RISC processors, arranged as 14 Super Computational Elements (SCEs) for parallel processing of large compute intensive tasks, and 14 Super Interactive Processors (SIPS) for multiprocessing serial compute jobs, executing system tasks, 1/0 and 3-D graphics processing. A single-Processor Module (PM) contains four Intel i860 processors, arranged as two SCEs and two SIPS. Up to seven PMs are supported in the FX/2800. The FX/2800 processors are connected to the system's memory subsystem through a crossbar switch. Its shared memory capacity ranges from 64 MBytes to 1 GBytes of physical memory and 4 GBytes of virtual memory. The FX/2800 high end system has peak performance of 1596 Whetstone MIPS and 296 Linpack 1000 MFLOPS.

Alliant's Concentrix operating system is a symmetric multiprocessing implementation of UNIX with extensions for parallel supercomputing. It is a general purpose, multi-user operating system for scientific and engineering applications and supports FX/Fortran, FX/C, C, FX/Ada, Pascal, and the Alliant assembler.

The FX/Fortran compiler automatically detects the potential for vector and parallel processing in a standard Fortran code. It also checks for data dependencies between loops and for constructs such as conditional branches, loop exits, and subroutine calls. The compiler can generate code for one of five different mode of execution: scalar, vector, scalar concurrent, vector concurrent and concurrent outer/vector inner. The FX/Fortran

compiler optimizes standard Fortran code in the following areas: loop optimization, array extensions for optimization, and automatic global optimization.

The FX/C compiler is an optimizing compiler that uses both global and local optimization methods via several means: loop optimization, automatic global optimization, automatic inline expansion, automatic parallel and vector processing.

8. BBN Advanced Computers Inc.

The BBN TC2000 System is a shared-memory MIMD machine. The TC2000 System consists of a number of independent and equal processor nodes interconnected by BBN Advanced Computers' proprietary, third generation Butterfly switch technology. Using packet-switching techniques, the Butterfly switch implements fast interprocessor communications and provides transparent shared access to all memory in the system. The Motorola 88100 RISC processor nodes operate independently, executing instructions from their local memory and/or referencing memory on other processor nodes.

It is currently available in configurations from four to 512 processing elements and memory sizes from 28 to 8,192 MBytes. In its fully configured 63-processor, a TC2000 System achieves 800 Whetstone MIPS and 1,260 MFLOPS with up to 1 GBytes of shared physical memory.

The TC2000 system provides an integrated programming and development environment for time-critical applications, including all the software engineering features in the UNIX operating systems, along with a full set of programming languages and compilers, including FORTRAN, Ada, C, and assembly language. The TC2000 FORTRAN and C compilers offer an extensive set of optimization, including: register optimization, subroutine and function-call optimization, loop optimization, code simplification, folding of constants and instruction scheduling.

The TC2000 development tools address all the major programming tasks—coding, documentation, compilation, debugging and performance tuning. The toolset includes:

- Xtra tool set, a platform for debugging, analysis, and performance tuning of multiprocessing programs. It provides users with a graphic view of the dynamic interactions among multiple processes and processors. Xtra consists of the TotalView debugger and the Gist performance analyzer.

- The TotalView debugger provides a means to observe, probe, and understand the complex interactions inherent in a multiprocessor system.

- The Gist performance analyzer is a graphic tool for examining the dynamic behavior of multiple-process programs. It provides a graphical picture of the interaction of processes executing in parallel on a microsecond-by-microsecond basis.

