

**A DYNAMIC TRAFFIC SIMULATION/ASSIGNMENT MODEL IN THE CONTEXT  
OF ADVANCED DRIVER INFORMATION SYSTEMS**

by

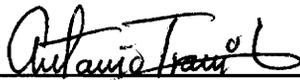
Kaan Ozbay

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of  
Master of Science  
in  
Civil Engineering

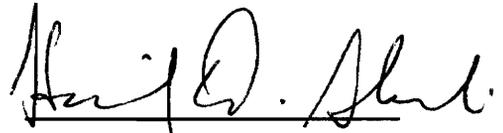
APPROVED:



A.G. Hobeika, Ph.D, Chairman



A.A. Trani, Ph.D



H.D. Sherali, Ph.D.

August 26, 1991

Blacksburg, Virginia

C.2

LD  
5655  
V855  
1991  
0922  
C.2

# A DYNAMIC TRAFFIC SIMULATION/ASSIGNMENT MODEL IN THE CONTEXT OF ADVANCED DRIVER INFORMATION SYSTEMS

by

Kaan Ozbay

Dr. A. G. Hobeika

(Chairman)

(ABSTRACT)

Growing congestion problems of many metropolitan areas which cause excessive traffic delay, instability of travel time generated the need for the development of an Intelligent Vehicle Highway System (IVHS) program that is capable of making significant improvements in mobility, highway safety, and productivity on highways and urban streets. The success of such real time control system highly depends on the new methods that address dynamic traffic assignment. Thus there is an urgent need for an effective dynamic assignment model.

The main objective of this research is to emphasize the importance of dynamic traffic assignment for Advanced Driver Information Systems (ADIS) which is one of the major components of IVHS and to present a practical traffic assignment model that is capable of running in real time and which can accurately predict link travel times, queue build up, and network performance. DYNTRAS (DYNamic Traffic Simulation Assignment), a simulation /assignment model is developed. The model uses an incremental loop that assigns a

portion of the Origin-Destination matrix, and simulates the movement of the vehicles. Then, it updates travel times and assigns an additional portion of the O-D matrix. In contrast to traditional traffic assignment models like "capacity restraint" and "incremental assignment" techniques that do not consider time dimension, DYNTRAS incorporates time as a third dimension by keeping track of the vehicle movements in time. As a result, it is capable of predicting time-dynamic impacts of congestion and effects of diverted traffic on traffic flow more realistically.

The model is applied to a test network. Several experimental factors are varied to test the sensitivity of the model. The results obtained are presented and general conclusions are derived. The differences between dynamic and static traffic assignment results are also discussed by considering results obtained from both methods.

The model needs to be calibrated using real traffic data. According to the results obtained, it needs to be validated. In addition, its long computation time should be reduced to be able to use it for real time applications.

# Acknowledgements

I would like to acknowledge Dr. A.G. Hobeika, Chairman of my committee, for his effort devoted to the development and completion of this research project, and for his financial support throughout my involvement in this research. In addition, I would like to thank Dr. H.D. Sherali and Dr. A.A. Trani for being my committee members.

I would like to thank Dr. R. Sivanandan for his helpful advice and comments in refining this work.

This thesis could not be completed without tremendous help of my lovely wife. She supported me morally and intellectually, in every step of this work. I would like to express my gratitude to my mother and grand-mother who gave me the opportunity to continue my studies in U.S.A by supporting me morally and financially. Finally, I would like to dedicate this thesis to my grand-father and grand-mother.

# Table of Contents

<b>1.0</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Urban/Suburban Congestion Problem in the United States	1
1.2	The Role of IVHS in Addressing Urban/Suburban Congestion Problem	2
1.3	Role of Dynamic Traffic Assignment in Developing Diversion Plans	5
1.4	Problem Statement and Research Objectives	7
1.5	Summary of the Other Chapters	10
<b>2.0</b>	<b>LITERATURE REVIEW</b>	<b>11</b>
2.1	Traffic Assignment and Advanced Driver Information Systems (ADIS)	11
2.2	Static Traffic Assignment Models	13
2.2.1	Heuristic Static Assignment Techniques	15
2.2.2	Mathematical Programming Technique	17
2.3	Dynamic Traffic Assignment Models	20
2.3.1	Linear Programming Approach	21
2.3.2	Optimal Control Theory Approach	25
2.3.3	Heuristic Approaches	26
2.3.4	Dynamic Models of Peak Period Congestion	31
2.3.5	Simulation/Assignment Approaches	33
2.3.5.1	Saturn	33
2.3.5.2	Contram	37
2.4	Summary of Literature Review	43
<b>3.0</b>	<b>THEORETICAL ASPECTS OF THE MODEL</b>	<b>45</b>
3.1	Introduction	45
3.2	A Dynamic Traffic Simulation/Assignment Model	46
3.2.1	Traffic Assignment in Three Dimension	46
3.2.2	The Proposed Simulation/Assignment Model	49
3.2.3	Simulation/Assignment Process	50
3.2.4	Macroparticle Simulation Model (MPSM)	56
3.2.4.1	Vehicle Generation Component	57
3.2.4.2	Traffic Simulation Component	59
3.2.4.3	Vehicle Moving Process	63
3.2.5	Travel Time Estimation	66
3.2.6	Diversion and Driver Behavior	67
3.3	Model Parameters	70
3.3.1	Simulation Time Step	70
3.3.2	Selection Length	71
3.3.3	Microparticle Size	72

<b>4.0</b>	<b>TECHNICAL ASPECTS OF THE MODEL .....</b>	<b>74</b>
4.1	Description of the Computer Code.....	74
4.2	Model Structure .....	75
4.2.1	Model Input and Output .....	75
4.2.2	Data Representation .....	76
4.3	General File Operations and Network Data Handling Routines .....	77
4.3.1	General File Operations Routines .....	77
4.3.2	Data Handling Routines .....	78
4.3.2.1	Data Management Subroutine (NETDATMAN)	78
4.3.2.2	Link Determination Subroutine (NODLINDET) .....	79
4.4	Shortest Path Subroutine (DIJSHOPATH2) .....	80
4.5	User Equilibrium Module .....	82
4.5.1	Convex Combination Method Subroutine (CONCOMNET) .....	82
4.5.2	Subroutine Bisect .....	83
4.5.3	Subroutine Update .....	85
4.5.4	Subroutine Allornot .....	86
4.6	Dynamic Traffic Assignment Module .....	86
4.6.1	Traffic Simulation Subroutine (TRAFSIMULA)	90
4.6.2	Subroutine Merge .....	93
4.6.3	Subroutine Generate .....	94
4.6.4	Subroutine Arrive .....	95
4.6.5	Subroutine Trafmove .....	98
4.6.6	Subroutine Divert .....	99
4.6.7	Subroutine Delete .....	101
4.6.8	Subroutine Fstatistics .....	102
4.6.9	Subroutine Output .....	102
4.6.9	Subroutine Radio .....	102
<b>5.0</b>	<b>MODEL RESULTS and ANALYSIS .....</b>	<b>103</b>
5.1	Introduction .....	103
5.2	Analysis Description .....	103
5.3	Experimental Factors .....	106
5.4	Analysis of Results .....	108
5.4.1	Scenario 1 .....	109
5.4.2	Scenario 2 .....	117
5.4.3	Scenario 3 .....	120
5.4.4	Scenario 4 .....	122
5.5	Conclusion .....	122
<b>6.0</b>	<b>CONCLUSIONS and RECOMMENDATIONS .....</b>	<b>126</b>
6.1	Conclusions .....	126
6.2	Recommendations .....	129
<b>REFERENCES</b>	<b>.....</b>	<b>131</b>

# List of Illustrations

Figure 1.1	The Two Approaches of Developing Diversion Strategies .....	5
Figure 1.2	User Optimal Diversion Approach .....	9
Figure 2.1	The Four Basic Cyclic Flow Patterns ....	35
Figure 3.1	Static versus Dynamic Routing Decisions	48
Figure 3.2	Overall Logic of Simulation/Assignment Model .....	51
Figure 3.3	Flowchart of Simulation/Assignment Model	54
Figure 3.4	Illustration of Vehicle Moving Process .	64
Figure 4.1	Flowchart of Shortest Path Algorithm ...	81
Figure 4.2	Convex Combination Method (CONCOMNET) ..	84
Figure 4.3	Framework of Simulation Module .....	87
Figure 4.4	Traffic Simulation Subroutine (TRAFSIMULA) .....	91
Figure 4.5	Flowchart of Subroutine ARRIVE .....	97
Figure 4.6	Flowchart of Subroutine TRAFMOVE .....	100
Figure 5.1	Test Network .....	105
Figure 5.2	Scenario 1 .....	111
Figure 5.3	Static versus Dynamic Traffic Assignment .....	116
Figure 5.4	Scenario 2 .....	118
Figure 5.5	Scenario 1 versus Scenario 2 .....	119
Figure 5.6	Scenario 3 .....	121
Figure 5.7	Scenario 4 versus Scenario 1 .....	123

## List of Tables

Table 3.1	Effect of Simulation Time Step .....	71
Table 3.2	Effect of Section Length on Computational Efforts .....	72
Table 5.1	Selected Routes .....	109
Table 5.2	Travel Times for User Equilibrium .....	112
Table 5.3	Travel Times on Selected Routes .....	113
Table 5.4	Total Number of Used and Congested Link	115
Table 5.5	Average Travel Times in System .....	125
Table 5.6	Average Travel Times in System, as Percentage of Base Case with no Information .....	125

# 1.0 INTRODUCTION

## 1.1 URBAN/SUBURBAN CONGESTION PROBLEMS IN THE UNITED STATES

Traffic intensity on freeways, particularly in cities continues to increase at a rapid rate, and peak-period traffic congestion is an everyday occurrence in most urban areas. In most cities "sixty percent of rush hour traffic on urban interstates was rated as congested in 1985, compared to forty percent in 1975. The number of cars owned in the U.S.A. tripled between 1960 and 1986 and total annual motor vehicles miles increased from 719 billion to 1,861 billion in the same period" [Koutsopoulos, 1989]. It is widely accepted that the limited construction of new facilities will not alleviate this growing urban and suburban problem which causes excessive delays, traffic bottlenecks, and increased fuel consumption. On the other hand, many transportation engineers are focusing on improving system usage of existing facilities in urban areas through traffic control. It appears that the implementation of traffic system management methods is one of the most effective ways of promoting a better use of transport

facilities in an urban area; this type of measure is most useful when the economic and the environmental conditions prevent big expenditures on new roads. The current problems caused by the excessive traffic congestion, generated the need for the development of an Intelligent Vehicle/Highway System (IVHS) program that is capable of making significant improvements in mobility, highway safety, and productivity on highways and urban streets.

## **1.2 THE ROLE OF IVHS IN ADDRESSING URBAN/SUBURBAN CONGESTION PROBLEM**

The use of information technology including advanced computers, communicators, sensors and associated electronics systems for integrating vehicles and highways in a total system control constitutes the concept of IVHS. The objective of that integration is to improve traffic flow, operational safety and the overall efficiency of highway transportation. The IVHS system has been grouped into four functional subsystem including:[Zhang 1991]

- Advanced Traffic Management System (ATMS)
- Advanced Traveller Information System (ATIS)
- Advanced Vehicle Control (AVC) System

- Commercial Vehicle Operations (CVO) System

Advanced Traffic Management System (ATMS) and Advanced Traveller Information System (ATIS), are considered as important tools to alleviate growing problems caused by urban traffic congestion. ATMS and ATIS provides drivers with timely and appropriately designed traffic information, coupled with route guidance which can maximize the efficiency of existing highway facilities (Operator's point of view) or minimize the travel times and the cost of users (User's point of view).

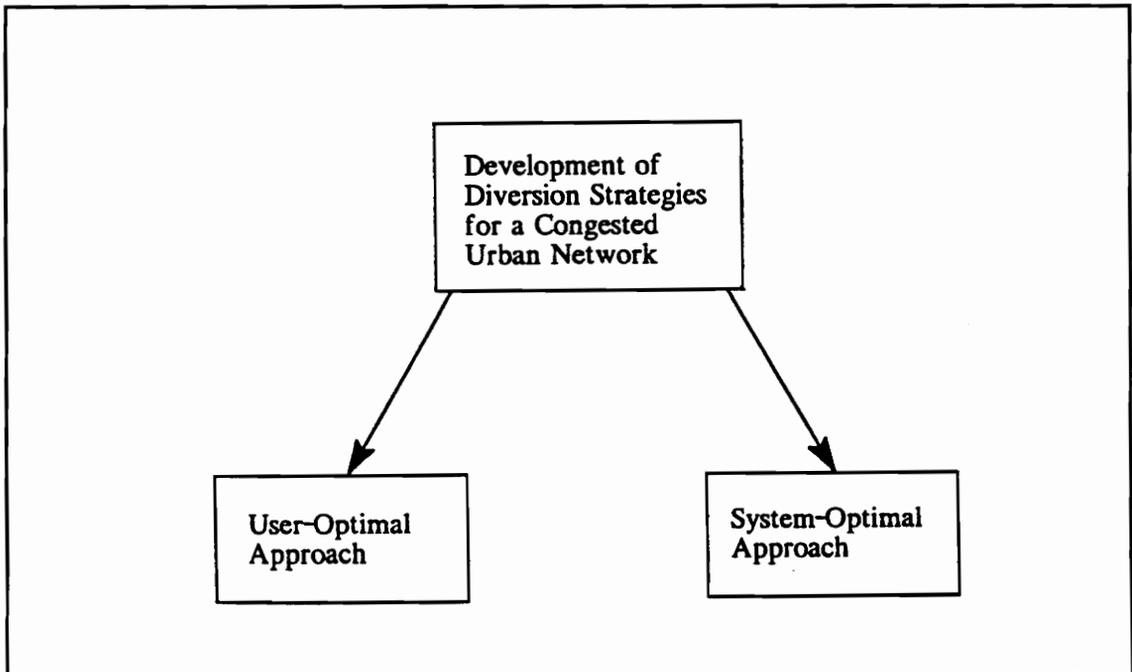
A comprehensive route guidance and drivers information systems should be able to assemble and disseminate comprehensive route guidance information to virtually all drivers. Drivers receive the information about alternative routes for their destinations through an information system formed of 3 compatible subsystems:

- pre-trip information systems (PTRD)
- changeable message signs (CMS)
- in-vehicle route guidance systems

The system as a whole will be capable of updating the routing decisions in real time by using the real time information. This will allow the entire system to immediately respond in real time to any changes in network traffic conditions.

After having received the real-time information, an appropriate route diversion decision should be made by either the driver or the controller. The implementation of the routing/decision strategies will depend on the modeler's approach to the problems. If we approach the problem from the users point of view, drivers want to receive the information that best meet their individual needs in terms of reaching their destination quickly. In this case the route choice will be based on the subjective evaluation of route attributes, such as travel times, for both the regular and alternative routes. It is clear that the user's objective will be to minimize his/her travel time and this may not be the optimal solution for the system wide usage. Thus, another approach that tries to optimize system wide usage can be adopted.

These are the two approaches taken by the Center for Transportation Research at Virginia Tech in initiating this research. Keeping in mind that there are two approaches for diversion strategies, user optimum and system optimum, two different models have been built. The two different approaches have been shown in Figure 1.1.



**Figure 1.1: The Two Approaches of Developing Diversion Strategies**

### **1.3 ROLE OF DYNAMIC TRAFFIC ASSIGNMENT IN DEVELOPING DIVERSION PLANS**

The success of the Advanced Traffic Management System, is highly dependent on new methods that can perform real time dynamic traffic assignment and traffic simulation to develop effective diversion strategies during congestion causing events. Many state departments of transportation have predetermined diversion plans, and in the event of an accident

or congestion, the appropriate plan is chosen for implementation. However, pre-determined diversion plans are of limited use since the time and the location of most accidents or congestion causing events are hard to predict. Hence, the development of efficient re-routing plans to suit the situation needs to be executed after the incident takes place, in real time conditions. This poses a challenge to traffic engineers and managers. The task is difficult on two accounts: (1) the complexities of predicting alternate diversion routes due to dynamic networks of traffic flow and complex user behavior and (2) the computational difficulties associated with evolving plans in a short time period, depending on the size of the problem.

Although the need for an effective dynamic traffic assignment model capable of evaluating dynamic route guidance strategies is evident, currently there is no computer model which can provide this information in real-time. Most of the current models are in the development stage; in other words, they are not complete yet. Many of IVHS projects like European DRIVE and PATH of California, are now seeking the development of reliable, reasonably fast network models which can be used to test and generate various control and routing strategies in a realistic, dynamic setting. The development of a realistic, real time dynamic traffic assignment model is

then one of the most needed tools to provide an effective traffic management system in the context of IVHS technologies.

## **1.4 PROBLEM STATEMENT AND RESEARCH OBJECTIVES**

The overall goal of the research effort is to develop a tool to assist traffic managers in developing efficient diversion strategies in real time.

The user optimal approach adopted in this research, is shown in Figure 1.2. The basis for this approach is to re-route the motorists through the best (shortest) alternate paths to their destination. Major components of this research is to develop, test, and evaluate a dynamic traffic simulation / assignment model for initiating diversion strategies.

The main effort in phase 1 of the project was devoted to developing appropriate models for user optimal diversion planning. Realizing the necessity of traveller destination information for effecting user-optimal diversions, a destination estimation model was developed [Hobeika et al., 89]. This model utilizes link traffic volumes to estimate traveller destinations. It is based on a linear programming approach, employing an efficient column technique. Complete

details of the model are present in the phase 1 project report [Hobeika et al., 89].

After the application of the developed destination estimation model to predict origin-destination trips for areal network, the next step is the development of effective diversion strategies during congestion causing events.

Realizing the importance of dynamics of traffic flow in real time diversion planning, the need for developing a dynamic traffic assignment model that is capable of representing time dependent behavior of the traffic was acknowledged. Considering this need as the main factor that initiated this specific research, the following steps were followed to respond to the main requirements of the overall research project:

- 1) Investigate different approaches to the dynamic traffic assignment problem.
- 2) Select the approach that best satisfies the majority of the research (Finally, a simulation/assignment method is chosen).
- 3) Develop a conceptual and a computer model to perform dynamic traffic assignment.
- 4) Apply the model to a test network.
- 5) Evaluate the dynamic traffic assignment model to test its sensitivity for different parameters, such as different

# USER-OPTIMAL DIVERSION PLANNING

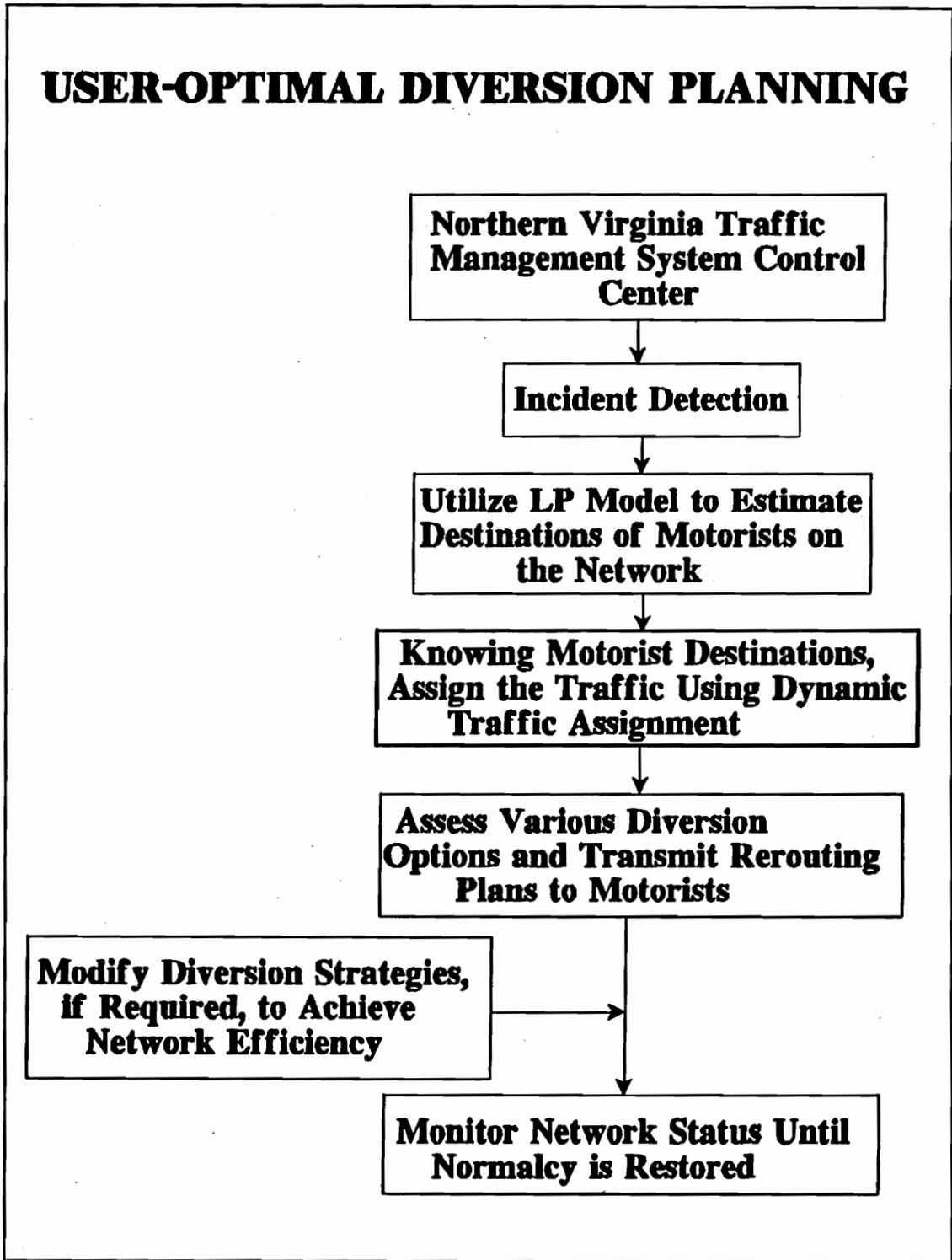


Figure 1.2: User Optimal Diversion Approach

fractions of users that have an access to information, behavioral rules, and different loading patterns.

- 6) Discuss the major differences between dynamic and static traffic assignments.

## 1.5 SUMMARY OF THE OTHER CHAPTERS

A brief introduction to the urban/suburban problem, the role of IVHS in solving this problem, and the main goals of this research were discussed in this chapter. The second chapter is devoted to a broad review of literature on static and dynamic traffic assignment. In this review, the shortcomings and disadvantages of both static and dynamic traffic assignment models are discussed. In Chapter 3, the theoretical aspects of algorithms used in developing simulation/assignment model are discussed. The detailed description of each subroutine of the computer program, and the interaction among them are presented in Chapter 4. The results of the model runs for a test network are presented and discussed in Chapter 5. Finally, in Chapter 7, conclusions and recommendations for future research works are discussed.