9. Encore Computer Corporation

Encore Computer Corporation currently supplies three families of systems: The CONCEPT/32 line of real-time computers. The Multimax family is a UNIX-based parallel processing/multiprocessing systems. It is a MIMD system utilizing a shared memory architecture in which all memory elements can be accessed by any processor element. The Encore 90 family of computers combines UNIX-standard parallel processing from

Multimax and Real-Time Computing Capability of CONCEPT/32

The Encore 91 Series is the low-end member of the Encore 90 family. It can be configured with two or four Motorola 88100 processors. The 88100 is a RISC (Reduced Instruction Set Computing) microprocessor with 51 instructions. The Encore 91 include 16 MBytes of RAM and can be expanded up to 272 MBytes, it also utilizes Motorola 88200 Cache Memory Management Units (CMMUs) to enhance the performance of each 88100 processor. The four processors Series 91 achieves up to 80 Whetstone MIPS of processing power.

The high end of the Encore 90 family is the 93 Series. It can be configured with up to 32 CPU's, its shared physical memory size range from 64 to 640 MBytes. The performance for a fully loaded system is 640 Whetstone MIPS and 128 MFLOPS.

The computation and input/output components of Encore 90 systems are linked by the Nanobus: a fast, bi-polar, pended system design which provides 100 MBytes/second of data throughput.

Encore high-level languages have advanced, highly optimizing code generators. The FORTRAN has two versions, sequential and parallel. The sequential version creates programs that conventionally execute on a single processor. The parallel version creates programs that allow parts of the application to execute on more than one processor at a time. This parallelization can be either user-directed, automatic, or a combination of both.

10. FPS Computing

FPS Computing, formerly Floating Point Systems, offers the System 500 SPARC supercomputer family. The System 500 is a shared memory computer system and is the industry's first supercomputer to incorporate the Scalable Processor ARChitecture (SPARC) standard created by Sun Microsystems, Inc. It is a 64-bit, stand-alone, UNIX-based supercomputer. The System 500 can be configured up to 8 SPARC processors and up to 1 gigabyte of physical memory. Compute-intensive applications can be performed within one system by combining computational techniques in scalar, vector, parallel, and matrix computing. FPS Computings's Integrated Heterogeneous Processing—VectorPlus

CoProcessors, Parallel SPARC Processors and Application-Specific Matrix Coprocessors rcombined to provide high performance computing. SPARC scalar processing is augmented by VectorPlus facilities. Automatic parallelization applies multiple SPARC processors or VectorPlus coprocessors to a single program. The Matrix Coprocessors extends heterogeneous processing to matrix algorithms. FPS Computing has developed a high-locality technology for executing matrix applications. Intel's i860 is used as a single-chip Processing Element (PE) to perform high-locality computing. The Matrix Coprocessors can be configured starting from 4 PEs up to 84 PEs. Peak performance ranges from 320 MFLOPS to 6.7 GFLOPS.

FPS Computing's conFORm, Concurrent FORTRAN Compilation System, offers automatic and user-directed parallelization. Simple compiler options invoke automatic parallelization. Experienced users can apply assertions and directives that further enhance parallelization. conFORm uses data dependence and code analysis techniques to identify opportunities for concurrency in programs. Among the optimization performed are: loop reordering, induction variable recognition, global forward substitution, variable lifetime analysis, loop peeling, subprogram inlining, dead code elimination, and array expansion.

FPS Computing also provides FPSMathTM, a math library for technical computing. It includes mathematical functions which address geophysical, image processing, simulation, signal processing applications, as well as vector and matrix algorithms that supported applications in structural analysis, computational fluid dynamics, electronic circuit design, and computational chemistry. The library supplements Basic Linear Algebra Subroutines (BLAS) levels 1, 2, and 3.

11. Thinking Machines Corporation

The CM-2 Connection Machine is a massively parallel computer. The system itself can have anywhere from 212 = 4,096 to 216 = 65,536 processors. Each processor, in turn, has from 216 = 65,536 to 218 = 262,144 bits of local memory, so the machine has an overall memory capacity of up to 2 gigabytes. The processors are arranged in hardware with 16 to a chip. In addition, every pair of chips (32 processors) is supported by floating-point accelerator hardware.

The module consisting of two chips (32 processors) and their associated floating-point hardware is the fundamental hardware building block of the Connection Machine. Sixteen of these blocks (512) processors are laid out on a board. Sixteen of these boards (8192 processors) comprise one octant of a full Connection Machine.

Each CM-2 octant has its own I/O bus and microcontroller. This makes it possible to spaceshare the machine in blocks of 8k processors; multiple users can access blocks of multiple of 8k processors, independently running programs and using I/O devices.

Two types of parallel I/O devices are available for the CM-2 system. The first is a graphical display device, or frame-buffer that acts as a "window" into Connection Machine memory, allowing the user to watch the real-time evolution of data stored in the processor array. The second is a disk farm, called the DataVault that consists of forty-two 51/4 inch Winchester disk drives that can read (write) data directly from (to) the CM-2's processors without involving the front end in any way.

In most scientific applications programs, the amount of data present is frequently measured in tens or hundreds of millions, in any event, it may be much greater than 65536. For this reason, the Connection Machine system includes very low-level system software that allows the user to program the machine as though it had more processors. It does this by having each physical processor simulate some larger number of virtual processors (VP's). The number of virtual processors per physical processor is called the virtual processor ratio (VP ratio).

Because this abstraction is supported by low-level system software, it is transparent to the high-level applications programmer who always works with VP's. All high-level software for the CM-2 is written for virtual processors; interprocessor communications really means inter-virtual-processor communications, and on-processor operations are really on-virtualprocessor operations. Thus, the same program will run unchanged on differently sized Connection Machine; one simply specifies the desired size of the (virtual) processor array, and the VP ratio is set accordingly.

Interprocessor communication is implemented by a special-purpose high-speed network. Processors that hold interrelated data elements store pointers to one another. When data is needed, it is passed over the network to the appropriate processors. The network supports general patterns of communication, but additional special hardware supports certain commonly used regular pattern of communication. Nearest-neighbor communication in a multidimensional rectangular grid is particularly efficient.

There are three forms of communication within the parallel processing unit: routing, NEWS, and scanning.

The most general mechanism is the *router,* which allows any processor to communicate with any other processor. In effect, the router allows the local memories of the data processors to be treated as a single large shared memory.

A faster but more structured communication mechanism is called the *NEWS* grid. The CM-2 supports programmable grids of up to 31 dimensions. The dimensions may be

of any length, as long as the products of all dimensions equals the total number of processors. The NEWS grid allows processors to pass data according to a rectangular pattern. Special hardware makes this structured NEWS communication significantly faster than general routing.

Scanning is a more powerful operation on NEWS girds that combines communication and computation. Scanning is supported by special hardware related to the NEWS hardware.

12. Research Approach

The exploitation of parallel processing will have direct bearing on the very design of the models and algorithm and on their underlying conceptual underpinnings. To run efficiently in a parallel environment, a sequential application must be partitioned (or decomposed). This involves dividing the data or code among the available or allocated processors. It may also involve changing DO-loop limits, array dimensions, and subroutine parameters so that each processor can operate on a subset of data.

There are two basic approaches to partitioning:

Control-Parallelism

This approach break up a standard program into more or less independent subsets of instructions and to assign on such subset to each processor. This approach is called multitasking or control-parallelism. It is used by vector supercomputers as well as by the numerous MIMD computers.

Though multitasking does solve the problem of getting all the processors to work on a single problem, it has several disadvantages. For instance, it cannot scale well to massively parallel regime with a very large number of processors. The breakdown of the instruction set into independent subunits is usually possible only to a certain level of granularity, beyond which no further such division is possible. For most engineering applications, the number of such independent subunits is typically measured in tens. Furthermore, synchronization and load balancing between the workings of different subunit become a real problem.

Data-Parallelism

The hardware and software paradigm that scale well into the massively parallel regime is to base the parallelization on a program's data rather than its instruction stream. Most programs manipulate tens of millions of pieces of data while very few programs have tens of millions of lines of code -- much less, tens of millions of independent subunits of

instructions. Data is much more parallelizable than instructions, and this will be even more true for the next century.