## **2.0 LITERATURE REVIEW**

### **2.1 TRAFFIC ASSIGNMENT AND ADVANCED DRIVER INFORMATION SYSTEMS (ADIS)**

The idea of using in-vehicle guidance systems as a tool for giving drivers real time traffic information on congestion is a promising technique for making better use of existing highway capacity. Potential benefits of in-vehicle guidance systems are extensive. Directing a driver to a destination through a network in order to eliminate navigational waste, and informing drivers of congested routes, or parking availability can be cited as some of the potential benefits of these systems.

This real time traffic information can be supplied to the users through various possible forms of advanced driver information systems (ADIS). The main functions of ADIS are:

- 1) To facilitate long term decisions of the drivers with respect to their travel patterns and route choices
- 2) To provide real time information on actual network conditions for avoiding bottlenecks and delays due to incidents.

The effectiveness of ADIS depends on the quality and reliability of the information supplied. This can be achieved by using a traffic assignment method that provides drivers with the optimal routes. Therefore there is need for a dynamic traffic assignment system to obtain optimal routes for different origin-destination pairs, and to evaluate traffic control systems and route guidance strategies. The network traffic assignment problem has been studied for the last thirty years. Thus, decades of research have resulted in efficient algorithms that allocate trips to specific routes in a transportation system in order to minimize system wide and user travel cost. This minimization is subject to flow balance at every point in the network. Many of the available models assume steady state conditions where the attributes of the transportation system do not vary with time. This is the major shortcoming of traditional traffic assignment models. They do not have time dimension, and they assume that all cars assigned to the network are present on all links at the same time. This assumption can give unrealistic results when traffic assignment is used for real-time route guidance operations usually carried by traffic control centers. It is clear that those conventional assignment methods are not likely to be sufficient for representing the dynamic nature of traffic. Thus, a dynamic traffic assignment model that can

incorporate time as a third dimension, can predict the time dependent nature of traffic much more realistically.

Before mentioning the different approaches used to formulate and solve dynamic traffic assignment, traditional traffic assignment models need to be summarized. This is a necessary step to understand dynamic traffic assignment methods, because most of them try to capture dynamic behavior of traffic flow by improving or reformulating existing static traffic assignment methods.

## **2.2 STATIC TRAFFIC ASSIGNMENT MODELS**

The traffic assignment problem can be defined as the process of determining a route or routes of travel of inter-zonal trips and allocating trips to those routes [FHWA, 1972].

The primary problem addressed by the static traffic assignment is the distribution of traffic flows through an urban network. The approach used to solve this problem is based on the notion of equilibrium. The equilibrium is between link performance (travel time) and traffic demand. The traffic flow is determined in terms of the number of travelers (or vehicles). The travel time is dependent to the traffic flow. If the traffic flow increases, the travel time increases

also. Therefore, the problem is to find out the equilibrium point between travel time and travel demand.

The solution of the problem is based on the behavioral assumption that each motorist chooses the path that minimizes the travel time from his or her origin to destination. This choice, also known as the Wardrop's first principle, implies that at equilibrium the link flow pattern is such that the travel times on all used paths connecting any given O-D pair will be equal, and the travel time on all used paths will also be less or equal to the travel time on any of the unused paths. When those conditions are satisfied, the network is in user equilibrium and no motorists can experience a lower travel time by unilaterally changing routes.

A mathematical approach, and an iterative approach are commonly used to solve the static equilibrium assignment problem. The iterative approach is a heuristic one that is easy to understand and apply. However, it has no global optimal, and it might have different local optimal points. The solution obtained from iterative assigned is an acceptable one, but it is not necessarily the optimal solution. On the other hand, the most common model used from the mathematical approach is the convex combination model developed by Nguyen [Nguyen, 74].

## 2.2.1 HEURISTIC STATIC ASSIGNMENT TECHNIQUES

The heuristic approach can be divided into two: capacity restraint methods and incremental assignment techniques. The capacity restraint method involves a repetitive all-or-nothing assignment in which the travel times resulting from the previous assignment are used in the current iteration. The steps of the modified capacity restraint method can be summarized as follows:

Step 0: Initialization. Perform all-or-nothing assignment based on  $t_a^0 = t_a(0)$ . This yields  $x_a^0$ . Set counter  $n=1$

Step 1: Update. Set  $t_a^n = t_a(x_a^{n-1})$ ,  $\forall a$

Step 2: Smoothing. Set  $t_a^n = 0.75t_a^{n-1} + r_a^n$ ,  $\forall a$

Step 3: Network loading. Perform all-or-nothing assignment based on  $t_a^0 = t_a(0)$ . This yields  $x_a^n$ .

Step 4: Stopping rule: If  $n=N$  goto step 5. Otherwise, set  $n=n+1$  and go to step 1

Step 5: Averaging. Set  $x_a^n = .25 \sum_{l=0}^3 x_a^{n-1}$ ,  $\forall a$  and stop. ( $x_a^n$  are the link flows at equilibrium) [Sheffi, 1985]

In this algorithm,  $t_a^n$  is the travel time on link "a" at iteration "t",  $x_a^n$  is the flow on link "a" at iteration "n", and "l" is the total number of links. The algorithm uses a

smoothing effect at step 3 to facilitate convergence. If it still fails to converge, it is terminated after a certain number of iterations. Equilibrium flow pattern is then taken as the average flow for each link over the last 4 iterations. The modified capacity restraint algorithm uses weights of .75 and .25 for the averaging process of travel times.

Incremental assignment is another heuristic method for attaining the user-equilibrium solution. It assigns a portion of the origin-destination matrix at each iteration. After each iteration, the travel times are updated and an additional portion of the O-D matrix is loaded onto the network. Although it is reasonable to believe, in general, as the number of increments grows, the incremental assignment algorithm may generate a flow pattern closer to the user-equilibrium condition, it may as well, like other heuristic methods, fail to converge.

Heuristic methods briefly outlined above do not converge or produce a set of flows that are not in agreement with the user-equilibrium criterion. Therefore, a better method of assignment is needed. As mentioned before, the user equilibrium problem can be formulated as a mathematical program and then this program can be solved. The mathematical approach will be discussed in detail and will be used for solving equivalent user equilibrium program.

## 2.2.2 MATHEMATICAL PROGRAMMING TECHNIQUE

To reach an equilibrium point is not always possible if a heuristic algorithm is used, instead a mathematical programming technique can be used. The convex combination algorithm [Sheffi, 1985] that includes a convex (non-linear) objective function, and a linear constraint set is suitable for solving the equivalent UE program. The direction finding step of the algorithm is executed efficiently through the use of a linear program. Given a known matrix of origin-destination flows, a network of directed links connecting nodes, and link performance functions that relates link costs to the corresponding flows, convex combination algorithm can solve single class user equilibrium problem [Mahmassani 88]. It loads the flows onto the individual links of the network, in such a way that its solution is consistent with Wardrop's principles stating that no driver can improve his/her travel time by switching routes once the equilibrium is reached.

The steps of the algorithm can be summarized as:

Step 0: Initialization. Perform all-or-nothing assignment based on  $t_a^0 = t_a(0)$ ,  $\forall a$ . This yields  $x_a^1$ . Set counter  $n=1$

Step 1: Update. Set  $t_a^n = t_a(x_a^n)$

Step 2: Direction finding. Perform all or nothing assignment

based on  $t_a^n$ . This yields a set of auxiliary flows,  $y_a^n$

Step 3: Line search. Find  $\alpha_n$  that solves

$$\min \sum_a \int_0^{x_a^n + \alpha (y_a^n - x_a^n)} t_a(\omega) d\omega \quad \dots 1.1$$

Step 4: Move. Set  $x_a^{n+1} = x_a^n + \alpha_n (y_a^n - x_a^n), \forall a$

Step 5 : Convergence test. If a stop criterion is met, stop(current solution),  $x_{a+1}^n$ , the set of equilibrium flows; Otherwise, set  $n=n+1$  and go to step 1 [Sheffi, 1985].

The algorithm uses an iterative procedure in order to minimize the total travel time spent in the network by assigning all cars to the shortest path connecting their origin to their destination. At each iteration, it first finds a search direction by solving a linearized approximation, then solves for the optimal move size along that direction. The direction finding part of the algorithm is performed by the all-or-nothing network loading procedure. This procedure needs the repeated determination of the shortest paths between origins and destinations.

The line search to find the optimal move size is performed by using the bisection method. This method is easily applied to that case because the derivative of the objective function  $z[x^n + \alpha(y^n - x^n)]$  with respect to  $\alpha$  can be calculated for any value of  $\alpha$  without any major difficulty.

The derivative of that function is:

$$\frac{\partial}{\partial x} z[x^{n+\alpha}(y^n-x^n)] - \sum_a (y_a^n - x_a^n) \quad \dots 2.2$$

If it is possible to evaluate the derivative of the function to be minimized, the method of interval bisection can be used to find the optimal move size. The derivative of the objective function  $z[x^{n+\alpha}(y^n-x^n)]$  is easily determined as shown above. Therefore, the bisection method is applicable to this case.

Another important issue associated with convex combination algorithm is the stopping criteria. The algorithm stops if a criterion that checks the change in flows is satisfied. The criterion that is used in this case can be formulated as:

$$\frac{\sqrt{\sum_a (x_a^{n+1} - x_a^n)^2}}{\sum_a x_a^n} \leq K' \quad \dots 2.3$$

The similarity between this algorithm and heuristic capacity restraint method summarized in Section 2.2.1 is very clear. In fact, if the move size  $\alpha_n$  is fixed at  $\alpha_n=1$  for all

n, the resulting algorithm is identical to the capacity restraint method. The main mechanism of the convex combination method is to take the flow away from the congested paths and assign them to less congested paths at each iteration. This is determined by the move size  $\alpha_n$ . This process equalizes the travel times among all paths and brings the system toward equilibrium. The number of iterations needed for convergence is related to the congestion level on the network. In relatively uncongested networks, only one iteration may be sufficient to achieve the equilibrium. On the other hand, as congestion builds up, more iterations are required for the network to reach equilibrium.

## 2.3 DYNAMIC TRAFFIC ASSIGNMENT MODELS

A dynamic user equilibrium model can be described as a pattern of travel in which some functions of trip making, such as cost supplies and demands vary over time and in which no individual can unilaterally alter his or her trip making so to achieve a lower personal cost [Hendrickson et al., 83]. It is the explicit time dependence of network functions that distinguishes dynamic and steady state equilibrium. Due to the complex nature of the dynamic traffic assignment, there have been different approaches for the solving it.

### 2.3.1 LINEAR PROGRAMMING APPROACH

The first attempt to formulate a dynamic traffic assignment was done by [Merchant et al., 78]. They presented a discrete time model for the dynamic traffic assignment. A traffic network is represented by a directed graph.

The analysis time is divided into a finite number of discrete periods. For each time period, external inputs are allowed at any node except destination. For each arc, there is an exit function which relates the amount of traffic entering and leaving the arc, during a time period. The model treats the congestion explicitly in flow equations. Congestion is modeled by assuming the exit functions to be non-decreasing, continuous and concave.

The basic flow equations in the model are:

$$x_{i+1,j} - x_{i,j} - g_i(x_{i,j}) + d_{i,j} \quad i=0, \dots, N-1, \quad \forall j \in A \quad \dots 2.3$$

$$\sum_{j \in A(q)} d_{i,j} - F_i(q) + \sum_{j \in B(q)} g_j(x_{i,j}), \quad i=0, \dots, N-1, \quad \forall q \neq n \quad \dots 2.5$$

Where,

$A(q)$  = set of arcs that leaves node  $q$

$B(q)$  = set of arcs that enters node  $q$

$F_i(q)$  = external input at node  $q$  in period  $i$

$x_{ij}$  = amount of traffic on arc  $j$  at the beginning of period  $i$

$g_i(x_{ij})$  = amount of traffic to exit from arc  $j$  in period  $i$

$d_{ij}$  = amount of traffic admitted to arc  $j$  in period  $i$

$h_{ij}(x_{ij})$  = cost of  $x_{ij}$  (the cost of these terms is to be minimized)

$N$  = analysis time

$i$  = index of time period  $i=0,1,\dots,N$

The model is a non linear non convex mathematical programming problem. The overall objective is to provide a macro-model that will find the feasible traffic flow, minimizing the total cost function which is assumed to be the sum of non-decreasing, non-negative, continuous convex functions of arc flows. Although it has the capability of handling multiple origins and general topology, it is limited to one destination.

The time-varying flows which minimize the flow costs are obtained as follows.

minimize:

$$\sum_{i=1}^N \sum_{j=1}^a h_{ij}(x_{ij}) \quad \dots 2.6$$

subject to

$$x_{i+1,j} - x_{ij} + g_j(x_{ij}) + d_{ij}, \quad i=0, \dots, N-1, \quad \forall j \in A \quad \dots 2.7$$

$$\sum_{j \in A(q)} d_{ij} - F_i(q) + \sum_{j \in B(q)} g_j(x_{ij}) \quad i=0, \dots, N-1 \quad \forall q \in N-n \quad \dots 2.8$$

$$x_{0j} - R_j \geq 0 \text{ (given)}, \quad \forall j \in A \quad \dots 2.9$$

$$d_{ij} \geq 0, \quad i=0, \dots, N-1, \quad \forall j \in A \quad \dots 2.10$$

$$x_{ij} \geq 0, \quad i=1, \dots, N, \quad \forall j \in A$$

...2.11

where  $R_j$  is the initial (period 0) volume on arc  $j$ . The constraint 2.7 and 2.8 ensure conservation of flows on arcs, and constraints 2.10 and 2.11 ensure conservation of flows at nodes. Ho proved that a piecewise linear version of the model, with additional assumptions on the objective function, can be solved for a global optimum using a one-pass simplex algorithm. The algorithmic questions of implementing the model has been resolved by Ho [Ho 80]. He presented an efficient condition for optimality which implies that a global optimum can be obtained by successively optimizing at most  $N+1$  objective functions for the linear program, where  $N$  is the number of time periods in the analysis time. Ho presented a successive linear optimization algorithm and computational results. Based on the computational results, it is stated that only when the network is extremely overloaded that a significant number of successive optimizations is required to obtain a solution. In more realistic cases even with substantial congestion, he states that successive linear optimization is an efficient approach to the dynamic traffic assignment problem.

### 2.3.2 OPTIMAL CONTROL THEORY APPROACH

The optimal control theory is applied to the problem of dynamic traffic assignment by [Friesz et al., 89]. The method can be considered as extensions of steady-state network equilibrium model, particularly Beckman's equivalent optimization problem, which is a mathematical programming formulation [Beckman, 52]. The relaxation of the steady state assumptions of this formulation leads to the problem of dynamic traffic assignment in which network characteristics are explicit functions of time. The analysis is restricted with one O-D pair that is connected by N parallel arcs. The dynamic model developed by [Friesz, 89] has similarities with models proposed by [Hendrickson et al., 84], [Ben-Akiva et al., 86], and [Mahmassani et al., 84], but it significantly differs in its formulation as a continuous time optimal control problem. This model formulates the problem of dynamic traffic assignment as an equivalent time optimal control problem. It only considers the route choice decision making because travel demand at each instant is assumed to be known and inelastic. The optimality condition of the dynamic model requires only equalization of instantaneously perceived unit path costs. Another interpretation of the optimality conditions can be found in Wardrop's first principle which

states that at equilibrium, travel time on all used paths is equal, and less than or equal to the travel time that a user would experience on any unused path.

An extension of the model mentioned above is developed by [Wie, 91]. The particular extension to the model is to include elastic time varying travel demand which implies the departure time choice. This new model considers that drivers have the perfect information and can estimate instantaneous perceived path costs from any route on the network to the destination node, on the basis of continuously updated traffic information. The optimality conditions remain the same as in the original model. "The problem considered by Friesz and Wie can be viewed as a type of noncooperative dynamic game, in which network users act independently without collaboration and compete with one another for the limited network capacity through route and departure time choices" [Wie, 91].

### **2.3.3 HEURISTIC APPROACHES**

Two different models can be cited as the heuristic approaches to the dynamic traffic assignment. They are developed by [Hammerslag, 89] and [Janson, 89].

The steps of the Hammerslag's algorithm are as follows:

- 1) Read a 2-D network where time dimension is not considered.

- 2) Determine the 3-D O-D matrix where third dimension is the time.
- 3) Determine the period capacity of the links
- 4) Calculate the delays in the links
- 5) Calculate the shortest routes in 3-D space
- 6) Assign the 3-D O-D matrix to the shortest routes
- 7) Load the network
- 8) if stop criterion has not been reached, return to step 4

Hammerslag's model is iterative and can be looked as an extension of iterative static assignment models. The main difference is that it divides the one hour origin-destination matrix into fifteen minute origin destination matrices and assigns them to the shortest routes determined in 3-D space. In 3-D space the determination of routes is similar to the 2-D space, however the route determination in 3-D space is done for all periods, rather than just one period, as in 2-D space. Unlike the static assignment that determines only one shortest path and assigns all cars of an O-D pair to all links along the shortest route, this algorithm assigns cars to different links during different time periods, because in 3-D space links are used during different periods. The ratio of the cars that are assigned to different links during different time periods is found by using a simple graphical method. Loading the network is done by considering of the all-or-nothing

assignment flows just calculated and flows from the previous iteration. To do this, the following equation is used:

$$q_{jkp}^i = q_{jkp}^t * g^i + q_{jkp}^{i-1} \quad \dots 2.12$$

where,

$q_{jkp}^i$  = the number of cars on link  $kj$  during period  $p$  at iteration  $i$

$q_{jkp}^t$  = the assigned traffic to the shortest routes in the network during previous iteration

The value of  $g^i$  depends on the number of iterations " $i$ " and is chosen in such a way that there are no overloaded links. The value of  $g^i$  is determined by using the following equation:

$$g^i = \min[1/(i+1), (q_{jkp}^{i-1} - C_{jkp}) / q_{jkp}^t] \quad \dots 2.13$$

The main problem associated with this model is the difficulty to reach an equilibrium and the computational effort required. It is reported by Hammerslag that the calculation time necessary for the model is ten times higher than an equilibrium assignment.

The other heuristic model has been developed by Janson. It has similar steps to the Hammerslag's model, but it is not

iterative. The model tries to determine the optimum shortest route from each origin to destination at the time departure. These route choice decisions are based on projected link impedances that account for changes in travel demand over future time intervals. Users are assumed to travel on this route and they are not allowed to switch to another route in the middle of their journey. This assumption simplifies the assignment procedure because the trip departure matrix assigned at each time interval is only a zone-to-zone trip matrix, and not a node-to-zone trip matrix which would be required to track trips through the network by their destinations so that paths could be revised en-route.

One important aspect of the algorithm is the way in which current link volumes are projected into future time intervals for the purpose of the finding shortest paths. This is because some trips departing in later time periods will concurrently use the same link with trips from the present origin for which paths are being found. It is very clear that these new trips will effect the travel time on those links. Therefore, while determining the shortest route for current link volumes, the effect of the future link volumes should be recognized. The projected link volumes are calculated according to equation 2.14:

$$y_k^{t+n} = \gamma \bar{\omega}_{t-1}^{t+n} * x_k^{t-1} + (1-\gamma) \bar{\omega}_t^{t+n} x_k^t \quad \dots 2.14$$

where,

$Q^t$  = total number of trips departing from all zones in time interval  $t$

$$\bar{\omega}_{t-1}^{t+n} = Q^{t+n} / Q^{t-1}$$

$y_k^{t+n}$  = projected volume on link  $k$  in interval  $t+n$

$\gamma^t$  = percent of  $Q^t$  that has not yet been assigned to the network

$x_k^t$  = assigned volume on link  $k$  in the current time interval " $t$ " after trips departing in time intervals 1 through " $t-1$ " have been assigned, and while trips departing in time interval " $t$ " are being assigned

Thus, for each link, the current and projected link volumes are estimated as weighted combinations of the final volume assigned to that link in the previous interval " $t-1$ " and the volume assigned thus far in the current interval " $t$ ", weighted by ratios of total trip departures from all origins in intervals " $t+1$ ", " $t$ ", and " $t-1$ ". Each projected link impedance is computed directly from its projected volume using its impedance function.

Because the algorithm can project the link volumes into the future, it can also predict travel times on links in the future. Thus, by knowing the future travel times, the path search routine used by this algorithm can determine the shortest route for each O-D pair, and the time intervals in which links belonging to the shortest route are going to be used. Then, for each period, it can assign trips to the links that they are going to use during that specific period.

Although the method seems useful, the projection function is too much arguable. A better function that can predict link volumes will obviously increase the credibility of the results of the model. For this model, convergence is not guaranteed, but the computation time is not as high as in the first method.

#### **2.3.4 DYNAMIC MODELS OF PEAK PERIOD CONGESTION**

Dynamic models of peak period congestion are generally used to represent the interaction between user decision and system performance. These models try to capture an important dimension of choice available to users for combatting congestion; this is the departure time choice.

Problems concerning departure time choice are beyond the scope of this research, for that reason those models are not

going to be described in detail. However, those models are very useful to understand real behavior of traffic and they need to be reviewed briefly.

A number of recent studies have addressed the time dependent demand pattern arising in an idealized situation of commuters departing from a given origin to a single destination along a unique route where congestion occurs at a single bottleneck. [Hendrickson and Kocur, 81] developed a simple model of the departure time pattern. The model assumes that arrival times are known and user travel cost includes travel times in addition to a penalty for late or early arrivals. Extensions of the original model to a case where travel times are stochastic were presented in a subsequent paper by [Hendrickson et al., 83].

Later [Ben-Akiva et al., 84] incorporated a probabilistic departure time choice model of the continuous logit form within the Hendrickson's equilibrium approach, and [Ben-Akiva et al., 86] used the same model to simulate the effects of congestion reducing measures along a route.

Finally [Mahmassani et al., 84] extended the same approach of Hendrickson to the situation where the user can adapt to congestion by not only changing departure times but also changing routes.

Although a number of simplifications have been made,

those models are only applicable to very simple networks and are not practical for large ones. [Mahmassani et al., 91] in their paper states that more detailed, simulation-based performance models are needed to understand and model the dynamics of user decisions on transportation networks.

### **2.3.5 SIMULATION / ASSIGNMENT APPROACHES**

Approaching the traffic assignment issue using simulation / assignment techniques is promising, especially along recently developed directions using improved integration of simulation with conventional techniques. A number of models that can handle complex network structures and are responsive to traffic management schemes have been developed. Among those models, are SATURN and CONTRAM, which are simulation / assignment models used for the evaluation of traffic management schemes. In addition to SATURN and CONTRAM, models developed by [Stephanedes, 90] and [Mahmassani, 85] can be cited as examples of simulation / assignment type of models. Their common characteristic is an iterative loop between simulation and assignment phases which operates until the stopping criteria set by the user is satisfied.