Programming Models

SIMD and MIMD architectures often have similar algorithmic issues and problem domains but currently offer very different programming environments. Brief description of differences in programming models for the three classes will be discussed:

Class I MIMD Distributed Processing of Distributed Data

Class II MIMD Distributed Processing of Shared Data

Class III SIMD Distributed Processing of Distributed Data

The shared memory of the class II systems allows programming models where the user and/or compiler does not need to worry about the location of data. For example, one can take conventional Fortran Code and either automatically, or with user directives, obtain concurrency from parallel execution of separate iterations of a DO loop. In the class I machines, the communication among the processors has to be made explicit by the user. The class III programming paradigm is the association of processors with data. An application might associate one processor per node in a finite difference calculation, one processor per walker in a Monte Carlo calculation, one processor per molecule in a molecular dynamics simulation, one processor per star for a galactic dynamics simulation. Then a set of communications is broadcasted to all processors which carry out the calculations on all the data at once.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

This chapter presents a summary of the conclusions reached as a result of the review and synthesis conducted under Task A, and discusses the study team's recommendations regarding the direction of the work to be performed under Tasks B and C.

1. Conclusions

There are two basic traffic modelling capabilities required in the ADIS/ATMS context. The first is a normative prescriptive capability by which a central controller can assign drivers on a real-time basis to the various paths of the network. The central controller could operate under different information availability scenarios, and will seek to optimize overall system performance subject to individual routing constraints. The second capability is a descriptive simulation-assignment capability, that can represent the traffic patterns that result from the decisions made by users in response to supplied real-time information. This capability is an integral component for computing the system performance measures in connection with the first problem (system optimal assignment), as well as to predict projected traffic conditions in a real-time environment.

The principal conclusion of our review under Task A is a confirmation that no models presently exist to satisfy either of the above two capabilities. However, there are elements in three principal areas that may be of direct relevance to the development of these capabilities.

Traffic simulation models can successfully provide a reasonable representation of the dynamics of traffic flow in a network, provided that the vehicles' desired movement is known and pre-specified (typically in the form of turning movements at nodes). Neither the existing simulation models, nor the so-called simulation-assignment models possess the capability of letting users choose their path from any location to the desired destination, in response to real-time traffic condition information or route-guidance instructions. Similarly, none of the existing models possess the kind of data structures to support path processing operations on the network (i.e., identifying specific paths between a given origin and destination, updating the trip times on paths given their link composition, searching for paths that meet certain criteria, and solving for shortest or k-shortest paths). Similarly, none of the existing models seem to be sufficiently developed with regard to the objects or data structures necessary to support the simulation and analysis of different

control actions (particularly relevant in ATMS context). These are major limitations of traffic simulation models with regard to the predictive capability needed in the ADIS/ATMS context.

On the other hand, procedures for network assignment, primarily static assignment, have gone a long way in terms of efficient data structures to support network path processing operations. However, they may not be entirely adequate for such operations for problems with time-dependent link travel times. Existing formulations for the dynamic assignment problem are still struggling with very basic issues of how to correctly and realistically represent the underlying traffic phenomena. No operationable formulations exist to date, and the principal approach used in connection with existing formulations, namely the use of link exit functions, in addition to link performance functions, is not particularly promising. It is evident that the proper representation of traffic dynamics is missing from virtually all existing formal dynamic assignment formulations. Ideally, traffic performance, even for the prescriptive, system-optimal problem, should be based on a network-level traffic simulation. However, existing traffic assignment procedures (mostly for the static problem) do not have the necessary data structures to support traffic flow simulation.

Another limitation of both static assignment models and of quasi-dynamic simulation-assignment heuristics (e.g., CONTRAM and INTEGRATION) is that the only path processing capability provided is that of shortest path computation. This is not enough in the ADIS/ATMS context. Shortest path assignment and single-path tracking are not sufficient for ADIS/ATMS applications. In particular, the user behavior/decisions component in the general case requires a richer array of information on alternative path choices. It is therefore desirable to be able to solve for and dynamically update traffic conditions on several paths simultaneously between a given O-D pair.

Since no satisfactory formulations are available for the dynamic system optimal problem, no general solution algorithms are available, a fortiori. It will therefore be necessary to undertake considerable fundamental work on this problem prior to developing a prototype code. However, such prototype code will also be useful as a tool to learn more about the problem and its properties and to consequently develop effective solution methods.

Because computational performance is of critical importance in the real-time operation of both modelling capabilities needed for ADIS/ATMS implementation, advanced computing environments and architectures must be considered for the methodological development underway. Our review of available hardware and discussions with experts in this field have revealed a wide range of configurations and possibilities, offering different

combinations and trade-offs in terms of speed, price and development effort. The opportunities made available by new advanced parallel architectures also pose challenges to the methodological development effort, as the hardware configuration influences the realm of practical algorithmic procedures and the very logic and structure of such procedures. To the extent that the models developed under the study are not likely to be actually deployed in the field for another five years, one needs to design these models in a manner that will not be obsolete or grossly sub-optimal relative to the hardware capabilities available at the time of implementation. For this reason, "generic" parallelism will be sought to the extent possible in the methodological development. However, it appears safe to say that two types of parallel environments available today are likely to cover quite well the range of opportunities that will become available in the foreseeable future:

1. Parallel asynchronous MIMD machines, with vector processors. The CRAY Y-MP supercomputer is a primary example of such machines, which additionally offers supercomputing capabilities (i.e. very fast individual processors).

2. Massively parallel SIMD machines, typified by the Connection Machine, which has shown considerable promise on certain types of problems, including network problems. However, no applications close enough to traffic flow modelling and simulation have been reported on such machines, thereby necessitating developmental work in connection with this project.

The next section discusses specific recommendations regarding the direction of future work on Tasks B and C in light of the above findings.

2. Recommendations

The principal recommendations pertain to the direction of future activities in connection with the system optimal dynamic assignment/routing problem (Task B) and the descriptive simulation-assignment capability (Task C). These are discussed in turn hereafter.

2.1 System Optimal Dynamic Assignment

As indicated in the previous section, one of the primary tasks is to develop specific formulations of the dynamic assignment problem faced by the controller. These will draw on the formulations reviewed in Chapter 2, along the lines indicated in that chapter. Once the formulations are agreed upon, their properties can be investigated, and solution algorithms and heuristics can be developed and tested.

The primary features of the formulations of interest are as follows:

a. It is essential to properly model the dynamics of traffic flow. Link exit functions and link performance functions do not provide an acceptable representation of the phenomena at hand. Traffic simulation will be used as the primary means to evaluate the objective function and other impacts resulting from a particular assignment/routing strategy.

b. The formulations will consider different possible scenarios of information availability to the central controller. Under full information, the controller would know all O-D trip desires over the whole planning horizon. This formulation will be pursued as a sort of "ideal" case. Other formulations will explicitly recognize partial information, consider a rolling horizon approach, and introduce stochastic effects. No predictive models of O-D trip desires will be pursued, as they fall outside the scope of this study.

c. As discussed in the study proposal, different solution approaches will be investigated. Essentially, one should recognize the complexity of the resulting formulation, especially since the evaluation of the objective function ultimately requires the execution of a simulation run. Therefore exact solution algorithms with nice properties are not likely (if only because the underlying problem is not likely to be well behaved, as discussed in Chapter 2). Therefore, heuristic approaches must be considered, including relaxation and decomposition techniques. In addition, simple heuristic control rules will be investigated and tested.

d. The algorithms and prototype code development should proceed in tandem with the development of the descriptive simulation-assignment modelling capability. To the extent that the latter is an essential ingredient for the former, it appears plausible to undertake both simultaneously.

e. Because no exact algorithm is likely to be established for this problem, it will be necessary to test the performance of alternative solution procedures. Even though the delivery of a working code is not required under this project, we envision the development of a prototype implementation that would allow the conduct of extensive performance tests of any meaningful solution procedure.