### 2.3.5.1 SATURN

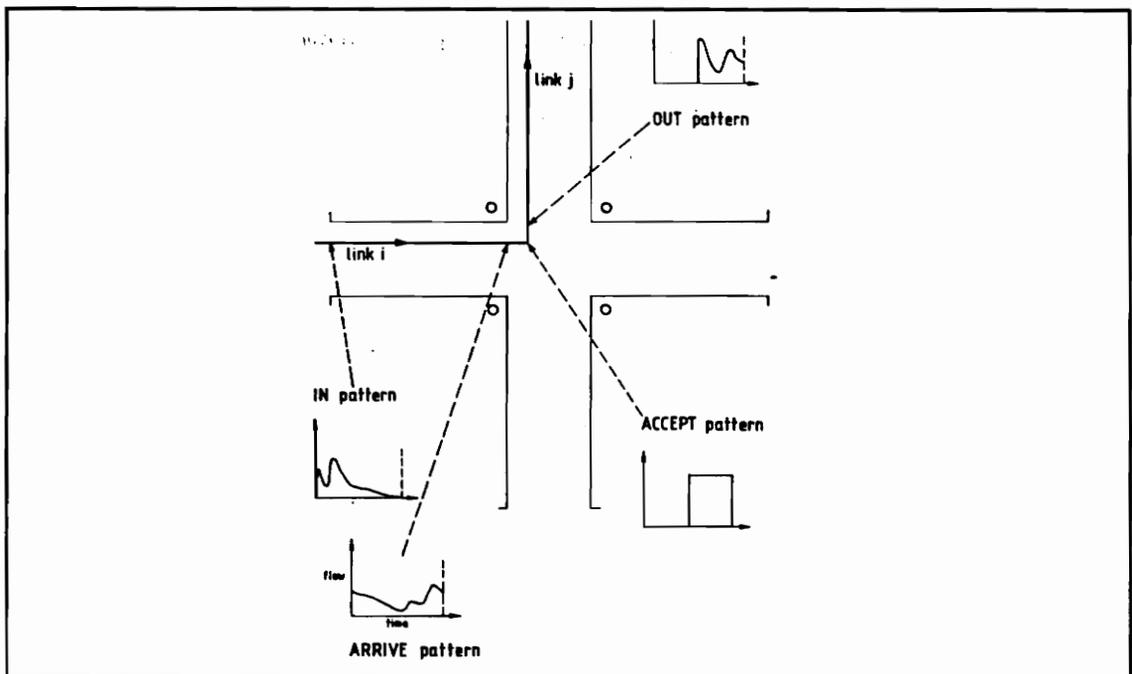
Saturn has been designed to evaluate schemes that affect traffic movement, such as one way streets, banned turns, bus only lanes and pedestrianization changes [Van Vliet et al., 79]. It is less suited to deal with measures whose primary effect is to change trip origins and/or destinations or trip departure times. SATURN incorporates two phases:

1) A detailed simulation model which has the aim of determining junction delays resulting from given patterns of traffic. The simulation model has two basic assumptions:

- a) the pattern of traffic flows is constant over time periods of the order of thirty minutes;
- b) a cyclical behavior is imposed on the flows by traffic signals operating with a common cycle time (typically in the range 60-120 seconds). This assumption allows the simulation to concentrate on a single cycle making a detailed simulation possible.

Within each cycle, the flow is represented as semi-continuous flow profiles as opposed to modeling individual vehicles or packets of vehicles. The cyclic flow profile (CFP) is the flow of traffic past a certain point as a function of time over the cycle length (the cycle length being that of the traffic signals in the network). The model can be

seen as the collection of different continuous flow profile patterns that are attached to individual turning movements. The four basic cyclical flow profiles for a turning movement are: IN pattern, ARRIVE pattern, ACCEPT pattern, and OUT pattern (Figure 2.1). They provide the basis for the detailed analysis of delays. These profiles are interrelated.



**Figure 2.1: The Four Basic Cyclic Flow Patterns**

The ARRIVE pattern is derived from the IN pattern by a process of platoon dispersion. The ACCEPT pattern is derived independently and is based on junction capacities, signal timings and offsets, and conflicting traffic. It also takes

into account the effect of queued traffic. The OUT pattern is derived from the ARRIVE and the ACCEPT patterns, and contributes to the total IN patterns for the succeeding turns. As a result of the contribution of the OUT pattern at a node to the IN pattern of succeeding links, the traffic is moved through the network. The simulation model is used to model the flow-delay curves by calculating the delays for each turning movement at zero flow, current flow and capacity with all other flows fixed. Given those three points a flow-delay curve is obtained for a specific junction. Then this flow-delay curves are used in the assignment phase.

2) An assignment model that selects for each element in the trip matrix minimum time routes through the network considering the relationships between travel time and flows. The model uses an equilibrium technique that optimally combines a succession of all-or-nothing assignments such that the ultimate flow pattern satisfies the Wardrop's equilibrium criterion.

The assignment model has two fundamental assumptions:

- a) travel time of each link is fixed independent of flow;
- b) the delay to each turning movement at an intersection is a function of that turning volume.

The complete model is based on an iterative loop between the assignment and simulation phases. As mentioned in the

simulation phase, the simulation model determines flow delay curves based on a given set of turning movements and feeds them to assignment. The assignment in turn uses these curves to determine route choice and hence updated turning movements. These iterations continue until the turning movements reach reasonably stable values.

SATURN requires two type of data. The first is the O-D trip matrix of the period of interest. The second input is network data. Network data is described as a set of nodes and connecting links. Additional node based information is also needed. This data refers to junction type, free run times, link lengths, lane structures, saturation flows and types. SATURN output consist of detailed performance measures at every intersection including delays for each turn flow and queue profiles and warning of congestion. In addition, standard network-wide performance indices such as total vehicle-hours are available.

#### **2.3.5.2      CONTRAM**

CONTRAM ( which stands for CONTinuos Traffic Assignment Model) is also a simulation assignment model that treats networks of a similar size to SATURN [Leonard et al., 76]. CONTRAM deals with time-varying conditions in a network. It

models time varying route choice, queuing and delays in urban networks with time-varying traffic demands and signal or priority control at junctions.

The network of streets in town is represented by a series of unidirectional links and junctions. Each link is one-way and may fan out up to 5 downstream links(fan-in is not restricted). Link-to-link movements may be restricted to certain vehicle classes. Traffic demands enter the network at origins and leave at destinations. Origins and destinations may occur within the network as well as on the periphery. Time variation is modeled by dividing the simulation period into a number of consecutive time intervals of fifteen minutes. The demands for each origin-destination movement are specified as flow rate(vehicle/hours) for each time interval. Traffic demands are also time varying. They are defined as vehicle /hour mean entry flow rate in each of up to 12 time slices for a number of origin, destination and vehicle class combinations.

There are three standard vehicle classes: car, bus, and goods vehicle. Each vehicle class is characterized by:

- 1) cruise time as a percentage of car cruise time
- 2) passenger car units (PCU) - effective number of units consumed by each vehicle class(eg: car=1, bus=2, good

vehicle=1.5)

- 3) independent origin destination demands and route
- 4) independent behavioral cost functions (optional)

Vehicles from each origin-destination pair are grouped to form packets. Each packet contains typically 1-10 vehicles. The packets from any origin destination pair enter the network equally spaced in each time interval. The journey time for a packet along each link of its route consists of the cruise time (free running time), and delay time which is dependent on the level of traffic on a link and the method of junction control. There are three types of different junction models: signal controlled, give-away, and uncontrolled.

Delay calculations are based on estimates of the average queue on each link at the end of each time interval. The estimate of queue depends on the queue at the end of the previous time interval, the number of arrivals at the stop-line in the interval, the maximum rate at which vehicles can leave the link and the duration of the time interval. It is assumed that the queue varies linearly between the queue values at the beginning and the end of the interval. The delay for a packet is calculated from the length of the queue encountered by the packet at the time it reaches the stop-line. This can be described as follows:

stop-line arrival time = entry time to link + cruise time on link

The delay is then the time taken, for the queue encountered, by the packet to be discharged. The maximum rate of discharge is the maximum capacity of the link.

CONTRAM uses an iterative procedure to predict the patterns of routes, queues and delays on a network. An iteration of the model is completed when all packets of the vehicles are assigned to their minimum journey time route through the network. The traffic patterns obtained at the end of an iteration are used in the next iteration when each packet is reassigned to its new quickest route. Before the reassignment of each packet, the flow corresponding to that packet is removed from each link of the packet's shortest route determined in the previous iteration. As a result of this process, a packet does not experience delays due to itself when being assigned to a new route. Each packet is reassigned in the time order in which packets enter the network. Then the delays for a given packet are determined by considering the flows due to packets which have entered the network prior to that packet during the current iteration, and the flows due to subsequent packets assigned to routes in the previous iteration. Several iterations are needed to obtain full convergence. The convergence is attained when all

vehicles are assigned to their same routes during successive iterations. This iterative procedure can be considered as the day-to-day familiarization of the drivers to network and traffic conditions.

CONTRAM is able to model the growth and decay of queues from time interval to interval. It also includes the effects of over-saturation that occurs during peak periods and that causes the growth and decay of queues. When vehicles queue back along the full length of links, blocking upstream junctions, the flow of vehicles from links feeding these junctions is restricted .

There are three types of data required for CONTRAM: Network and Time, Traffic Demand (origin-destination movements), and Control data. Time data defines the time period to be simulated and the time intervals into which the total period is divided. The length for each time interval is chosen to allow an adequate modelling of traffic conditions over that interval in the simulation period. Thus time intervals are shorter when the traffic conditions are critical for the network. Typical lengths for time intervals are 10 to 15 minutes. Network data describes how origins, links and destinations are interconnected. The traffic demand for each origin-destination movement in a network is specified as a series of flow rates (vehicles/hour) for each time interval.

Control data has two parts. The first part controls the running time of the program, and defines the number of iterations and the types of output required. The second part provides additional data required for signal controlled junctions and data required for vehicles with a fixed route.

The output data for CONTRAM is calculated from link by link and route information stored during the assignment procedure. There are six forms of outputs provided by the model. Those are:

1) Total time spent in the network and distance travelled throughout the complete network for each time interval and total simulation period constitute the first form of data. These values can be used to examine convergence of the iterative process.

2) Changes in stop-line arrivals between consecutive iterations, for all links for each time interval are the second form.

3) Link-by-link data contain initial queue length, vehicle arrivals, final queue, spare throughout capacity, link store left, degree of saturation, total travel distance, total travel time spent, average speed of a car for each time interval

4) Summary of link-by-link values of flows includes data about

queues, queue times, and average speeds data selected from the previous data type

5) The form that gives the variation with time of the average speed for selected origin-destination movements is intended for use as a means for comparing the effects of traffic management schemes on overall journey speeds to ensure that improvements for one set of movements are not made at the expense of other movements

6) The last form of output provide detailed information about the variation in turning flows from each approach to a specified junction and from each link.

## **2.4 SUMMARY OF LITERATURE REVIEW**

There is an immediate need for a dynamic traffic assignment method that can be used in real-time traffic management, because static traffic assignment models that do not reflect the real behavior of traffic flow cannot be used for on-line applications. An overall evaluation of the literature on traffic assignment problem revealed that solution methods for static traffic assignment models are well established. However, solution methods for dynamic traffic assignment are still in the development stage.

There are two principal approaches for solving dynamic

traffic assignment problems: mathematical and heuristic approaches. Although the mathematical approach appears to be a good one, models adopted in this case are either too simple to be used in real world applications or too complex to solve. On the other hand, heuristic approaches, like SATURN, or CONTRAM seems to be more promising for immediate use, but both models use highly macroscopic simulation methods that causes a loss in the accuracy of traffic flow estimations. A model that will be used for IVHS purposes cannot tolerate such approximations, and needs a simulation approach which is more microscopic.

Therefore, a heuristic simulation/assignment model that can be used by a real-time traffic information system has been developed. This model by keeping track of the movements of vehicles in time differs from the existing models. The approach and the methodology used for developing the model are described in the following chapters.

## 3.0 THEORETICAL ASPECTS OF THE MODEL

### 3.1 INTRODUCTION

The goal of this research is to develop a heuristic dynamic traffic assignment method, and apply it to a test network in order to test its sensitivity to different parameters and to understand the differences between static and time-dependent dynamic traffic assignment. This is a simulation / assignment model which is called DYNTRAS (DYNAMIC TRAFFIC ASSIGNMENT/SIMULATION). The inputs to the model are the key physical and operational features of the network, the number of links, the length of the links, and the number of lanes, as well as the allowable free flow speed and the capacity of each section. The parameters of the speed-density model and the control parameters of the simulation model also form inputs. The intermediate output of the model is the average concentration per sector, the mean speed per sector and the physical position of each macroparticle at each time step. The final output of the model is concentration mean speed per sector, queue lengths and travel times on the links, as a function of time. In this chapter, the development of a

new dynamic simulation/assignment approach, the theoretical aspects of the simulation model, and the general considerations for selecting the simulation parameters are described.

## **3.2 A DYNAMIC TRAFFIC SIMULATION/ASSIGNMENT MODEL**

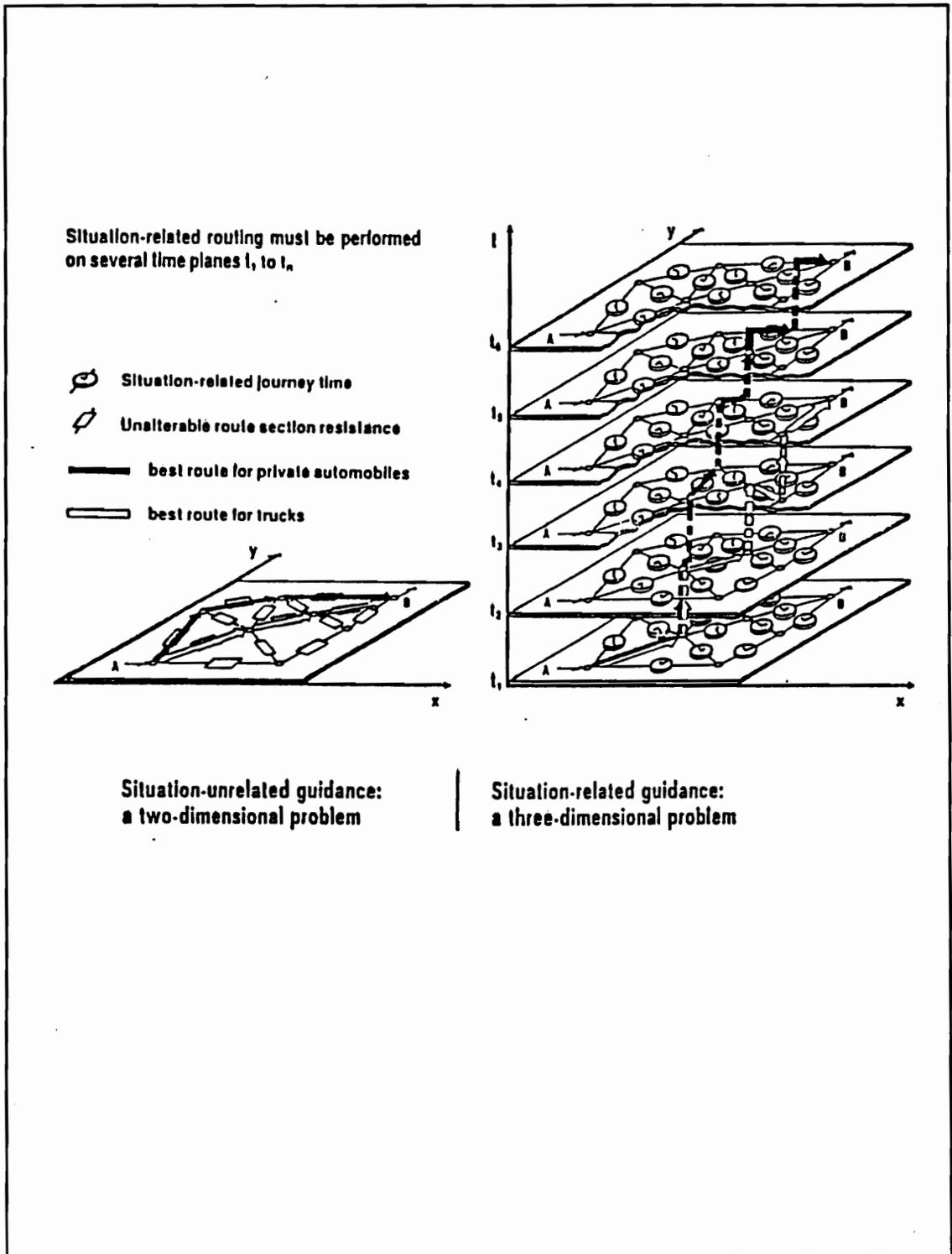
In this section the approach to improve the weaknesses of static traffic assignment methods is carefully investigated. An assignment method that uses simulation to keep track of the vehicles in a network is proposed. The proposed method is expected to give results closer to the real behavior of the time-dependent traffic flow.

### **3.2.1 TRAFFIC ASSIGNMENT IN THREE DIMENSIONS**

The principal drawback of the traditional traffic assignment models is that they assign traffic to a network without a time dimension. In other words they assign vehicles to all the links on the shortest route and assume them to be present at the same time on all links on that route. The remedy to this problem is to give a time dimension to the

assignment process. Figure 3.1 illustrates the differences between static and dynamic routing. To introduce a time dimension, the overall analysis period needs to be divided into sub-periods. For example, if the overall period is sixty minutes, the sub-periods can be fifteen minutes long. During each of these periods, the corresponding fifteen minute O-D matrix is calculated by using the sixty minute O-D matrix, and assigned to the system. These time dependent O-D matrices are generated by dividing the sixty minute O-D matrix into fifteen minute O-D matrices using a specific distribution function.

Simulation of dynamic behavior of the traffic flow in time is the major task of this model. A special purpose simulation tool that can capture the complex dynamics of a traffic system, especially the fluctuation of travel time with departure time and the time dependent congestion patterns is a convenient approach for this purpose. A macroparticle Simulation Model (MPSM) is proposed as the tool [Mahmassani et al., 85]. The simulation enables the system to adopt varying travel times for each link, depending on traffic flow. In addition, a period capacity instead of hourly capacity for each period is used. Thus, the introduction of the time dimension to the model eliminates the drawbacks of the traditional assignment models.



**Figure 3.1: Static versus Dynamic Routing (Koutsopoulos, 90)**

As described in Chapter 2, an assignment process includes the route finding step and the trip loading step. An important assumption made in the dynamic traffic assignment algorithm used in that research is that route choice decisions are made at the time of trip departure. These decisions are based on the travel times which are updated at the beginning of the traffic assignment but travelers are allowed to change their chosen path during any successive periods of assignment if any alternative route has a better travel time than the old route. In addition to these two components, the proposed dynamic traffic assignment model also incorporates simulation component.

### **3.2.2 THE PROPOSED SIMULATION/ASSIGNMENT MODEL**

The model is based on an iterative feedback loop between an assignment and a simulation phase, resulting in the assignment of volumes to the links. The assignment phase loads trips to the network and the simulation phase provides detailed information about the network performance, given its geometric and operational characteristics. The loop terminates when all the trips have been loaded. The method is similar to the incremental assignment method which is used in case of the static traffic assignment. The objective of the

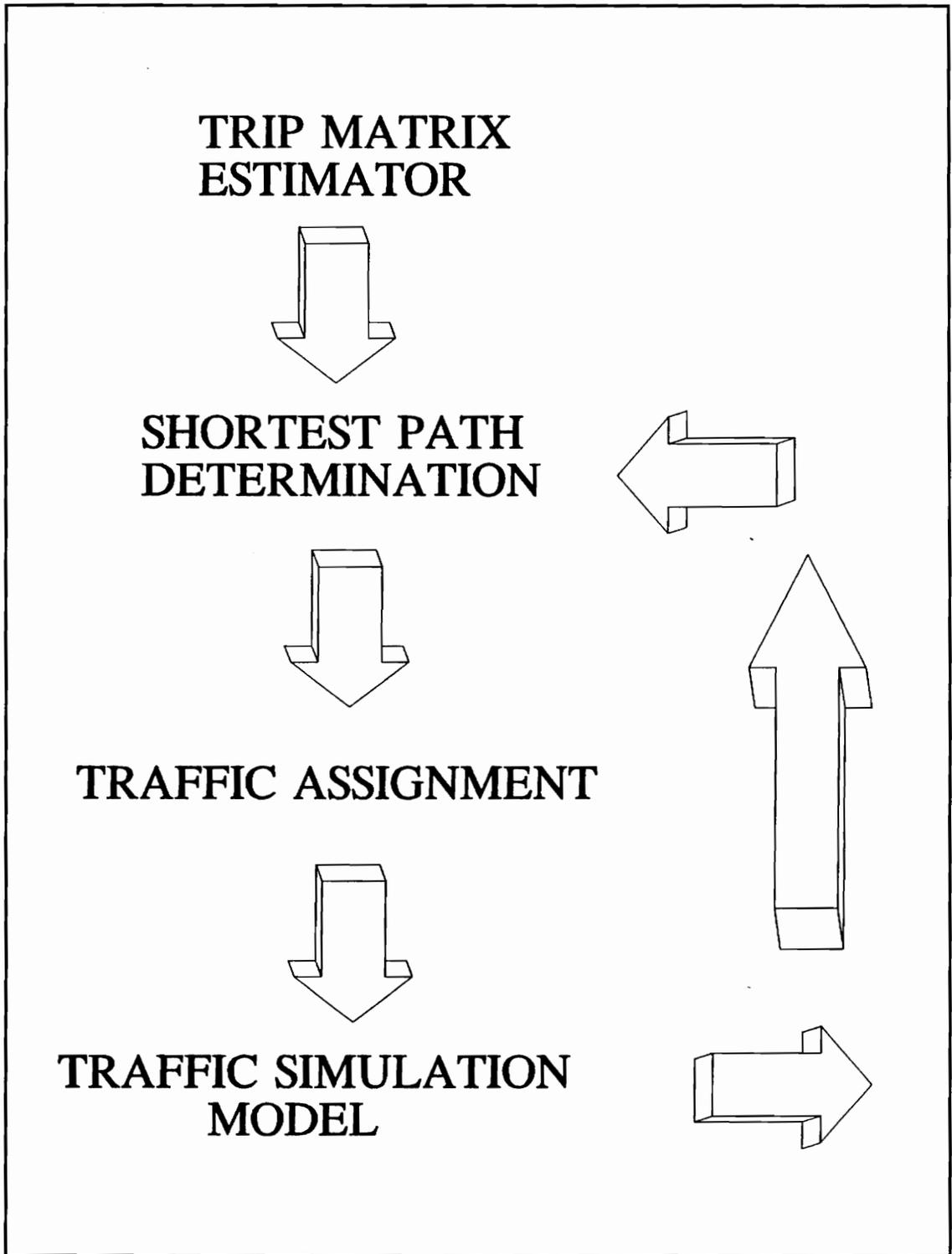
assignment phase is to assign trips of a specific time period to the network bearing in mind the relationship between the travel times and the flows for every link or set of links in the network. To accomplish the assignment under dynamic traffic flow, the simulation phase provides detailed information about varying link travel times. As mentioned above a Macroparticle Simulation model is used.

### 3.2.3 SIMULATION/ASSIGNMENT PROCESS

The model uses a successive process to predict the patterns of link flows and queues as they vary in time. The main mechanism of the model is to import simulated results on the traffic characteristics of the network, such as travel times and link traffic volumes and use them as an input to determine the shortest route for any origin destination pair in the network. Overall logic of the dynamic simulation /assignment model is given in Figure 3.2.

The steps of the model can be described as following:

1. Read network data (free flow travel times, link-node incidence, capacities for each time period). Initialize link volumes for each time interval to zero.
2. Set period  $n=0$ , and iteration number  $i=1$ .



**Figure 3.2: Overall Logic of Simulation/Assignment Model**

3. Set  $n=n+1$ . Determine the O-D matrix for period  $n$ .
4. Group vehicles from each origin-destination pair to form packets. The packets from each origin-destination pair enter the network equally spaced in each time interval.
5. Determine link travel times (link delays) from:
  - a) if it is the first period, from the input data
  - b) else, from the output of the simulation at step 8
6. Process the packet according to its entry time to the network. If the packet enters the network for the first time, determine its shortest path between its origin and destination. Assign the packet to its minimum route, and go to step 8; else if the packet was already on the network go to step 7.
7. If the packet's next link does not have enough capacity, try to divert it to an alternative shortest route; if there is not any available alternative route, leave the packet on its old route.
8. Simulate vehicle movements on the network for the interval  $[t, t+dt]$ , where " $t$ " is the time at any moment and " $dt$ " is the time increment.
9. If the period has ended go to step 10; else go to step 5.
10. If all departure time intervals in the analysis period have been processed, continue; else return to step 3.
11. If all the vehicles did not leave the network go to step

3; else continue.

12. Check for the stopping criteria; if the number of iterations, "i", is smaller than a specified number; else stop. In this research, "i" is defined as one, implying there will be no further iterations.

At this point it is necessary to explain briefly the logic of step 9. The first iteration is unlikely to give the equilibrium distribution of traffic. In order to obtain equilibrium it is necessary to carry out further iterations. By iterating, the model transfers the experience acquired in the previous iteration to the current one. Although it is likely that the model converges within satisfactory limits for a sufficient number of iterations, it is not guaranteed, and the process of iterating is time consuming. Thus, for this research, the process of iterating is not going to be implemented, but it remains a strong alternative to reach an equilibrium. Another important point is that, the algorithm continues until all the vehicles assigned to the network leaves it. Algorithmic steps are given in Figure 3.3.

The most important component of this model is the one that simulates the traffic flow. This simulation part keeps track of the traffic and gives the necessary information about the actual condition of the network. The simulation model is going to be discussed in the next section of this chapter.

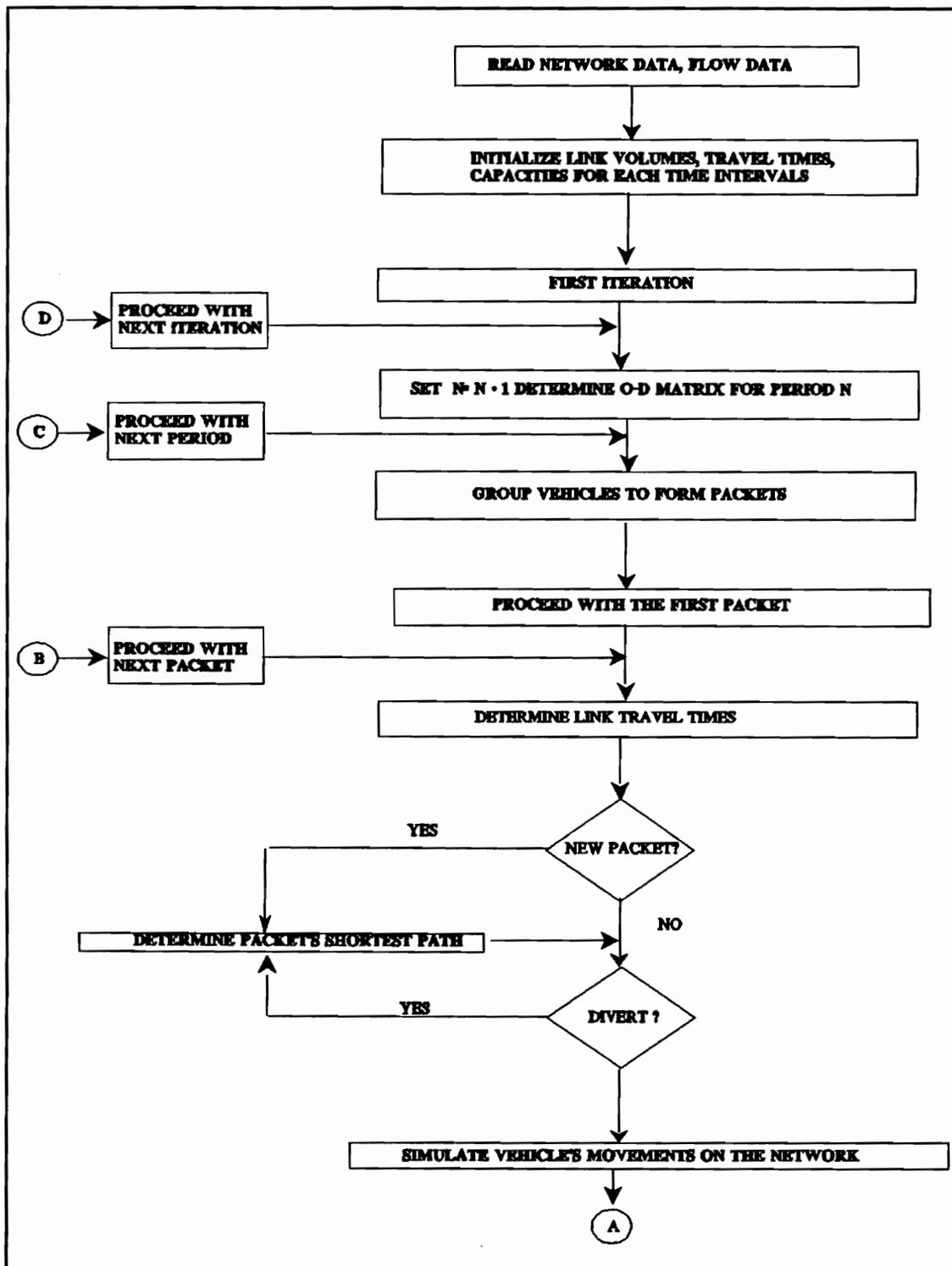
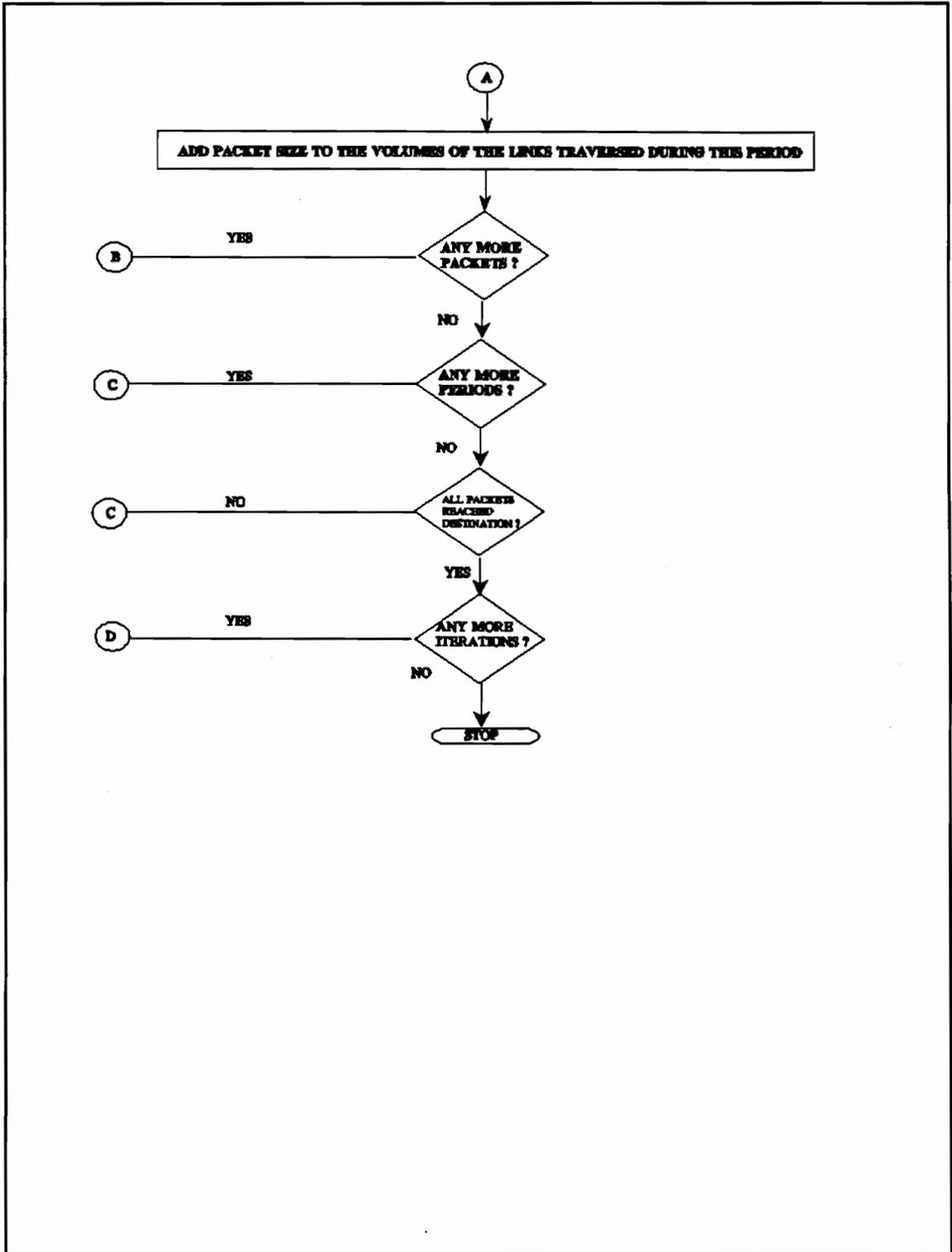


Figure 3.3: Flowchart of Simulation/Assignment Model



**Figure 3.3: Flowchart (Continues)**

### 3.2.4 MACROPARTICLE SIMULATION MODEL (MPSM)

The major concern is with the dynamics and convergence properties of the network system. Therefore the simulation model must possess the capability of providing the following information:

- 1) The concentration fluctuation in each section of the network at a given time for a given time-dependent demand,
- 2) The variation of travel time and queue lengths versus time-dependent demand.

The MPSM is a fixed-time macroscopic traffic simulation model. MPSM was first used in plasma physics [J.N. Leboeuf, 79] Then, [Mahmassani et al., 85] adopted this concept for traffic simulation. It uses well known traffic relationships to simulate the movement and the interaction of vehicles on a network. Most of the macroscopic simulation models like MACK and FREFLO, model traffic flow in an highway as a compressible fluid, but MPSM considers the traffic flow as a collection of vehicle groups, called macroparticles. This is similar to the platoons used in TRANSYT-7F. The model keeps track of the physical position of the particles using pre-specified speed-density relationships. Due to the macroparticle approach, the

significant cost of tracing individual vehicles is avoided. The overall framework of the MPSM is shown in Figure 3.2

There are similarities and differences between most of the commonly used macroparticle simulation models and the MPSM model.

Like the other macroparticle simulation models, MPSM assumes that:

- 1) time is discretized into small equal intervals
- 2) the highway facility is divided into sections or links
- 3) traffic demand and system capacity is fixed over a given time interval

On the other hand, MPSM differs from those macroparticle simulation model in:

- 1) It does not view traffic flow as a compressible fluid like most of the other models, but as a collection of macroparticles.

Two important components of the MPSM model are the vehicle generation component and traffic simulation component.

#### **3.2.4.1 Vehicle Generation Component**

The vehicle generation component of MPSM processes the daily time-dependent vehicle demand patterns. The approach in

this research is to get the demand matrices for a particular time period, from another source and to use an O-D split function to generate demands for each time slice. Following this approach, the model user can specify the departure time distribution as a discretized function giving the number of departures from one origin, in each time interval.

In this approach the O-D matrix is distributed over all the periods according to a specific distribution. This distribution defines the fraction,  $f_p$  of the O-D matrix which departs in period "p". This distribution process can be shown as follows:

$$OD_{o,d,p} = OD_{o,d} * f_p \quad \dots 3.1$$

which is subject to

$$\sum_{p=1}^{P-P_{od}} f_p = 1 \quad \dots 3.2$$

where,

$$OD_{o,d} = \text{Cumulative O-D matrix (1 hour)} \quad \dots 3.3$$

$$P_{od} = T_{od} / T_{per} \quad \dots 3.4$$

$$OD_{o,d,p} = \text{O-D matrix for period p}$$

and  $T_{od}$  is the time-interval for which the traditional O-D matrix is defined. The form of departure distribution

function "f" is subject to further study. The form can be different for morning or evening rush and for different days of the week.

The DYNTRAS model divides the one hour origin-destination matrix into twelve, five minute origin-destination matrices by using two types of distributions: the first has a peak value to represent peak hour distribution, and the second is a constant distribution function.

The five minute O-D matrices will not be affected by the actual conditions of the network. They are assumed to remain unchanged during each period. This means that the travel demand is assumed to be inelastic in order to simplify the complexity of the problem.

#### **3.2.4.2. Traffic Simulation Component**

Having generated five minute O-D matrices, these trips are then assigned to the network. Vehicles are loaded uniformly. They are grouped into packets of fifteen vehicles. The vehicles in a packet are assumed to remain together throughout the entire travel time. A departing packet is identified by its departure time, origin and destination. Each packet is assigned to a shortest route which is determined at the time the packet is loaded to the network.

Thus, the shortest path selection decision is based on the actual conditions of the network. Travel time changes that may occur in the future are not encountered for the shortest path determination process. Once the shortest path of a packet is determined, this packet is assumed to move along this path until it reaches its destination, unless there is congestion. In case of congestion a diversion decision needs to be made. The packet will be diverted to a better route, if feasible.

Naturally, all vehicles cannot enter the facility simultaneously due to its physical and operational constraints. A simple deterministic queuing approximation is used to handle this phenomenon. In addition to that, if number of vehicles that try to enter to a link exceeds the capacity of that link, they are put on a queue list that belongs to that specific link and wait until there is available capacity.

The service rate denoted by "s" is determined using the capacity of each link and the demand at time "t" at the beginning of link "i" is denoted by  $D(t,i)$ . A packet wishing to depart or enter a link in the time interval  $[t, t+\Delta t]$  is considered to be in the queue, if and only if,  $D(t,i) > s * \Delta t$ . In this case,  $\Delta t$  is taken as 1 minute. The service discipline in the queue is first come, first served. Therefore, a packet that enters the queue first will also be

served first, there is no priority defined for the vehicles that wants to depart simultaneously.

The MPSM model, like the other macroscopic traffic simulation models, uses concentration and speed-concentration equations. It does not use the flow relation equation, which is the product of speed and concentration equations. MPSM considers the vehicles as groups, called macroparticles, and moves them in accordance with the local speed on each link, specified by the speed-concentration relation. Thus, the concentration at each link is updated at every time step by tracing the actual position of the particles. The concentration equation in the MPSM has the following form:

$$k_i^t = \frac{(M_i^{e,t} - M_i^{o,t} + N_i^t)}{l_i * \Delta x_i} \quad \dots 3.5$$

where,

$k_i^t$  = concentration in section i during the t-th step, in vehicles per lane-mile

$l_i$  = number of lanes in section i

$\Delta(x_i)$  = length of section i

$M_i^{e,t}$  = vehicles that enter the section from the preceding section for a given time step  $\Delta t$

$M_i^{o,t}$  = vehicles that move onto the next section for a

given time step  $\Delta t$

$N_i^t$  = number of vehicles that remain in section  $i$  at the end of time period " $t-1$ "

Using the above equation, the concentration in each section is updated, at the beginning of every time step, and it is assumed to remain constant over the interval  $[t, t+\Delta t]$ . The corresponding mean speed during this interval can be obtained from the speed-concentration relation which has the following form:

$$v_i^t = (v_f^i - v_0) (1 - k_i^t / k_0)^\alpha + v_0 \quad \dots 3.6$$

where,

$v_i^t$  = mean speed in section  $i$  during the  $t$ -th time step

$v_f^i$  = mean free speed

$v_0$  = minimum speed

$k_0$  = maximum or jam concentration

$\alpha$  = a parameter

Macroparticles are moved according to the mean speed of the prevailing section. As a result of this movement during a particular time step, the distance traveled and the resulting position of the particle at the end of the interval are determined. The section concentrations are then updated for the next time interval. The new concentrations are employed

to estimate mean travel speed on each section. This method keeps track of the particles and is able to produce a more realistic representation of the traffic flow compared to the models that monitor the traffic flow using the following flow equation:

$$q_i^{t+1} = k_i^t \cdot v_i^t \quad \dots 3.7$$

This equation that is used to control the flow from one finite section to another can produce unrealistic results by transporting material in very short times over long distances. As a result of this, nonphysical high travel speeds can be obtained. Therefore particle moving process adopted by the MPSM model constitutes an advanced step for the more realistic representation of the traffic flow [Mahmassani et al. 85].

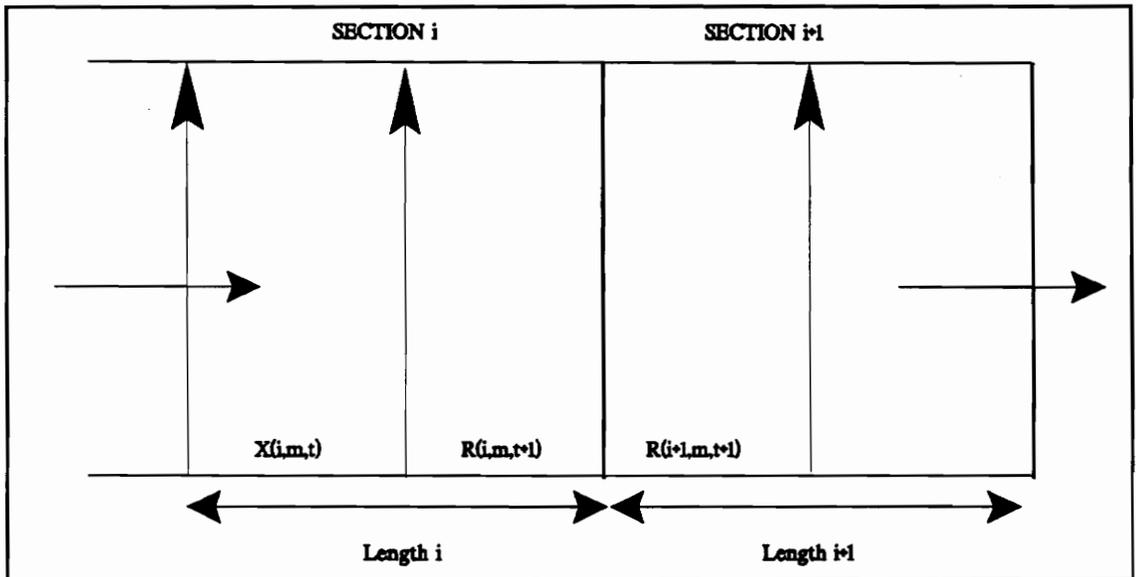
### 3.2.4.3 VEHICLE MOVING PROCESS

One of the attributes of each macroparticle is its physical position on the network. Its position is updated at the end of each time step, then stored as the actual position of that particle. Figure 3.4 illustrates this process.  $X(i,m,t)$  denotes the physical position of a macroparticle "m" at the section "i" at the end time interval "t" and  $R(i,m,t)$  denotes the distance from the macroparticle's current position

to the beginning of the next downstream section which is obtained as follows:

$$R(i,m,t) = \Delta(x_i) - X(i,m,t) \quad \dots 3.8$$

To find the new position of the macroparticle, "m", at the end of the time interval  $t+1$ , the macroparticle is advanced by a distance equal to  $d(m,t+1) = \Delta t * v_i^{t+1}$ , if the distance



**Figure 3.4: Illustration of Vehicle Moving Process**

traveled by the macroparticle is equal to or less than the distance from its current position to the beginning of the next downstream section. In that case, the particle remains in section "i". On the other hand, if the distance traveled by the particle is longer than the remaining portion of the link "i", the particle will have to travel to the next section,  $i+1$ , during a fraction of the time interval  $\Delta t$ . The

above conditions can be summarized in a mathematical form as follows:

If  $d(m,t+1) \leq R(i,m,t)$

then, "m" remains in section i during time interval  $\Delta t$ .

If  $d(m,t+1) > R(i,m,t)$

then, "m" travels to the next section "i+1" during time interval  $\Delta t$ , where it travels during a fraction of time interval  $\Delta t$ .

In the second case, the travel time  $\Delta t$  should be broken into two parts; the first part will represent the time spent on the link "i" and second part will represent the time spent on the link i+1. During the first part, distance denoted by  $R(i,m,t)$  has been traveled by the macroparticle at the mean speed  $v_i^{t+1}$ . Therefore the first fraction of  $\Delta t$  can be determined as:

$$\Delta t_1 = R(i,m,t) / v_i^{t+1} \quad \dots 3.9$$

The remaining part the time interval which can be denoted as  $\Delta t_2$  is then equal to:

$$\Delta t_2 = \Delta t - (R(i,m,t) / v_i^{t+1}) \quad \dots 3.10$$

To determine the distance traveled on the section i+1 during  $\Delta t_2$ , the speed-time relationship can be used again. But it is not realistic to assume that drivers will adjust

their travel times as soon as they enter the section  $i+1$ . Therefore, it is assumed that drivers will adjust their travel speed while entering the section " $i+1$ " in a manner that their speed is consistent with the traffic conditions in both sections; this is achieved by taking the average of the speeds of section " $i$ " and " $i+1$ ". Then, the distance of the macroparticle from the beginning of the link " $i+1$ ", denoted by  $R'(i+1,m,t+1)$  is determined as:

$$R'(i+1,m,t+1) = \Delta t^2 * \left[ .5 * (v_i^{t+1} + v_{i+1}^{t+1}) \right] \quad \dots 3.11$$

Thus, using the vehicle moving process, traffic is moved on the network and the position of each packet is determined.

### 3.2.5 TRAVEL TIME ESTIMATION

Travel time estimation is based on the actual conditions of the network. Travel times are updated for each link at the end of every simulation step. These travel times are assumed to be fixed for the next simulation step. They are found at the end of each simulation step by adding the time to cross the link, and queue waiting time. The former is calculated using the current link speed and latter using the rate of queue clearance over a few previous time steps.

Thus for link "i" at time "t" the travel time is found by using the equation:

$$t_i = t_{iq} + t_{im} = (Q_t T / N_T) + (L_i / V_i^t) \quad \dots 3.12$$

Where,

$t_{iq}$  = current queue wait time

$t_{im}$  = current movement time

T = Min(five minutes, length of time with queue)

$N_T$  = number of vehicles cleared in time T

$Q_t$  = current queue length, in vehicles

$L_i$  = length of link

$V_i^t$  = speed on link i at time t

As the equation above states, current movement time is found by dividing the length of the link by the speed of that link.  $T/N_T$  gives the rate of queue clearance, and is used to estimate the time needed for the clearance of the existing queue,  $Q_t$ .