2.2 Descriptive Simulation-Assignment Models

As noted previously, this modelling capability is critical to the success of the preceding one, since it determines the traffic patterns resulting from particular dynamic assignment/routing strategies, and thereby evaluates the system level objective function and figures of merit of interest to the controller.

In light of the deficiencies of existing traffic simulation models, as well as those of network assignment codes, four possible development strategies can be identified:

1. Interface existing traffic simulation models and network assignment models.
2. Add network path processing capabilities to an existing network assignment package.
3. Add traffic flow simulation capability to an existing network assignment package.
4. Configure the simulation-assignment model structure to best fit the functional requirements of the ADIS/ATMS context.

We believe that the last strategy is by far the most desirable and promising for this particular problem. This of course does not mean that one does not "borrow" different elements from a variety existing procedures, but that one uses a new framework and structure that recognize the very special needs of this problem. As discussed throughout, the latter poses challenges that go far beyond any of the capabilities presently available in existing packages. Strategy 1 suffers from the limitations of each of its two main components individually, in addition to the fact that interfacing separately conceived models often leads to inefficiencies in computational performance and increased likelihood of conceptual flaws. Strategy 2 is particularly tedious and not likely to succeed, and would be more difficult and more prone to error than starting with a fresh slate. Strategy 3 suffers from some of the same limitations as strategy 2, with the added concern that the static assignment packages that would form the starting point for such effort do not themselves meet the requirements for network path processing in the ADIS/ATMS context. This leaves strategy 4, which essentially seeks to build the traffic modelling methodologies for ADIS/ATMS on a sound basis, rather than be continually limited by an inadequate starting point.

Another recommendation regarding the simulation-assignment capability pertains to the appropriate level of detail in the simulation of traffic movement. Such simulation is not likely to be microscopic. Such detail may not be necessary, except for assessing detailed aspects of the system configuration. On the other hand, the ability to track individual vehicles or bunches of vehicles is important to the behavioral component of the simulation-assignment methodology, as well as to the design of individual routing instructions. However, traffic phenomena can most likely be captured adequately using macroscopic traffic flow relations, all the while tracking the path choices of individual vehicles

Taking the functional requirements of the simulation assignment model into account, and following up on the above recommendations and conclusions, it appears that the assignment-simulation model framework prototyped by Mahmassani and Jayakrishnan (1990) at the University of Texas at Austin is likely to provide a useful starting point for the

model development effort. This model represents vehicular movement, and explicitly represents the path selection decisions of individual motorists, particularly in response to real-time in-vehicle information, both en-route and at the trip's origin. The dynamics of traffic simulation are modelled using a traffic simulator, thereby avoiding the drawbacks of analytical link performance and link exit functions. The simulator is adapted from the macroparticle traffic simulation (MPSM) approach (Chang, Mahmassani and Herman, 1985). It follows a fixed time-step logic, and uses macroscopic traffic theoretic relations to find the prevailing speeds in a given section, though vehicles are subsequently moved individually (or in bunches or macroparticles).

One of the primary advantages of the above framework for the ADIS/ATMS context is that it is structured around a powerful network path processor specifically intended for this context. In addition, its modularity facilitates adaptation to different formulations and assumptions. In particular, it possesses an explicit user decisions component, which can represent in a more or less elaborate manner the behavioral processes governing the users' responses to supplied real-time information. However, the traffic simulation component per se is not as elaborate, especially in terms of user interfaces, as some of the well known dedicated traffic simulation packages (such as the TEXAS, NETSIM/TRAF and FREQ models). However, the latter models could eventually supply the logic and possibly partial code for components that could be relatively easily integrated in the overall framework.