### 3.2.6 DIVERSION AND DRIVER BEHAVIOR

As mentioned before, the model does not have the capability of predicting the future conditions on the network. At the beginning of a packet's trip, its shortest route is

determined based on the actual network conditions. It is clear that as time passes, other packets will enter the network and will concurrently use some of the links of this packet's shortest route. This may cause congestion on those links and the shortest route determined in the beginning of the packet's route may end up not being the shortest route anymore. However, the model does predict this occurrence. Since this research is closely related to the IVHS, the idea is that driver will have a device in the car which will give him/her information about the travel time on the shortest route from the current node to his/her destination. A user is assumed to switch his/her current path to the best alternative route, only if the improvement on the remaining trip time exceeds an indifference band. The indifference band of trip time saving can be viewed as a threshold that can be expressed relative to the remaining trip time. Therefore, a user will switch, if the trip time saving is higher than a threshold value set by the model. A switching rule with a relative indifference band subject to a minimum trip time saving, can be stated as:

$$\delta_i(k) \begin{cases} 1 & \text{if } TC_i(k) - TB_i(k) > \eta_i \\ 0 & \text{otherwise} \end{cases} \quad \dots 3.13$$

Where,

$\partial_i(k)$  = 1, indicates a route switching; 0, no switch

$TC_i(k)$  = Trip time from node k to CBD on current path

$TB_i(k)$  = Trip time on the best alternate

$\eta_i$  = Indifference threshold;

The model can be improved by introducing an indifference threshold fraction that changes for every link on the network, according to their position relative to the destination, and which is distributed among the drivers with a statistical distribution. This distribution may be triangular or normal. But for the sake of simplicity a fixed threshold value for all the drivers and for all links is considered. The value of this threshold is choose arbitrarily. Further studies are needed for the determination of realistic threshold values that reflect real driver behavior. This requires extensive work of calibrating and testing the model using real data.

The model also has the option of modelling different fractions of drivers receiving information. Drivers who have access to the information will be able to switch, while the others assigned to specific routes will remain on those till they reach their destinations. This option will enable the model users to understand the effects of different levels of information availability, and to determine the system

performance and the optimum market penetrations under different network conditions, and loading patterns.

### 3.3 MODEL PARAMETERS

In this section, some general considerations for selecting the key simulation parameters of MPSM are discussed. The parameter values used in equation 3.6 are obtained from the well known traffic engineering theory, and they are  $v_f = 45$  mph/60 mph,  $v_0 = 6$  mph,  $k_j = 180$  vehicles per lane-mile,  $\alpha = 1$ . There are three key simulation parameters in the MPSM: simulation time step,  $\Delta t$ , macroparticle size, "m", and section length,  $\Delta x$ . Determination of these parameters are based on previous results obtained from tests performed using similar macroparticle simulation models [Mahmassani et al., 85]

#### 3.3.1 SIMULATION TIME STEP, $\Delta t$

Traffic performance characteristics are assumed as constant during each simulation step,  $\Delta t$ . Very long simulation steps will give unrealistic results. It is clear that smaller simulation time steps will increase the accuracy of the simulation. Table 3.1 shows the sensitivity of the MPSM model to the length of  $\Delta t$  [Mahmassani et al., 85]. There

is a big computation cost difference between two  $\Delta t$  values which are 0.5 and 1.0. In order not to achieve more accuracy, the simulation step size is assumed to be equal to 1.0.

**Table 3.1: Effect of Simulation Time Step ( $\Delta t$ )**

$\Delta t$ (min)	CPU <sup>a</sup>	CPU <sup>b</sup>
0.5	5.32	14.105
1.0	2.98	8.042
1.5	2.16	5.975
2.0	1.82	4.684
3.0	1.43	3.873

Note:  $\Delta m = 10$  vehicles/particle, 400 vehicles/sector, two lanes in each sector,  $v_f = 45$  mph,  $v_0 = 6$  mph,  $\alpha = \pi$ , and simulation duration = 60 min.

<sup>a</sup>No intermediate report is included in the program output.

<sup>b</sup>Intermediate report of system performance and all vehicle positions during each simulation interval is included in the output.

### 3.3.2 SECTION LENGTH, $\Delta x$

In most macroscopic simulation models, the continuous highway facility is divided into discrete sectors. The system characteristics such as concentration and speed in a sector are assumed to remain constant within the sector, during each simulation time interval. Although a shorter sector length can give more realistic results, the increase of computation costs due to the increase of the number of sectors need to be

considered. Previous tests performed to determine the relationship between computational cost and number of sectors in case of MPSM shows that the incremental cost of increasing the number sectors is rather small relative to the high efficiency of MPSM [Mahmassani et al. 85]. This is shown in table 3.2.

**Table 3.2: Effect of Section Length ( $\Delta x$ ) on Computational Efforts**

No. Sectors <sup>a</sup>	$\Delta X$ (mile)	CPU <sup>b</sup>	CPU <sup>c</sup>
21	0.3	3.56	9.304
13	0.5	3.25	8.468
7	1.0	2.98	8.042
5	1.5	2.54	6.89
4	2.0	2.01	5.77

Note: In all experiments,  $\Delta t = 1$  min,  $\Delta m = 10$  vehicles,  $V_f = 45$  mph,  $V_0 = 6$  mph,  $\alpha = \pi$ , and simulation period = 60 min.

<sup>a</sup>The last 1 mile of this 7-mile facility was treated as one sector in all the cases reported here.

<sup>b</sup>No intermediate report is included in the program output.

<sup>c</sup>Intermediate report of system performance and all vehicle positions during each simulation interval is included in the output.

Therefore, for this research, sector lengths are chosen in such a way that they do not exceed 1 mile and that the geometric features remain the same within each sector.

### 3.3.3 MACROPARTICLE SIZE, $m$

Vehicles that are moving to the same destination are grouped to form macroparticles. Although macroparticles that

have variable number of vehicles would not cause any theoretical problems, for programming convenience, macroparticles are assumed to be identically sized. It is obvious that bigger particle sizes will increase the computational efficiency by decreasing the total number of particles. However vehicles in the same packet are assumed to remain together and travel together. Therefore very big particle sizes will not be logical. For example, if the number of cars in a packet is more than the service rate for a given simulation step, the packet will not be allowed into the link. Hence "m" should be less than the maximum service rate. Thus, a macroparticle size of fifteen cars has been considered convenient for this simulation model.

## 4.0 TECHNICAL ASPECTS OF THE MODEL

### 4.1 DESCRIPTION OF THE COMPUTER CODE

This chapter discusses the principal components of the DYNTRAS computer simulation/assignment model, and provides detailed information about the model's capabilities.

DYNTRAS program has two main modules: static and dynamic traffic assignment modules. The reason for including the former is to interpret and discuss the results of the latter. The results obtained from the static assignment model will clarify the difference between static and dynamic traffic assignment methods. The computer code can be classified into three main sections, as follows:

- 1) General file operation and network data handling routines. These routines are common to both methods.
- 2) Static traffic assignment routines. Those are mainly routines related to convex combination method.
- 3) Dynamic traffic assignment routines. These routines mainly comprise of those performing the simulation process. Other routines in this category update the time and vehicle movements, collect statistics about the network status at different time periods, and form output files.

The computer package is designed to be menu driven, which will make the program user friendly. The code has been developed in QUICK BASIC language.

## 4.2 MODEL STRUCTURE

The first module of the model uses a convex combination algorithm. Steps of the algorithm are explained in detail in chapter two. The second module uses a simulation/assignment model to perform dynamic traffic assignment. Basically, this is a fractional simulation model. Traffic demand is divided into different time intervals using a function explained in chapter two. The trips in one interval are then loaded onto the network according to their arrival time to the network.

### 4.2.1 MODEL INPUT AND OUTPUT

The basic input to the DYNTRAS program includes two major groups:

- 1) Highway network topology and trip demand patterns (necessary to both modules, Static and Dynamic). These include network geometry, nodes, links, link distance, speed, existing volume and capacity of each link.

2) Simulation characteristics (only for dynamic traffic assignment module). These include total simulation time, simulation periods, simulation step size.

The output of the DYNTRAS model is designed to suit the specific needs of the research. It can be summarized as:

1) Report of the input data.

2) Congested links and network characteristics, including link concentration, link volume, link travel times, and queue lengths are provided for each simulation time step and each fifteen minute period. These values are considered as network performance indices.

3) Packet data: Each packet's route, total travel time from the origin to the destination, and diversion decisions made by the packet.

#### **4.2.2 DATA REPRESENTATION**

Fifteen cars are assumed to form a packet. Each packet carries a set of information, which is updated for each simulation step and written to memory to minimize the use of random access memory. Like the packet information, link data is also represented as a block. Some of this information is static, such as link length, link capacity, and link nodes; but the other part, such as link volumes, link concentration,

link speed are time varying due to the traffic flow. On the other hand, link capacity for different periods can be changed by the user, in order to simulate some temporary link closures, like accidents.

The data is represented in form of blocks, enabling the programmer to define the attributes of a major element as a complete body. This kind of data representation facilitates the maintenance and modification of the overall computer code.

## **4.3 GENERAL FILE OPERATIONS AND NETWORK DATA HANDLING ROUTINES**

In this section, two types of routines are explained: routines that handle general file operations and those that manage input and output network data.

### **4.3.1 GENERAL FILE OPERATIONS ROUTINES**

The following routines handle network data files.

**OPENRANFIL:** This subroutine opens random data files that store network data such as node numbers, link lengths, initial travel costs, O-D travel matrix.

**CLOSEFILE:** This subroutine closes random data files.

INPNETDAT: This subroutine is used to input and save general network data needed for the computation.

GETODTABLE: This subroutine is designed to get the origin-destination table from the user

INDATA: It calls subroutine GETODTABLE to get O-D table from the keyboard and saves the data, if so specified by the user.

SAVEDATA: This subroutine asks the user whether or not he wants to save the entered data. If the user wants to save it, a command is sent to the subroutine INDATA to do so.

PRTNETDAT: This subroutine prints network data input by the user.

## **4.3.2 DATA HANDLING ROUTINES**

These routines are designed to transform the input data into an efficient structure so that computation times of other routines using this data are reduced.

### **4.3.2.1 DATA MANAGEMENT SUBROUTINE (NETDATMAN)**

This algorithm accepts a link data item and stores it in an arc list ordered by an ascending origin node [Jensen, 80]. This list is very useful to the label-correcting algorithm in determining the minimum travel-time paths from an origin node

to all other nodes in a network. This subroutine sorts the links in such a way that all links emanating from the same node are stored adjacently avoiding repeated searches of the link list. In order to save computer memory space, each link in this list is represented only by its end node.

A pointer indicates the position of links beginning with the node for which it is kept. Therefore, the network can be stored with one link length and one node length arrays, instead of keeping two link length arrays, which will consume more memory.

#### 4.3.2.2 LINK DETERMINATION SUBROUTINE (NODLINDET)

This algorithm determines the list of links (and their terminal nodes) originating from the same node I. The steps involved in this algorithm are listed below.

1. Find pointers to start and end of list of arcs originating at node I.
2. If there are no such links, return. Otherwise, put links originating at I in the link list. Find the terminal node of each link, and put it in the node list.

## 4.4 SHORTEST PATH SUBROUTINE (DIJSHOPATH2)

The algorithm used is called label-correcting algorithm [Sheffi 85]. It finds the shortest paths from a given origin node to all other nodes in the network. The algorithm scans the network nodes in an iterative manner in order to find a path from the root. At each iteration a shorter path than the previous path is sought. Iterations continue until no better path can be found from the root to any other nodes in the network. The algorithm requires an examination of all the network nodes at least once. The algorithm uses a list called sequence list, including all the nodes that will be examined, as well as the nodes requiring further examination. This list helps to manage and keep track of the nodes. Another list called the list of predecessor nodes is continuously updated. Once the algorithm stops, the minimum paths can be traced backward, from every node to the root by using this predecessor list.

The criteria used to stop the algorithm is the check of the sequence list. If the sequence list is empty, the algorithm stops. Figure 4.1 shows the flow chart for this algorithm.

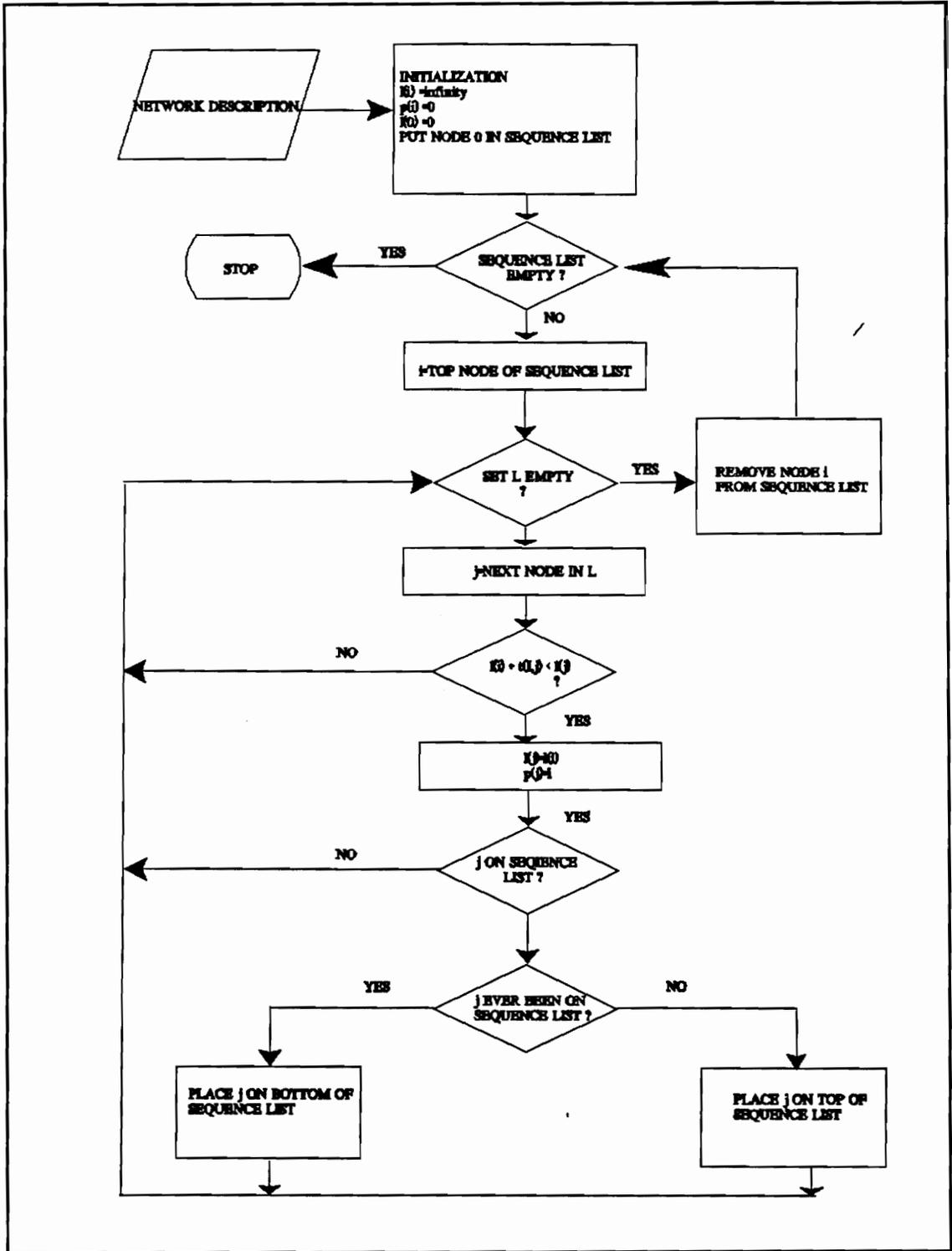


Figure 4.1: Flowchart of Shortest Path Algorithm

## 4.5 USER EQUILIBRIUM MODULE

This module performs static traffic assignment and uses a convex combination method. The main purpose of this module is to obtain static user equilibrium results in order to understand major differences between the dynamic and static traffic assignments.

### 4.5.1 CONVEX COMBINATION METHOD SUBROUTINE (CONCOMNET)

This subroutine achieves user equilibrium flows. Figure 4.2 shows a flowchart of convex combination method. Its algorithmic steps explained in detail in chapter two are implemented in the computer code as follows:

Step 0: This step initialize all the flows on the links to zero. Then it calls subroutine UPDATE to calculate the travel times with zero flows. Given these travel times, it calls subroutine ALLORNOT to perform all-or-nothing assignment for all origin-destination pairs.

Step 1: It calls subroutine UPDATE to update link travel times using new link volumes obtained from the previous all-or-nothing assignment.

Step 2: It calls again subroutine ALLORNOT to assign the trips of all origin-destination pairs to the

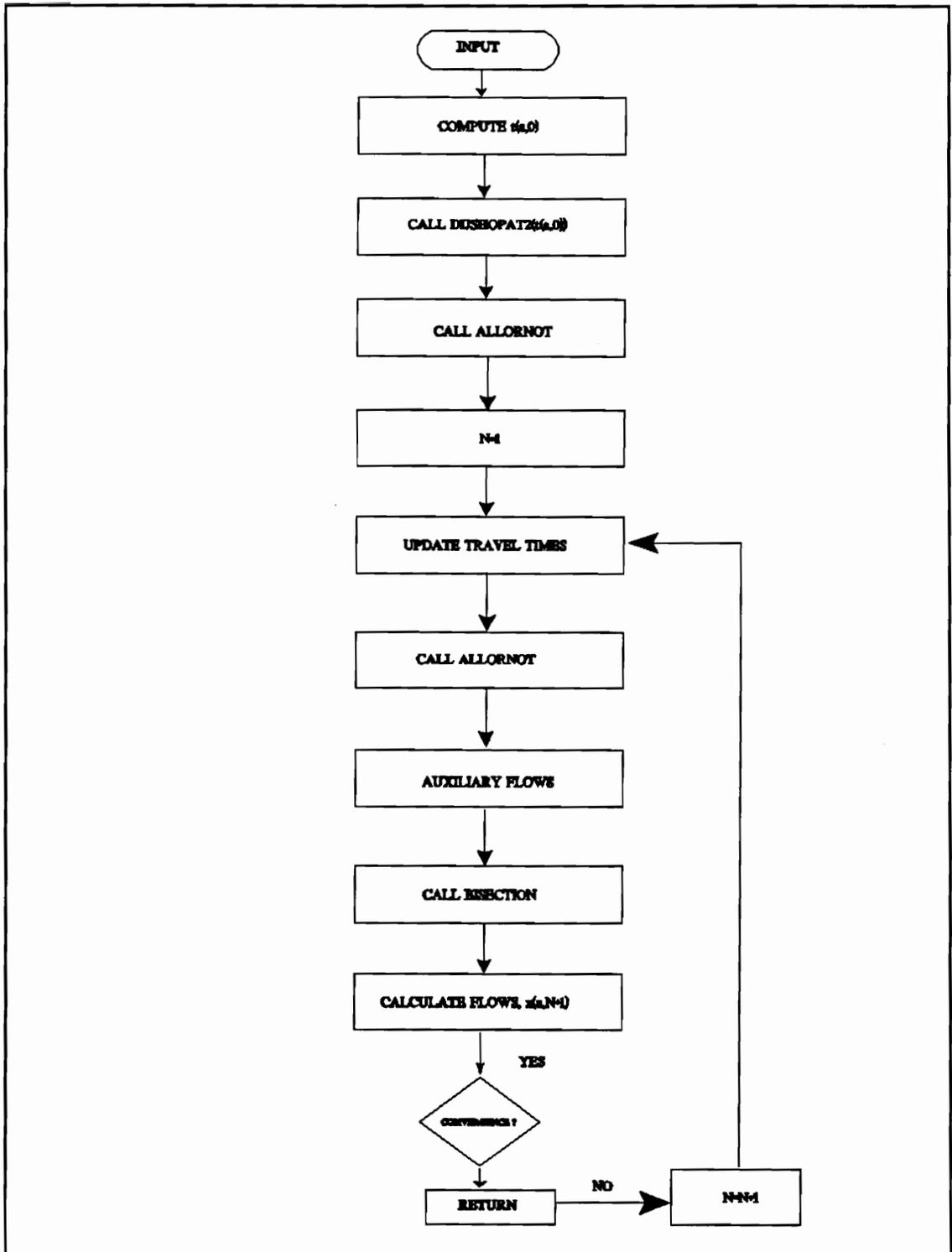
corresponding links.

Step 3: It calls subroutine BISECT which calculates the move size "gama" that minimizes the objective function. "Gama" is used in updating the new volumes that are going to be assigned to links.

Step 4: It checks if the convergence criterion is satisfied. If the change in flows of two successive iterations are less than an infinitesimal value, the equilibrium is achieved, and the program stops. Otherwise, it goes back to step two, and repeats the same steps until the convergence criterion is satisfied.

#### 4.5.2 SUBROUTINE BISECT

The line search to find the optimal move size is performed by using the bisection method. The bisection method requires the evaluation of the derivative of the objective function in every iteration, and in the case of traffic assignment, the derivative of the objective function to be minimized can be evaluated without major difficulty. The existence of the derivative implies that the function is monotonic on each side of the minimum. In other words, for the function to be minimized,  $dz(x)/dx$ , is negative for  $x < x^{\min}$  and positive for  $x > x^{\min}$ .



**Figure 4.2: Convex Combination Method (CONCOMNET)**

The algorithm computes the derivative of  $z(x)$  at the midpoint of the current interval  $[a^n, b^n]$ . This midpoint can be denoted as  $x^n$ . If  $dz(x^n)/dx < 0$ , then  $x^n < x^{min}$ , meaning that the next interval will be  $[x^n, b^n]$ . If  $dz(x^n)/dx > 0$ , then  $x^n > x^{min}$ , meaning that the line search will be performed in the interval  $[a^n, x^n]$ .

The procedure terminates when the convergence criterion is met.