2.3 Computational Environments for Prototype Development

In light of the discussion in the previous section, the prototype code development undertaken under the present project will take place in two main computing environments:

1. The CRAY Y-MP of the Center for High Performance Computing of the University of Texas System. As noted it offers insights into several major advanced computing capabilities, namely: (1) parallel ascynchronous processors (MIMD machine), (2) vectorization and (3) supercomputing.

2. The massively parallel architecture of the Connection Machine, a SIMD machine, available at the University of Maryland.

The former will provide the primary test bed for prototype development for both simulation-assignment and system optimal dynamic assignment capabilities. However, it is worthwhile also to investigate first hand the second environment because major improvements in performance have been reported for certain classes of problems, including network problems. Because no prior experience appears to have been reported on problems with similar characteristics as traffic flow simulation on the Connection Machine, it is recommended that some prototype development be undertaken in this area so as to

develop some basis for estimating computational performance and real-time execution feasibility of the procedures developed for the ADIS/ATMS context.

REFERENCES

1. Beckmann, M.J., McGuire, C.B., and Winston, C.B. (1956). *Studies in the Economics of Transportation*, Yale University Press, Conn.

2. Ben-Akiva, M. (1985). "Dynamic Network Equilibrium Research" *Transportation Research*, Vol.19A, no. 5/6, pp 429-431.

3. Boyce, D.E. (1989). "Route Guidance Systems for Managing Urban Transportation Networks: Review and Prospects", Presented at Tenth Italian Regional Science Conference, Rome, Italy.

4. Carey, M. (1986). "Optimal Time Varying Flows On Congested Networks", *Operations Research*, Vol.35, no.1, pp 58-69.

5. Carey, M. (1986). "A Constraint Qualification for a Dynamic Traffic Assignment Model", *Transportation Science*,Vol.20, pp 55-58.

6. Carey, M. (1989). "Nonconvexity of the Dynamic Assignment Problem",TRB, 1990.

7. Dafermos, S. (1980). "Traffic Equilibrium and Variational Inequalities", *Transportation Science* 14, pp 42-54.

8. Dafermos, S. (1982). "The General Multimodal Network Equilibrium Problem with Elastic Demand", *Networks* 12, pp 57-72.

9. D'Ans, G.C. and Gazis, D.C. (1976) "Optimal Control of Oversaturated Store and Forward Transportation Networks", *Transportation Science* 10, pp 1-19.

10. Fisk, C.S. and Boyce, D.E. (1983). "Alternative Variational Inequality Formulations of the Network Equilibrium-Travel Choice Problem", *Transportation Science* 17, pp 454-463.

11. Friesz, T.L., Luque, J., Tobin, R.L. and Wie, B.W. (1989). "Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem", *Operations Research*, Vol.37, no.6, pp 893-901.

12. Gartner, N.H. et al. (1983). "Simulation Study of OPAC : A Demand-Responsive Strategy for Traffic Signal Control", in *Transportation and Traffic Theory*, Eds-Gartner, N.H. and Wilson, N.H.M., Elsevier Science Publishing Co.

13. Hendrickson, C., and Kocur, G. (1981). "Schedule Delay and Departure Time Decisions in a Deterministic Model", *Transportation Science* 15, pp 62-77.

14. Hendrickson, C., and Plank, E. (1984). "The Flexibility of Departure Times for Work Trips", *Transportation Research*, 18A, (1), p 25-36.

15. Ho, J.K. (1980). "A Successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem", *Transportation Science* 14, pp 295-305.

16. Hurdle, V.P. (1981). "Equilibrium Flows on Urban Freeways", *Transportation Science* 15, pp 255-293.

17. Leonard, P.R., Gower, P. and Taylor, N.B. (1989). "CONTRAM: Structure of the Model", *TRRL Research Report* RR 178, United Kingdom.

18. Lindley, J.A. (1988). Personal Notes, FHWA, Washington, D.C.

19. Lisco, T.E. (1983). "Procedure for Predicting Queues and Delays on Expressways in Urban Core Areas", *Transportation Research Record* 944, pp 148-154.