### 4.5.3 SUBROUTINE UPDATE

This subroutine is used to update travel times on links every time link volumes changes. It uses the well known BPR curve which has the following form:

$$t_i = t_i^0 [1 + a(V_i / C_i)^b]$$

where,

- $t_i$  = travel time of link  $i$
- $t_i^0$  = free travel time of link  $i$
- $V_i$  = volume of link  $i$
- $C_i$  = capacity of link  $i$
- $a$  = a parameter, 0.15
- $b$  = a parameter, 4

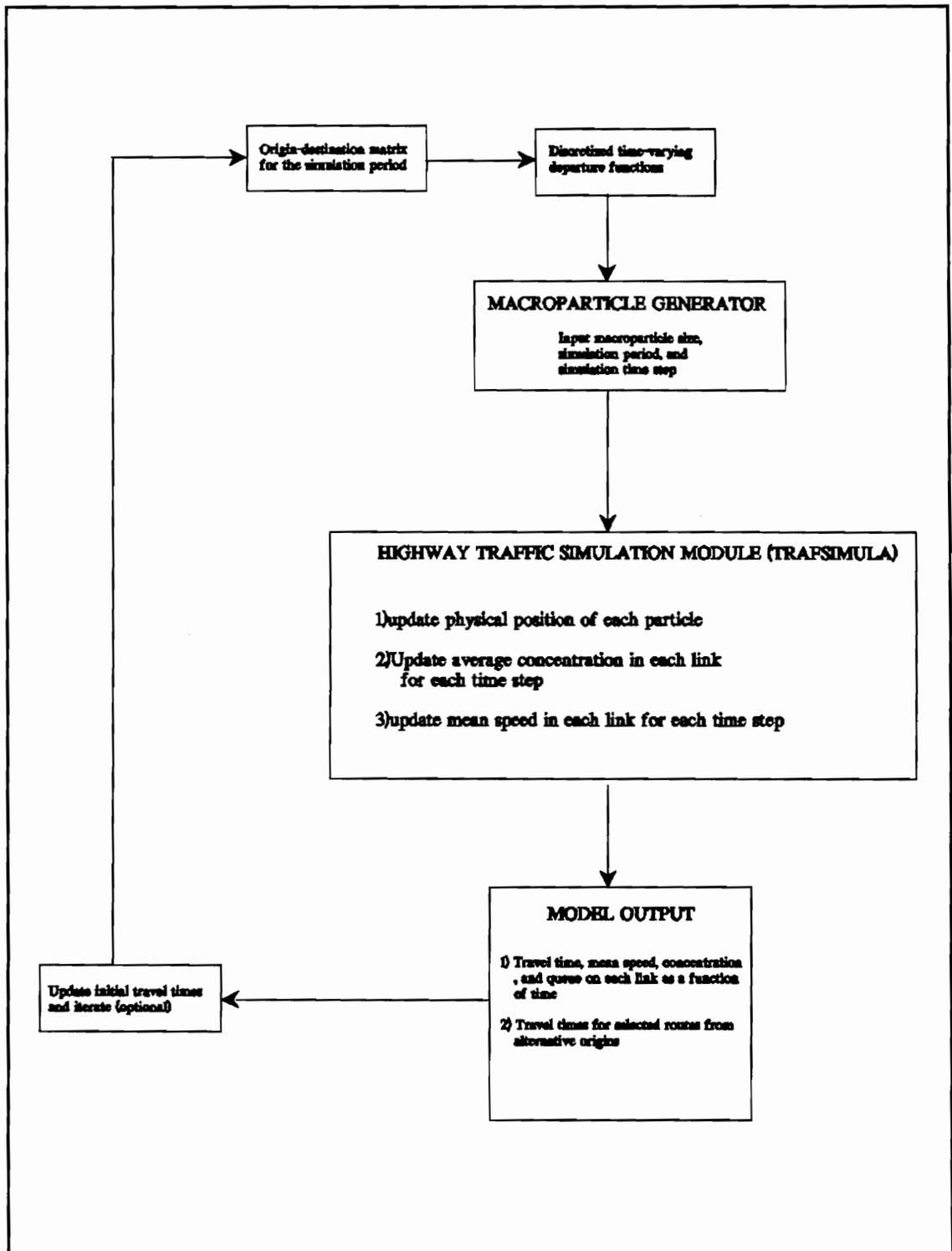
#### **4.5.4 SUBROUTINE ALLORNOT**

This subroutine simply performs an all-or-nothing assignment. In the all-or-nothing procedure, the O-D flow is assigned to every link that is on the minimum travel time path connecting origin "o" to destination "d". All other paths connecting this O-D pair are not assigned to any flow. During this process, the link travel times are assumed to be fixed at values calculated before the assignment.

This subroutine calls subroutine DIJSHOPAT2 , as many times as the number of origins, to identify the shortest path for each O-D pair. Then it allocates the flow for each O-D pair to the links comprising the shortest path for that O-D pair.

### **4.6 DYNAMIC TRAFFIC ASSIGNMENT MODULE**

This module performs dynamic traffic assignment which has two major functions. It simulates the traffic flow on the network, and updates the performance based on the results of the simulation. Then, it determines the shortest routes for each origin destination pair and assigns new arrivals to these routes. The framework of this module is shown in Figure 4.3.



**Figure 4.3: Framework of Simulation Module**

The simulation is carried out by using a set of subroutines that control and perform the simulation process simultaneously . The whole program is written in QBASIC, naturally the simulation part of the code also is written in QBASIC. Use of high level programming language creates some extra complications.

In general, a simulation model is implemented by using a simulation programming language such as SLAM, SIMSCRIPT, SIMULA, SIMAN. However, in this case a high-level language, QBASIC, is used for the implementation of the simulation model, due to the need for integrating the simulation model with other parts of the code that have already been developed in QBASIC.

Programming a simulation model in a high level language is a difficult task. The logic represented in the simulation program is concurrent in nature. For example, in simulating a traffic intersection many simultaneous activities such as arrivals of vehicles from different lanes, light changes, and movements of the vehicles through the intersection, must be represented. On the other hand, a computer without a parallel processing capability is a sequential device and it executes its instructions sequentially. Hence, processing the logic in a sequential manner, but preserving the concurrency of the

activities in the system being simulated, is what makes simulation in a high level programming language difficult [Balci, 91].

To implement the simulation model using QBASIC, a conceptual framework (CF), which is the structure of concepts and views under which the simulation model is developed, must be chosen. In implementing a simulation model in a high level language, the use of a conceptual framework facilitates the implementation and reduces its complexity. The "event scheduling" conceptual framework is adopted in this research. The initialization step includes the assignment of initial values to all attributes, such as link travel times, speeds, and the initialization of the event list at the beginning of the simulation. It is assumed that the network is empty when the simulation starts, and the event list is initialized by the first possible events. After generating the first possible events, they are merged into the event list with respect to their occurrence time. The occurrence time, in other words inter-arrival times for all the events, is assumed to be uniform. Each event representing a packet formed of fifteen cars, has six attributes: occurrence time, packet type, packet origin, packet arrival time, packet number, and packet queue type. Packet type is needed to identify whether an event is already in the network, waiting to enter the

network, or if it already left. Packet arrival time is needed to find out the total travel time of the packet in the network. Packet number identifies the packet and it is given when the packet is first scheduled. Packet queue type identifies if a packet is on the queue or not.

A fixed time increment is used and the next event to be executed is determined, sent to the merge list, and executed when the simulation clock is equal to its occurrence time. After the execution of some events, simulation termination condition is tested. All packages should leave the system before the simulation stops.

If this condition is satisfied, the simulation ends, and the program prints the simulation output, or stops depending on the user specification.

In the next sections, subroutines used for the implementation of "event scheduling" simulation are explained.

#### **4.6.1 Traffic Simulation Subroutine (TRAFSIMULA)**

This subroutine is the main subroutine that controls the simulation process. The steps of TRAFSIMULA are shown in figure 4.4. The detailed sequence of steps followed in the subroutine is described below:

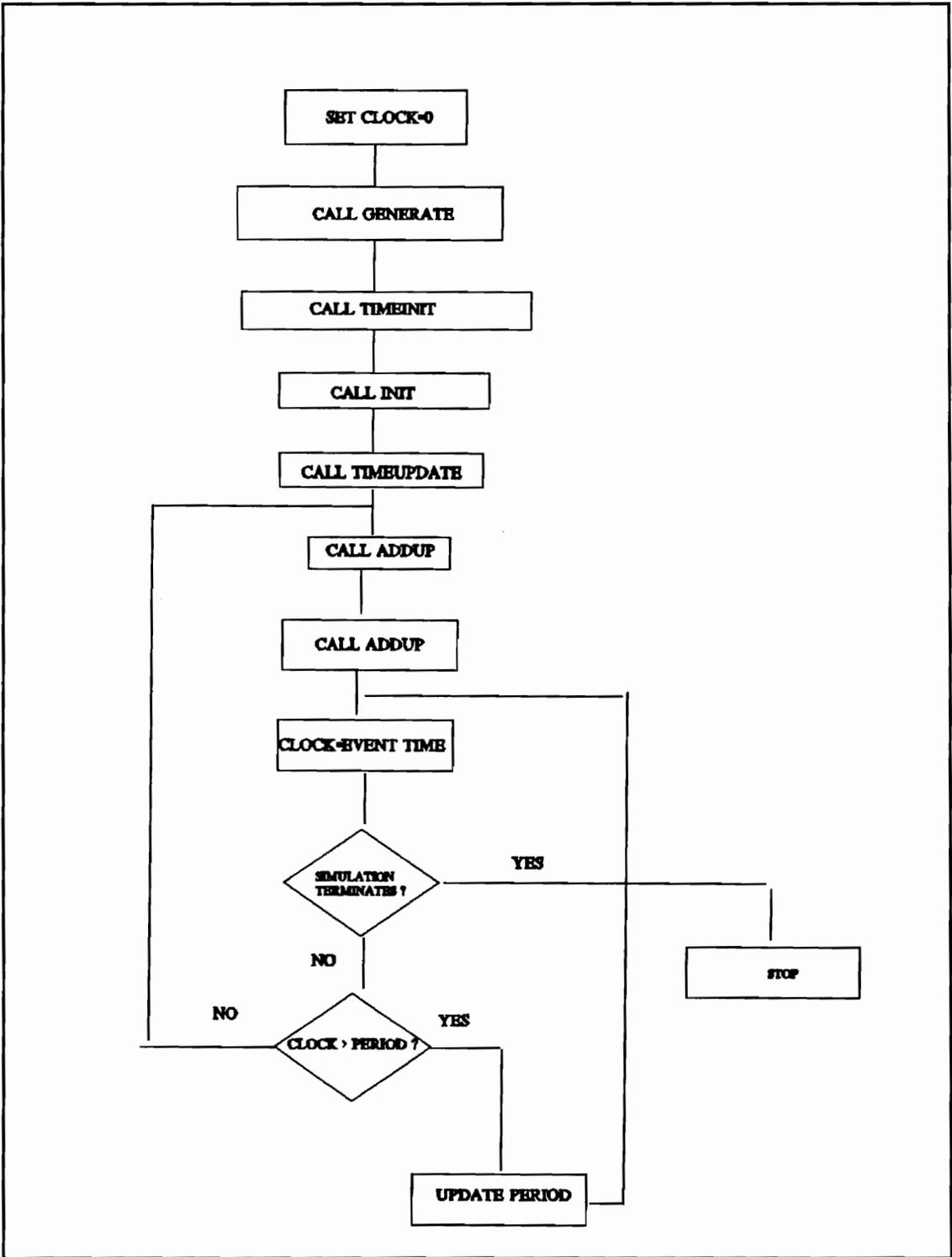


Figure 4.4: Traffic Simulation Subroutine (TRAFSIMULA)

- 1) It splits one hour O-D matrix into four fifteen minutes O-D matrices. At the start of simulation, when the simulation clock is zero, it calls subroutine GENERATE that generates first possible events.
- 2) It calls subroutines TIMEINIT, and INIT to initialize simulation clock, link travel times, speeds, and capacities. The event list and the attributes in the event list are also initialized by sending created events and their corresponding attributes to the event list.
- 3) Then it starts the simulation process for the first period. At the beginning of each period, it calls subroutine TIMEUPDATE to update the travel times for the actual period used in travel times obtained at the previous period, it also calls subroutine ADDUP to initialize link usage parameters.
- 4) It calls subroutine ARRIVE that schedules new events or processes old ones that are already on the network.
- 5) This process goes on till all the events, packets of cars, that are assigned to the network reach their destinations. Therefore the stop criterion for the simulation is the number of departures from the system.

## 4.6.2 SUBROUTINE MERGE

It creates and sorts an event list. The list is sorted according to the event occurrence times. The event that is going to occur first is the first event of the event list. When a "new event" or an "old event" with an updated occurrence time is sent to the event list, the event list is updated to insert the event into the list. In the event list five attributes for each event are defined. These are:

- 1) event(x,1)=event occurrence time.
- 2) event(x,2)=event type; if event is a "new event" event(x,2) is equal to "1"; else, if it is an "old event", event(x,2) is equal to "2", and if event leaves the network, event(x,2) is "0".
- 3) event(x,3)=event origin.
- 4) event(x,4)=event arrival time (total travel time of each packet).
- 5) event(x,5)=event number (number given to each packet for identification purposes).
- 6) event(x,6)=event queue type; if an event is in the queue event(x,6) is equal to "1"; else, it is equal to "0".

### 4.6.3 SUBROUTINE GENERATE

This subroutine is called only at the beginning of the simulation. It first determines the number of packets for each period and for each O-D pair by dividing the number of vehicles to the macroparticle size  $\Delta m$ . In some cases the total number of vehicles is not divisible to  $\Delta m$ , then it adds the remaining of the division to the last packet's volume. In that case the size of the last packet becomes bigger than the other packets. However, this is not going to effect the results of the simulation due to the macroscopic nature of the simulation. For the example runs presented in this research, the number of cars are chosen in such a way that this problem is circumvented.

This subroutine determines the inter-arrival times of the packets. Inter-arrival times are assumed to be uniform and determined for each O-D pair separately. Inter-arrival time for each origin destination pair is calculated by simply dividing the length of one period by the number of packets belonging to that specific O-D pair.

Another function of the subroutine is to generate the first elements to be processed at the beginning of the simulation. Then it calls subroutine MERGE and REGINFO to put them into the event list, and to write the information

regarding these elements to a data file. Once the simulation starts, subroutine GENERATE is no longer called. All the operations related to the generation of new events are taken care of by the subroutine ARRIVE.

#### 4.6.4 SUBROUTINE ARRIVE

This subroutine is the one that takes care of the event arrivals and departures. It either calls subroutine GENERATE to create new event, or processes old events that have already been generated. A flowchart of ARRIVE is shown in figure 4.5. The steps of the subroutine ARRIVE are:

- 1) If the first element of the event list, i.e the event that is going to be processed is a "new event" that waits to be assigned to the network, the subroutine schedules the "next event". It sends this new event to the event list where it is going to be inserted into the event list according to its occurrence time and it calls subroutine REGINFO to register its attributes to a file. After the scheduling of the "next event" is completed, it starts processing the "new event". ARRIVE determines the shortest path for this "new event". This event will move along this route for the rest of its journey to its destination unless some conditions that initiate

diversion are satisfied. It then sends the event to subroutine TRAFMOVE where it will be moved on the network along its shortest route during the simulation step. When the event returns from TRAFMOVE, its attributes are changed, it is not a "new element" anymore. Therefore, its event type and event occurrence time are updated. It is then sent to the event list, because it should be inserted into the event list with respect to its new occurrence time. After it has been merged, the first event of the event list, which is the same event with the old occurrence time, is deleted from the event list.

- 2) If the first element of the event list is an "old event", that has already been processed during the previous simulation steps, it is directly sent to the subroutine TRAFMOVE, because it is already traveling on the network and its shortest path has previously been determined. When it returns from subroutine TRAFMOVE, it is processed in a similar manner as explained in the first step.
- 3) On the other hand, if the event has reached its destination, it is deleted from the event list by subroutine DELETE and a message of "event number ... will be deleted from the event list" appears on the screen.

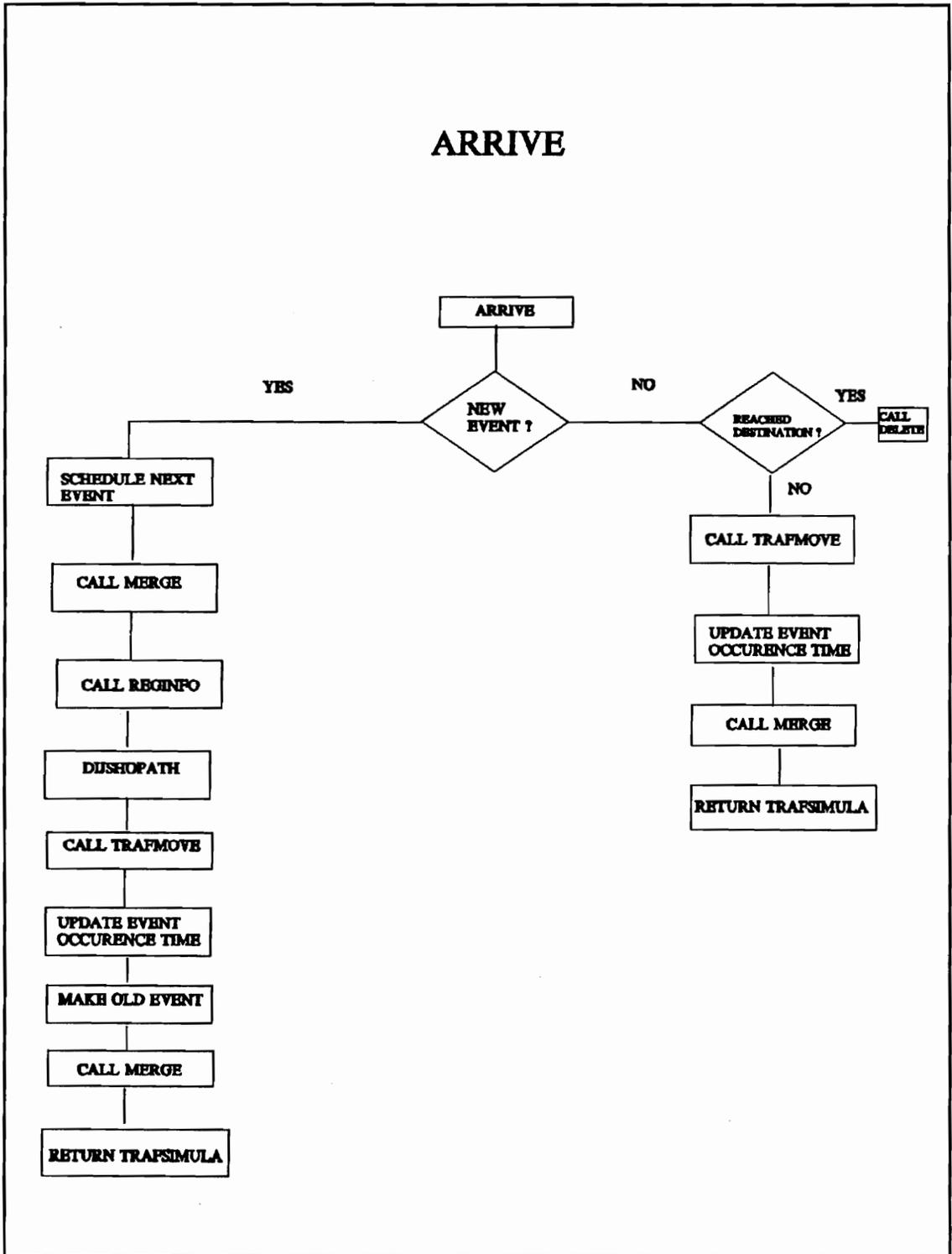


Figure 4.5: Flowchart of Subroutine ARRIVE

#### 4.6.5 SUBROUTINE TRAFMOVE

This subroutine is the one that moves the vehicles on the network according to the vehicle moving process defined previously in this chapter. In addition to moving the vehicles, if the next link on the shortest route of the vehicle does not have enough capacity, it calls the subroutine DIVERT to find out if there is another available shortest route for the packet. A flowchart of TRAFMOVE is shown in figure 4.6. The description of the steps of the subroutine is as follows:

- 1) It updates link concentration, travel time, and speed at the end of each simulation step. These values remain constant during the next simulation step. It also calls subroutine FSTATISTICS for storing them in a data file.
- 2) It moves each packet along its shortest path which is determined in subroutine ARRIVE, and determines the packet's exact position at the end of each simulation step. Then, at the next step, the packet will continue to move towards its destination from that position.
- 3) If there is queue at the entrance of the link, and the packet has access to the diversion information it calls the subroutine DIVERT, and if an alternative route to divert the package is available, it diverts it to that

route by changing its shortest route, and moving it along this shortest route for the rest of its trip. This is the subroutine RADIO that determines whether or not information is available to a packet. On the other hand, if the packet cannot get information or there is not an alternative route, it puts the packet in the queue.

- 4) If the packet reaches its destination, it changes the packet type to "0" ,which means that it has reached its destination, and the packet is deleted from the event list.

#### **4.6.6 SUBROUTINE DIVERT**

This subroutine checks for the best alternative route to divert the packet. If there is an acceptable alternative route, it lets the packet to divert. The search for an alternative route is done by using the following steps:

- 1) If there is a queue at the entrance of one link, and there is available capacity at the other links connected to the origin node of this link, it initiates the search for a possible diversion.
- 2) It considers the point where the vehicle stopped as the intermediate origin, and it looks for other routes from that intermediate origin to the packet's destination.

# TRAFMOVE

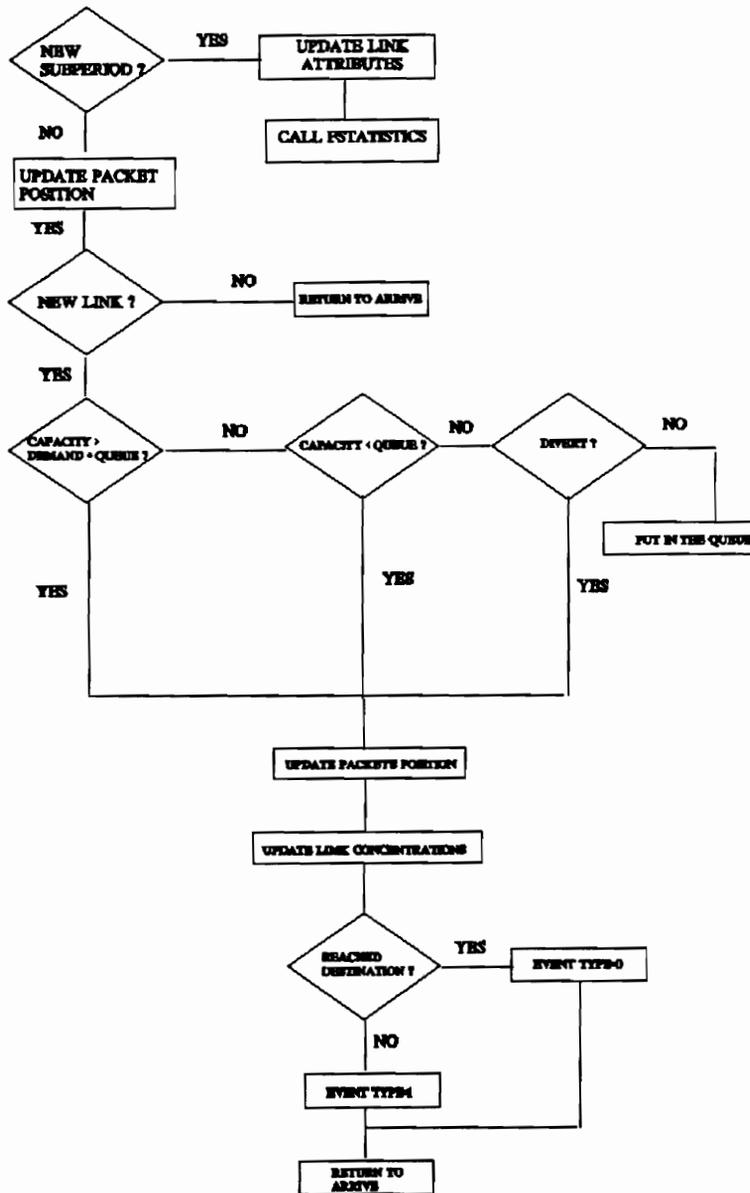


Figure 4.6: Flowchart of Subroutine Trafmove

- 3) It then calculates the remaining travel time on the packet's original route and denotes it as  $TT_r$ . It selects the new route as an alternative if the total travel time on this new route is smaller than  $(1+IB)*TT_r$ . The multiplication by "1+IB" comes from the concept of indifference band explained in Chapter 3. IB can get any value, for example, "IB" equal to ".1" means that the new route is suitable only if it is ten percent shorter than the old one.
- 4) If there is a route that satisfies these conditions, the packet's shortest route for the rest of its trip is updated. The remaining part of its trip will be along this new route.
- 5) If an alternative route does not exist, the packet is not diverted, and it stays in the queue till the next time it is going to be processed.