20. Mahmassani, H.S., Herman, R., Walton, C.M., Jones, E.G., Baaj, M.H., Jayakrishnan, R. (1989). "Travel Time Characteristics and Opportunities for Real-Time Information Systems in an Urban Commuting Corridor", Research Report GM-1989-F, Center for Transportation Research, The University of Texas at Austin.

21. Mahmassani, H.S. and Mouskos, K.C. (1988). "Some Numerical Results on the Diagonalization Algorithm for Network Assignment with Asymmetric Interactions Between Cars and Trucks", *Transportation Research B*, 22B, pp 275-290.

22. Merchant, D.K., and Nemhauser, G.L. (1978). "A Model and an Algorithm for the Dynamic Traffic Assignment Problems", *Transportation Science*, Vol.12, no.3, August 1978, pp 183-199.

23. Merchant, D.K., and Nemhauser, G.L. (1978). "Optimality Conditions for a Dynamic Traffic Assignment Model", *Transportation Science*, Vol.12, no.3, August 1978, pp 200-207.

24. Midler, J.L. (1969). "A Stochastic Multiperiod Multimode Transportation Model", *Transportation Science* 3, pp 1-7.

25. Nagurney, A. and Kim, D-S (1989). "Parallel and Serial Variational Inequality Decomposition Algorithms for Multicommodity Market Equilibrium Problems", *The International Journal of Supercomputer Applications*, Vol. 3, pp 34-58.

26. Nagurney, A. (1986). "Computational Comparisions of Algorithms for General Asymmetric Traffic Equilibrium Problems with Fixed and Elastic Demands", *Transportation Research B*, 20B, pp 78-84.

27. Papageorgiou, M., Messmer, A. and Senninger, H. (1990) "Dynamic Network Traffic Assignment Via Feedback Regulation".

28. Powell, W.B. (1987). "A Comparative Review of Alternative Algorithms For The Dynamic Vehicle Allocation Problem", Prepared for Vehicle Routing : Methods and Studies.

29. Ran, B., and Shimazaki, T. (1989a). "A General Model and Algorithm for the Dynamic Traffic Assignment Problems", Proceedings of the Fifth World Conference on Transport Research, Yokohoma, Japan.

30. Ran, B. and Shimazaki, T. (1989b). "Dynamic User Equilibrium Traffic Assignment for Congested Transportation Networks", Presented at the Fifth World Conference on Transport Research, Yokohoma, Japan.

31. Ran, B., Boyce, D.E. and LeBlanc, L.J. (1990). "An Instantaneous Dynamic User-Optimal Traffic Assignment Model and Solution Algorithm", Submitted for Publication consideration in *Operations Research.*

32. Robillard, P. (1974). "Multipath Traffic Assignment with Dynamic Input Flows", *Transportation Research* 8, pp 567-573.

33. Sheffi, Y. (1985). Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming Methods, (Prentice-Hall, NJ).

34. Smith, M.J. (1984). "The Stability of a Dynamic Model of Traffic Assignment—An Application of a Method of Lyapunov" *Transportation Science*, Vol.18, no.3, August 1984, pp 245-252.

35. Tittemore, L.H., Birdsall, M.R., Hill, D.M., and Hammond, R.H. (1972). "An Analysis of Urban Area Travel by Time of Day", US Dept. of Transportation, Washington, D.C.

36. Wagner, H.M. (1977). *Principles of Operations Research*, 2nd ed., Prentice Hall, NJ.

37. Wie, B.W. (1990). "Dynamic Analysis of User Optimized Network Flows With Elastic Travel Demand", prepared for TRB Meeting, Washington, D.C., 1991.

38. Yagar, S. (1976). "Emulation of Dynamic Equilibrium in Traffic Networks", in Florian, M. (ed.) *Traffic Equilibrium Methods*, Springer-Verlag, pp 240-264.