#### **4.6.7 SUBROUTINE DELETE**

This subroutine is used to delete events from the event list. It deletes a processed event, and an event that has reached its destination.

#### **4.6.8 SUBROUTINE FSTATISTICS**

This subroutine is used to store link travel times, volumes, concentrations, speeds at each period.

#### **4.6.9 SUBROUTINE OUTPUT**

This subroutine gets the data file that stores simulation statistics, and prints them. It also asks the user whether he or she wants to keep track of a specific package. If the user wants to do so, enters the number of the package, and the subroutine displays its position, and speed at every minute.

#### **4.6.10 SUBROUTINE RADIO**

This subroutine determines whether a packet has access to the diversion information. The user specifies the fraction of packets that has the capability of getting the information, and subroutine RADIO randomly determines the packets that can access to that information. It uses a built-in random number generator function of Quick Basic. This feature allows the user to test the effects of different fractions of information availability on the congestion and travel times.

## 5.0 MODEL RESULTS AND ANALYSIS

### 5.1 INTRODUCTION

The DYNTRAS simulation/assignment model described in the previous chapters was applied under different scenarios of information availability, trip demand, O-D matrix split functions, and indifference band values, on a relatively small test network. The same network was used to run user equilibrium assignment model.

### 5.2 ANALYSIS DESCRIPTION

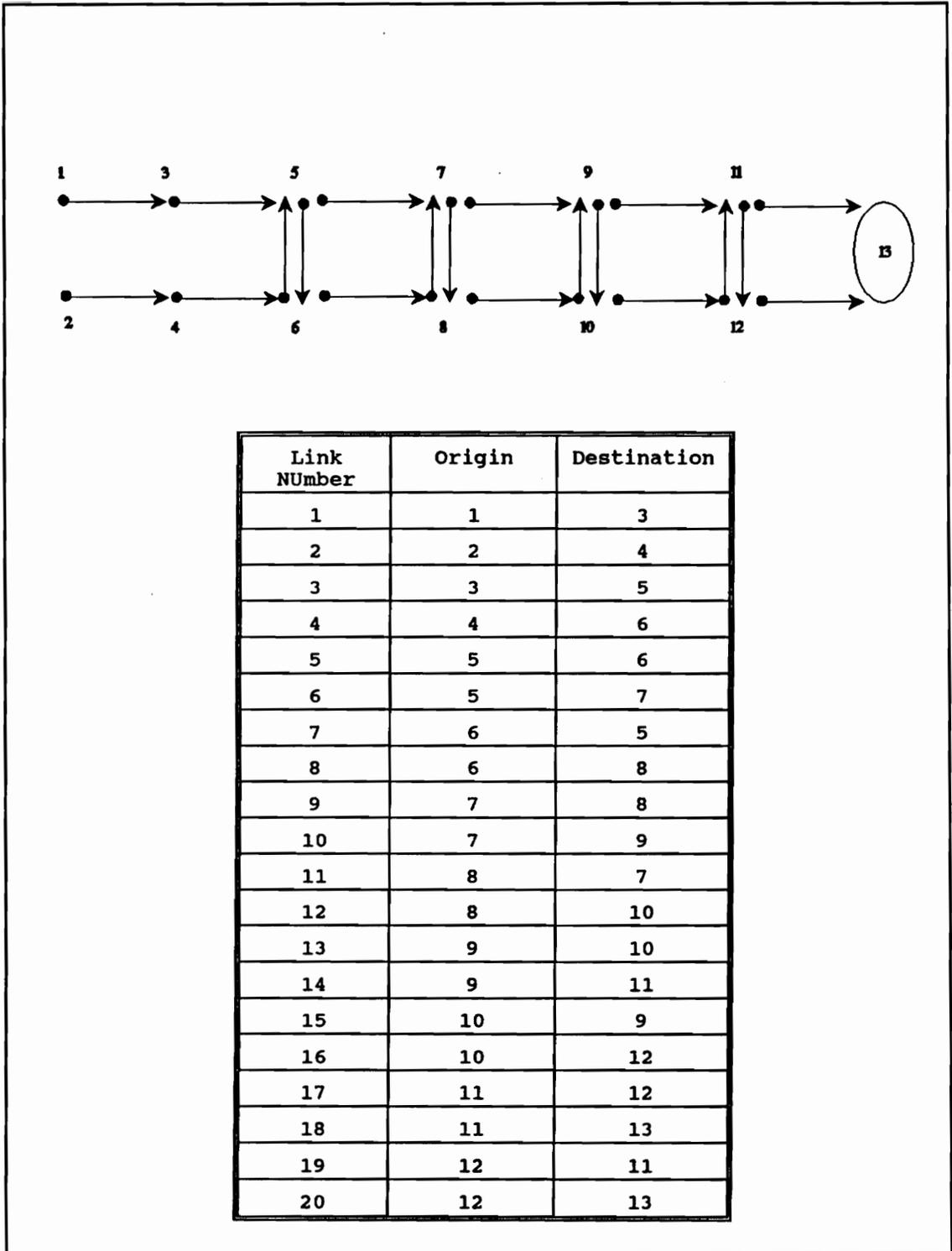
The simulation experiments and user equilibrium assignment are performed for a network that consists of thirteen nodes, and twenty links of which four are connectors between two highways. Both highways are six miles long and each is divided into six one mile segments. Four crossover links at the end of second, third, fourth, and fifth miles allow switching from one highway to another (see Figure 5.1). Highway-1 has a free speed of 48 miles per hour and Highway-2 has a free speed of 60 miles per hour. All cross over links

have a free mean speed of 55 mile per hour. The practical capacity assigned to links was set at 3600 cars per hour for Highway-1 and cross over links, and 4000 for Highway-2.

Cars enter the network from two origins and travel to a single destination. This destination can be thought as central business district or a major industrial area. In all simulation runs, there was a total of 6300 trips taking place between origin and destination pairs of the network. These 6300 trips were loaded to the network over a period of forty-five minutes. A start-up period of fifteen minutes was used to fill the network, and the statistics of all the vehicles that entered the network during the thirty minutes following the start-up period were accumulated. Simulation continued until all the cars loaded onto the network left it.

Simulation experiments were conducted to examine the opportunities for system improvement with respect to four factors, namely; information availability, loading pattern, O-D split function which were also referred as O-D distribution functions in Chapter 3, and indifference band value (behavioral factor). In all cases, inter-arrival times were assumed to be uniform.

In addition to the simulation runs, the user equilibrium module was used to obtain the static traffic assignment results. The user equilibrium assignment was performed for



**Figure 5.1: Test Network**

two different O-D matrices which are used as loading patterns in simulation experiments. The aim of the user equilibrium assignments is to gain insight to the problem of traffic assignment by analyzing the results of static and dynamic methods.

### **5.3 EXPERIMENTAL FACTORS**

Four major factors used to test the sensitivity of the simulation/assignment model were:

- 1) Information availability: Four different levels of the fraction of users with real-time were considered (100%, 60%, 40%, 0%). Packets that could get information were randomly determined according to the above fractions.
- 2) Loading patterns: This is one of the major factors that effects the existing traffic network conditions. To capture this effect, two loading patterns were adopted. First loading pattern has departing rates of 4800 cars per hour for Highway-1 and 3600 cars per hour for Highway-2. Under the second loading pattern, 3600 cars per hour enter Highway-1 and 4800 cars per hour enter Highway-2. The first pattern and the second patterns were selected to test the network performance when one of the highways is congested.

3) O-D Matrix split (distribution) functions: 45 minute O-D matrices were split into 5 minute O-D matrices using a split function described in Chapter three. Two types of split functions were used:

- a) A parabolic function (Type-1).
- b) A constant function that divides the 45 minute O-D matrix in 9 equal 5 minute O-D matrices (Type-2).

These functions determine departure time choices. Under the first case, the departure patterns vary in time. Number of departures are lower in the beginning, they reach a maximum in the middle of the assignment period, and then drop to a lower value again. On the other hand, the second function splits the commuter departures uniformly, assuming that the departure pattern does not change in time.

4) Indifference band: In the simulation runs, each user with information has access to the diversion information. Two different indifference band values are considered (0.0 and 0.1). Under the first indifference band value, the user with information will always switch to another path, if it offers an improvement in trip time, no matter how small. However, the second indifference band value indicates that switching is allowed if the travel time on the alternative routes is at least 10 percent shorter than the initial one.

## 5.4 ANALYSIS OF RESULTS

A base scenario was developed. Several factors were varied from this scenario to test the sensitivity of the model. The one minute data obtained from the model was aggregated into a five minute data by taking the average of five consecutive minutes to simplify the analysis procedure. This did not cause big accuracy losses due to the fact that drastic changes were not observed within five minute periods. System performance for each simulation run was evaluated by using the average travel time on the selected routes, and average travel time on the system. Link travel times are very good measures of system performance, because they are estimated by using both concentrations on links and queue build-ups. The notion of selected routes was introduced in order to be able to calculate average travel time on the system. Due to the dynamic nature of the simulation/assignment process, the routes followed by the vehicles are not fixed. Therefore, eight different routes were selected to test the model sensitivity to different simulation parameters, and to determine the average travel time on the system. These routes were selected among the ones that had the highest concentrations during the overall simulation period. Selected routes for the analysis of the

simulation results are shown in Table 5.1.

**Table 5.1: Selected Routes**

SELECTED ROUTE NAMES	LINK NUMBERS
FREEWAY1 (FW1)	1-3-6-10-14-18
FREEWAY2 (FW2)	2-4-8-12-16-20
ALTERNATIVE1 (A1)	1-3-5-8-12-16-20
ALTERNATIVE2 (A2)	2-4-7-6-10-14-18
ALTERNATIVE3 (A3)	1-3-6-9-12-16-20
ALTERNATIVE4 (A4)	2-4-8-11-10-14-18
ALTERNATIVE5 (A5)	1-3-6-10-13-16-20
ALTERNATIVE6 (A6)	2-4-8-12-15-14-18

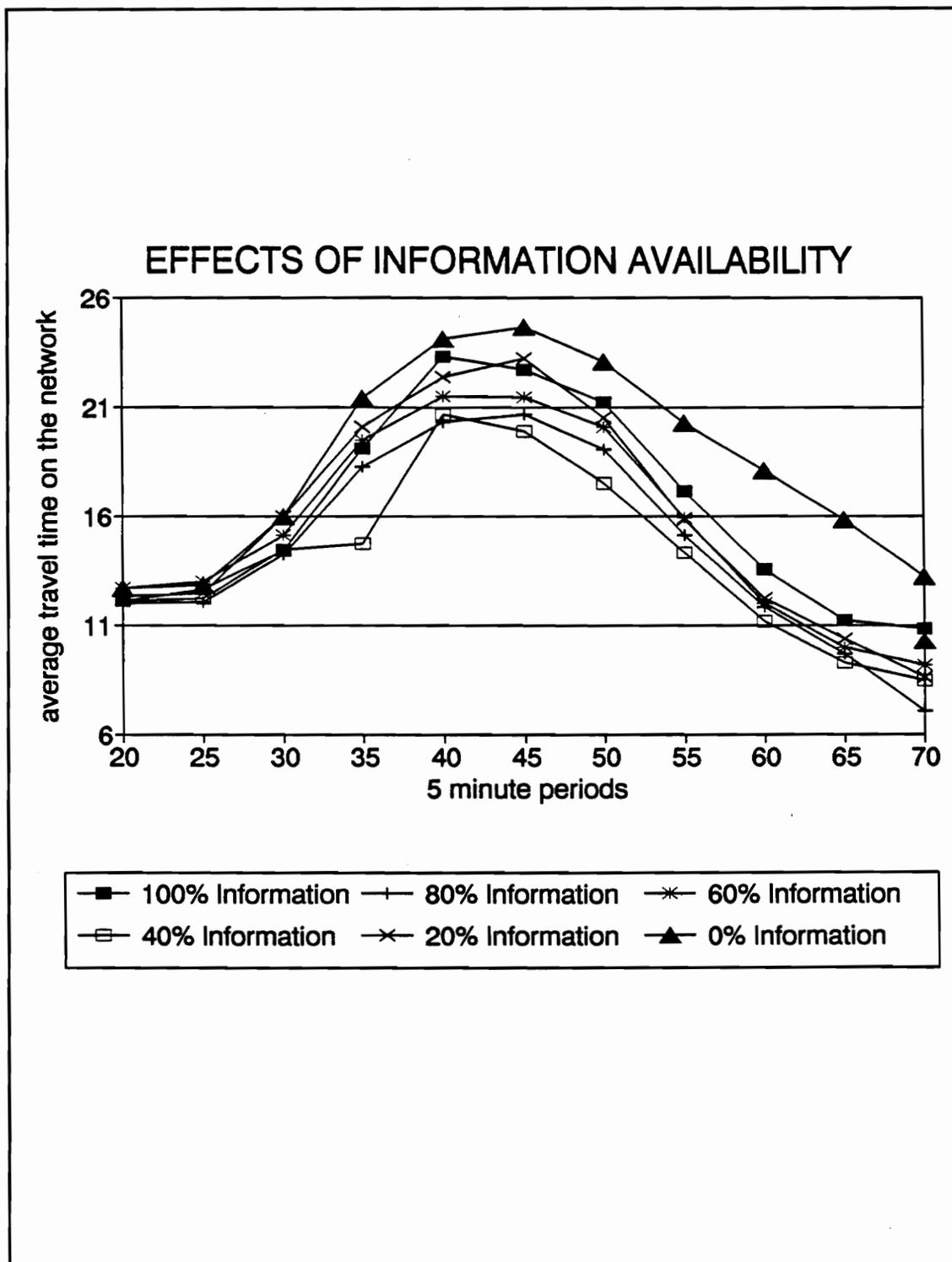
#### 5.4.1 SCENARIO-1

This scenario was developed by using the first loading pattern where 4800 cars per hour depart from the first origin and 3600 cars per hour depart from the second origin. The indifference band was assumed to be zero and type-1 O-D matrix split function was used. Six different fractions of users who are able to receive information were modeled (100%, 80%, 60%,

40%, 20%, 0%).

Figure 5.2 shows the variation in average travel time in the system under different percent information availability. It is clear that the travel time variation pattern is similar in each case. Travel time increases with increasing volumes, and start to decrease after the forty fifth minute where loading process stops. The best results, shortest average travel times, were obtained for 40 percent information case. The worst results were obtained for no information case. This shows that the diversion information helps to optimize users travel times. On the other hand, the second worst results were obtained for 100 percent information availability. This indicates that the extent of improvement in travel times depends on the level of information availability. This increase is due to the fact that everybody with access to information diverts to better routes, causing a congestion in these routes and in turn, an increase in travel time. There might be an optimal amount of information availability, and to determine this amount of information for different traffic conditions further investigations are needed.

Scenario-1 with 100 percent information availability has similar features with static assignment. In static assignment, it is assumed that users have perfect knowledge of the system (100% information ), and they change their routes if a better



**Figure 5.2: Scenario-1**

alternative route exists (Indifference band=0).

**Table 5.2: Travel Times for User Equilibrium**

---

ROUTE NO	LINKS	TRAVEL TIME
1	1-3-6-10-14-18	9.474
2	1-3-5-8-12-16-20	9.483
3	2-4-8-12-16-20	7.39

---

On the other hand, static traffic assignment tries to optimize the system, whereas simulation/ assignment tries to simulate the network conditions. That's the main reason why it is not possible to compare the results obtained from these methods, but they can be analyzed to understand major differences between these two approaches.

Table 5.2 shows travel times on three routes in case of user equilibrium for the same O-D matrix. Table 5.3 shows the change of average travel times for selected routes. An average travel time value for each route is also shown in this table. In table 5.3, the variation of travel times within each five minutes, for different routes departing from the same origin are always smaller than one minute. These variations are

**Table 5.3: Travel Times on Selected Routes**

ROUTE	20	25	30	35	40	45	50	55	60	65	70	AVERAGE
FM1	13.71	14.58	17.4	24.38	27.96	30.00	26.43	21.21	15.13	11.71	10.93	15.24
FM2	9.53	9.79	10.39	12.34	15.38	14.30	14.21	11.24	10.76	9.79	9.91	9.11
A1	14.73	15.21	18.22	24.61	29.34	29.62	30.93	23.80	16.18	11.19	10.73	16.04
A2	10.17	10.46	11.12	14.86	16.15	16.71	14.52	13.78	12.25	11.80	11.20	10.43
A3	13.88	14.74	17.9	24.93	28.86	30.45	27.17	20.50	15.52	11.28	10.99	15.43
A4	10.58	10.91	11.44	13.54	15.84	15.38	15.05	13.04	11.64	11.40	10.95	10.19
A5	14.57	15.23	17.52	24.69	28.30	30.93	27.03	20.81	15.21	11.65	11.15	15.56
A6	9.84	10.37	11.54	13.47	16.36	14.77	14.83	12.78	11.89	11.00	10.70	9.82

even smaller for optimal information availability cases (40%-60%). Therefore, it can be concluded that the system is close to an equilibrium state in most of the time. Although due to the dynamic nature of the simulation, this equilibrium state is not guaranteed for all of the cases, the incremental nature of the algorithm assignment helps the system to produce reasonably close average travel times for different alternative routes departing from same origins.

On the other hand, there are radical differences between dynamic and static assignment results. The first important difference is the magnitudes of the travel times. In case of user equilibrium, travel times are lower compared to the dynamic model. Two reasons for this difference can be cited: a) different travel time estimation methods were used; BPR curves in case of static assignment and travel time equation 3.12 which was explained in Chapter three, in case of dynamic assignment, b) user equilibrium optimizes the travel times whereas dynamic model does not have an optimization mechanism.

In Figure 5.3, travel times on highways are plotted against time for both assignment methods. This emphasizes even more clearly the difference between these two methods. Other important factors that can be considered are congestion levels and number of links used in each method, and they are

**Table 5.4: Total Number of Used and Congested Links**

T	20		25		30		35		40		45		50		55		60		65		70	
	U <sub>1</sub>	C <sup>2</sup>	U	C	U	C	U	C	U	C	U	C	U	C	U	C	U	C	U	C	U	C
ST.	13	6	13	6	13	6	13	6	13	6	13	6	13	6	13	6	13	6	13	6	13	6
DY.	18	2	19	2	20	2	20	3	20	4	20	5	19	5	15	4	15	5	16	2	5	0

1) U: Number of links with current traffic

2) C: Number of congested links

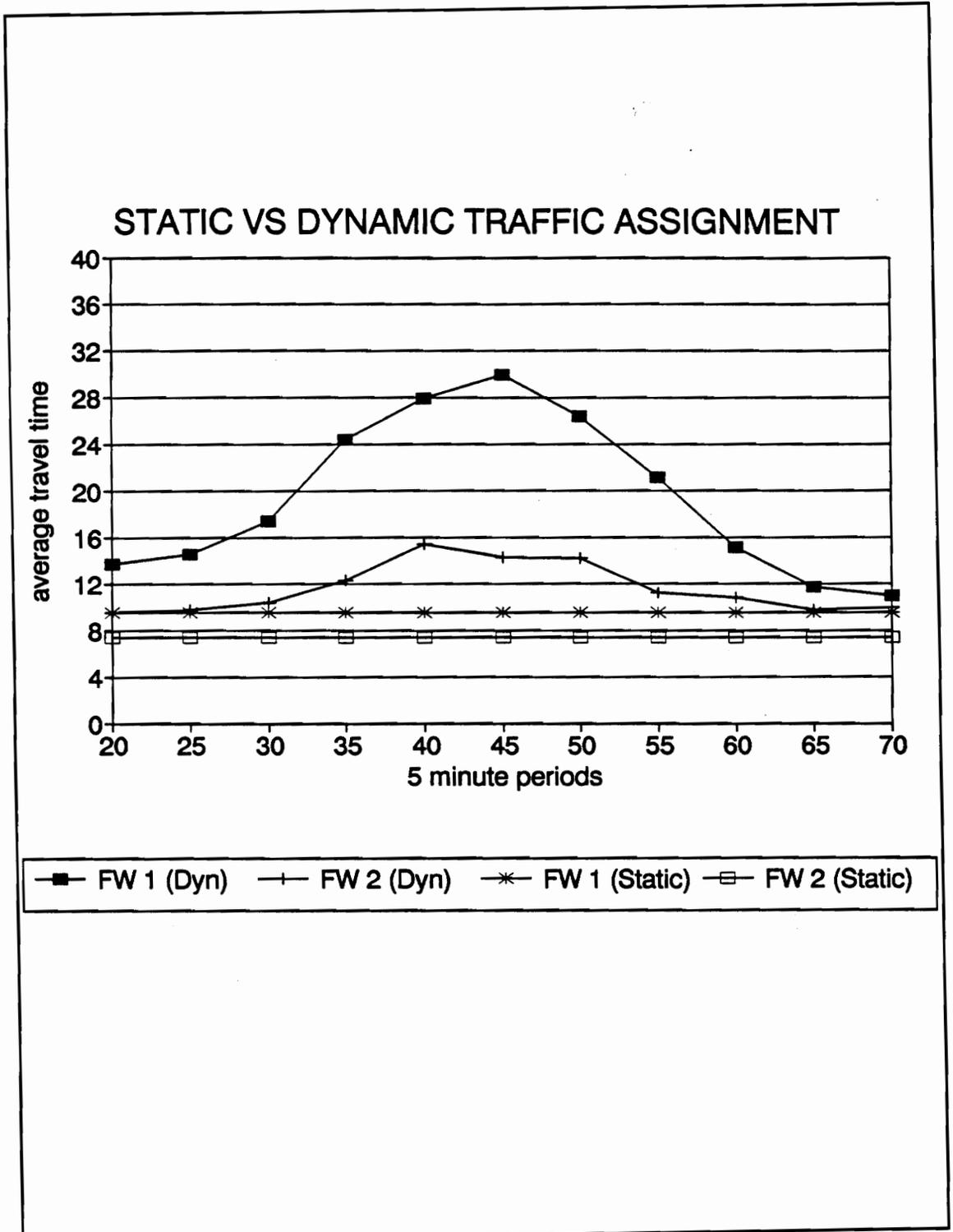


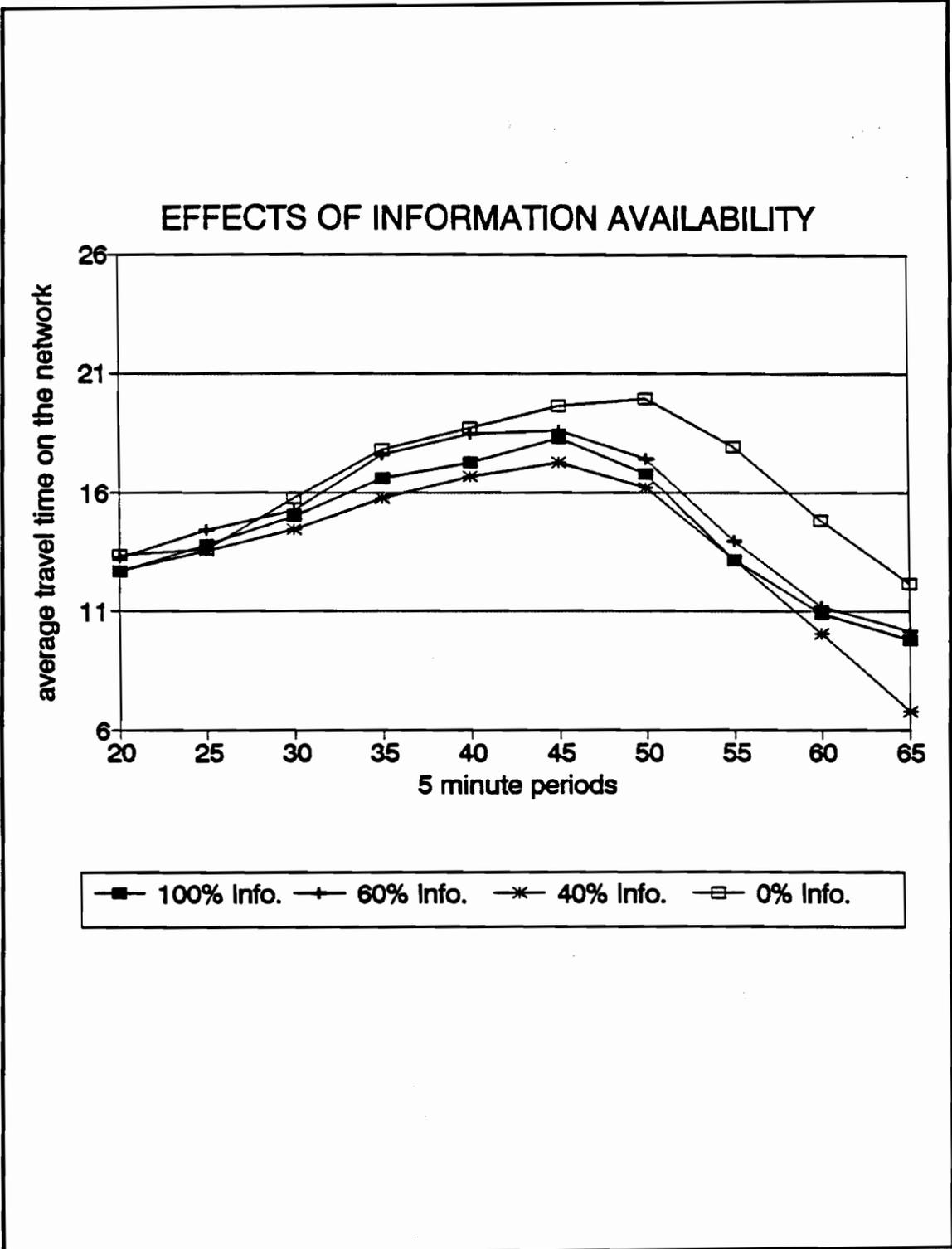
Figure 5.3: Static versus Dynamic Traffic Assignment

shown in Table 5.4. In user equilibrium only thirteen links were used, and other seven links were not used. Out of these thirteen links, six were congested all the time. On the other hand, dynamic method uses an average of nineteen links except last period, and number of congested links are always lower than static equilibrium. A link was considered congested if the concentration of the link is higher than half of the jam concentration, or if the queue length on the link is two times bigger than one minute capacity.

#### **5.4.2 SCENARIO-2**

This scenario is a variation of scenario-1. The O-D split function was the experimental factor that had been changed. It shows the effects of departure patterns in relieving the existing congestion. Figure 5.4 depicts the average travel time changes in system. Although, average travel time changes on selected routes has the same pattern with the base scenario, they are shorter. The reason for better travel times was the constant O-D split matrix that evenly loaded users making better use of the available capacities.

Figure 5.5 clearly shows that the average travel times on the system in scenario-2 are all the time shorter than the ones of scenario-1.



**Figure 5.4: Scenario-2**

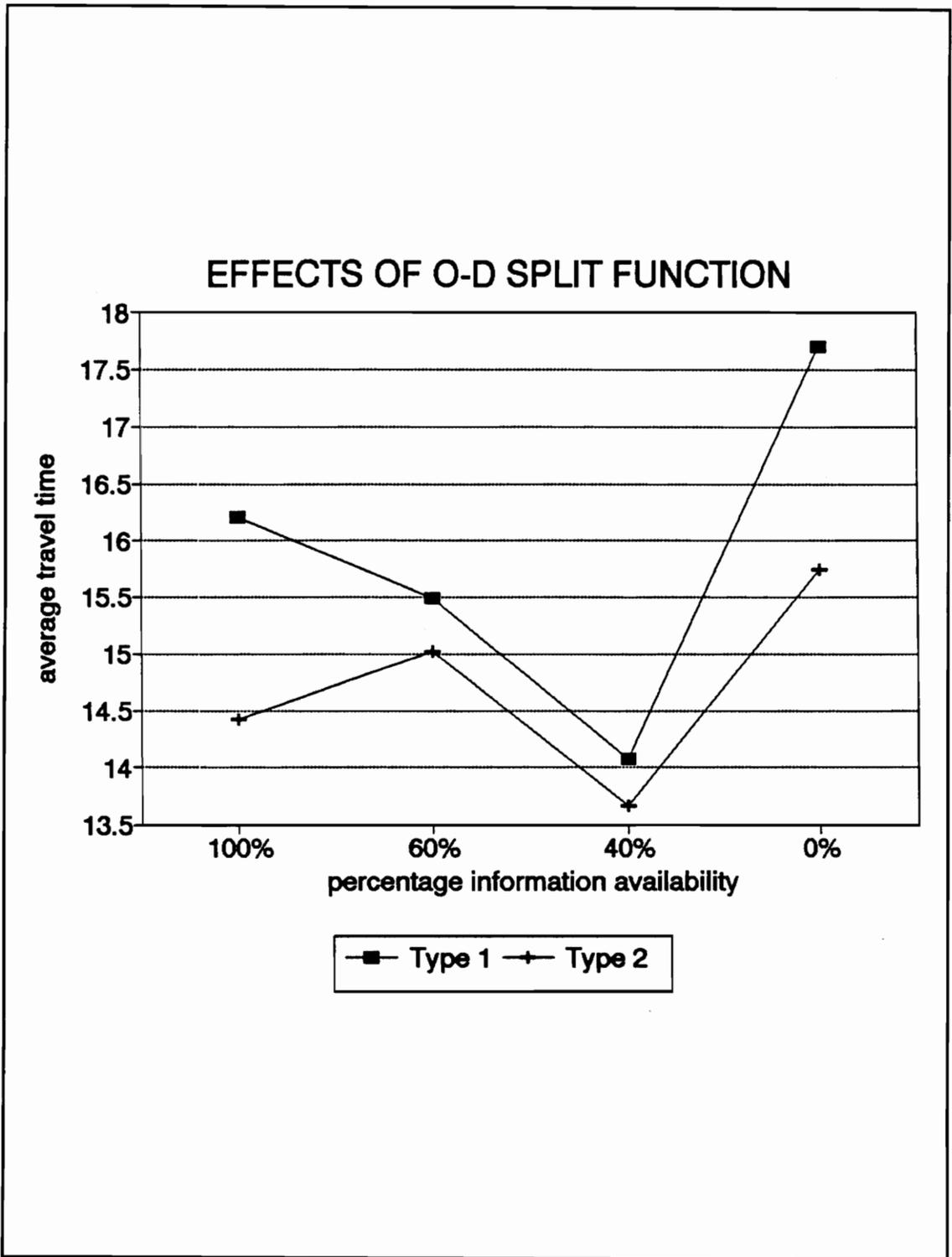


Figure 5.5: Scenario-1 vs Scenario-2

### 5.4.3 SCENARIO-3

In this scenario, the effect of loading was studied. Both O-D matrix and O-D matrix split function were changed. The second loading pattern where 3600 cars per hour depart from the first origin and 4800 cars per hour depart from the second origin was used. In this case Highway-2 was assumed to be congested, but a constant O-D split function used to see the effects of departures on the travel times. The indifference band was taken as zero. Four different fractions of users who are able to receive information were modeled (100%, 60%, 40%, 0%).

Figure 5.6 demonstrates the influence of loading on the travel times. Although the travel time variation patterns were similar to the ones in base scenario, the magnitudes were lower. This shows that initial loading patterns effect overall performance of the system. Loading pattern-2 which loads more cars to the highway with higher capacity and shorter travel times produced better results. In addition to lower travel times, this scenario appears to provide an assignment of vehicles that is closer to an optimal value than the first scenario. This scenario can be used to show the importance of home-base information that can make use of the

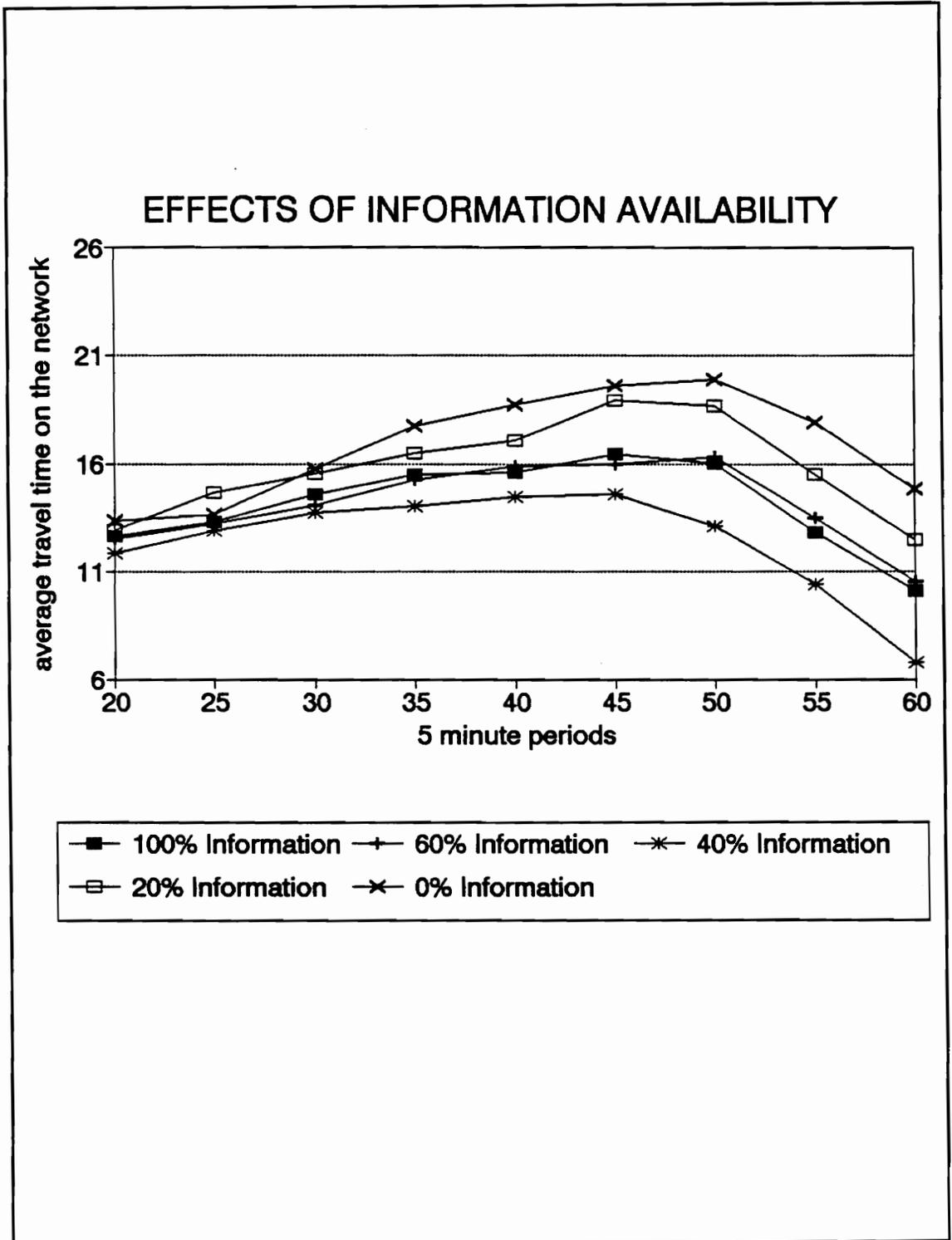


Figure 5.6: Scenario-3

high capacity facilities and regulate departure time patterns in such a way that the capacities are fully used.

#### **5.4.4 SCENARIO-4**

The only difference of scenario-4 from scenario-1 was the indifference band value. It was selected as 0.1. Figure 5.7 shows a slight improvement in travel time due to the indifference band change, but the changes were not as big as expected. The biggest differences were observed between 30 and 50 percent information availabilities where the system produces lowest travel times. Therefore, it can be concluded that the use of an indifference band dampens the negative effects of switching at high levels of information.

### **5.5 CONCLUSION**

Average travel times for each five minute periods were found by averaging travel times on eight selected routes. Then, the average travel time for each scenario was found by averaging these five minute average travel times. These results were shown in Table 5.5. The best result was obtained in scenario-3 where congestion on the network was decreased because of the initial loading pattern and departure time

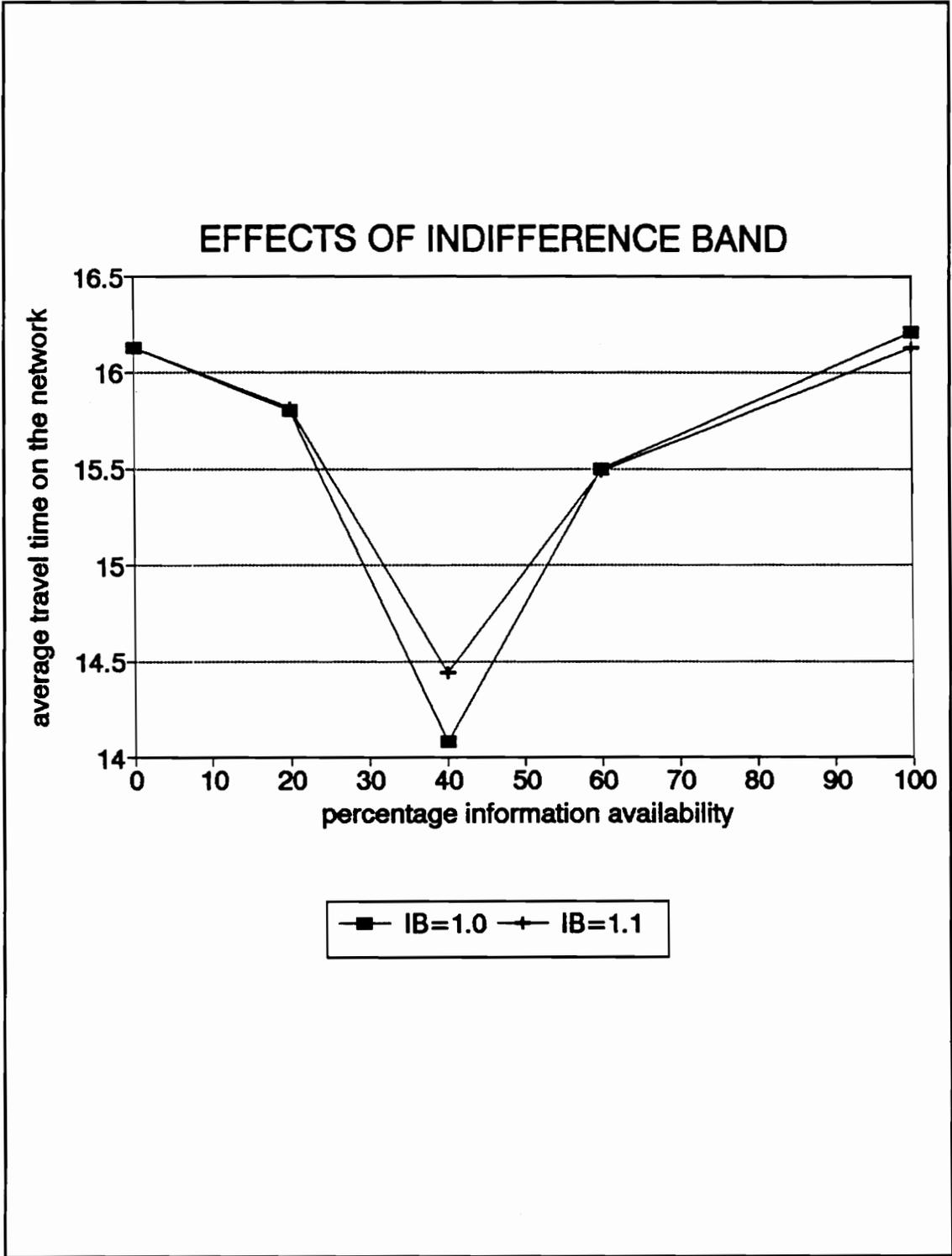


Figure 5.7 Scenario-4 vs Scenario-1

choices. Table 5.6 gives the total travel time in the system expressed as a percent of the system-wide travel time in the no-information base case. Improvements in system-wide travel time seem possible in this specific network. As it can be expected, there appears a decrease in the improvements as the fraction of the equipped vehicles increase. Most potential benefits are achieved between 40 and 60 percent information levels. Finally, static and dynamic traffic assignment generated different travel time values and these values were higher in the case of dynamic model.

**Table 5.5: Average Travel Times in System**

---

INFORMATION AVAILABILITY	S-1	S-2	S-3	S-4
100%	16.12	14.43	14.11	16.1
60%	15.55	15.03	14.15	13.5
40%	14.43	13.67	12.40	14.44
0%	19.31	15.75	15.75	16.94

---

**Table 5.6: Average Travel Times in System, as Percent of Base Case with No Information**

---

INFORMATION AVAILABILITY	S-1	S-2	S-3	S-4
100%	16.52	25.27	26.93	16.62
60%	18.5	22.16	26.72	30.09
40%	25.27	29.21	35.78	25.25
0%	-	18.44	18.44	12.27

---

# 6.0 CONCLUSIONS AND RECOMMENDATIONS

## 6.1 CONCLUSIONS

The simulation/assignment approach to investigate the problem of dynamic traffic assignment in the context of advance driver information systems has proven to be useful. The simulation experiments showed that the relief of urban congestion is possible through the effective use of traffic management systems. The results discussed in the previous chapter proves that traffic management tools like DYNTRAS are needed to understand dynamics of traffic flow in real time. It has been that standard network equilibrium formulation fails to capture the dynamic nature of network performance under time varying demands.

A traditional network equilibrium model deals with the flow pattern during a fixed time interval, generally a peak period. The distribution of traffic during this period is called the temporal distribution of traffic and it is assumed to be fixed and uniformly distributed during the analysis period, but this is not true and it is shown by the simulation experiments that the temporal distribution of traffic flows is

not uniform over the rush period. Instead temporal distribution of vehicular traffic in urban areas depends on the level of the network congestion. DYNTRAS model by using the temporal vehicular concentrations, the locations and extend of queues in predicting peak period operation speeds avoids the main reason of inaccuracy in case of the static traffic assignments. It can also be used to understand the effects of information availability, loading patterns, and behavioral factors on the system performance.

In spite of its advantages cited above, there are several problems associated with dynamic traffic assignment. Among these problems, three directly related to the realistic and efficient formulation of the dynamic traffic assignment is worth discussing.

1. Before starting to develop a dynamic traffic assignment model an exact methodology to keep track of the vehicles on the network should be determined. It is clear that without recognition of the exact location of the vehicles on the network it is very difficult or impossible to achieve dynamic network assignment. Even though there are highly detailed microscopic traffic simulation models like TRAF-NETSIM, they are computationally too expensive. DYNTRAS addressed this problem by adopting macroparticle simulation approach that keeps track of packets of vehicles instead of individual

vehicles.

2. Time representation is another question to be addressed. Like most of the models, DYNTRAS also adopts discrete time representation approach (dividing time into periods). However, there are few models that have adopted continuous time approach [Wie, 91]. Static user equilibrium problem has been studied in both discrete and continuous case. There is also a need for the same sort of studies in case of dynamic user equilibrium problem.

3. Another point of concern is the calculation of travel time. The majority of current dynamic traffic assignment models adopted BPR curves that are based on the steady state capacity and volume without any clear justification. We know the formulation of traditional traffic assignment procedure only applies to the steady state conditions. As a consequence, the existing link capacity functions have been specifically developed for long periods during which effects of any time dependant variations are smoothed over, and they can only reflect the journey speed-flow relationship under steady state conditions. But DYNTRAS should represent the microscopic characteristics of a traffic stream and it cannot ignore fluctuations of traffic flow over time. For small analysis intervals there can be instantaneous changes in flow and in speed. The travel time estimation method used in

DYNTRAS proved to be useful, because it updates the link travel times according to the fluctuations of traffic flow in time.

It was shown that with further enhancements DYNTRAS can be an effective tool to evaluate various control and routing strategies in a dynamic and realistic environment.

## 6.2 RECOMMENDATIONS

The following recommendations are necessary to enhance the DYNTRAS model in order to be able to use it in real world applications.

The proposed recommendations for future research are:

- 1) Enhance the model developed in this research, especially new theories of traffic flow are needed to model dynamic aspects of traffic.
- 2) Enhance the model for simulating different kind of facilities and information sources (home-based, en-route)
- 3) DYNTRAS can be rewritten in another programming language. A portable simulation language, such as SIMSCRIPT, will be a good candidate for that purpose.

4) Gain more knowledge on behavioral aspects of the users. The most difficult part to model is the one that deals with user behavior. Develop different O-D split (distribution) functions for different conditions. Therefore more research is needed on that subject.

5) Develop a travel time prediction mechanism. Path switching decisions that take place from one period to another are based on the current available network information. There is not a widely accepted method which can predict travel time on a given link for the future time periods. The prediction problem is complex one that depends on current and future traffic conditions, user behavior and decisions. It is a very important research topic which needs to be addressed.

## REFERENCES

[Zhang, W., 91] Engineering Design Concept for IVHS Safety Transportation Research Board 70th Annual Meeting, January 13-17, 1991, Washington, DC.

[Wie, B.W., 91] Dynamic Analysis of User Optimized Network Flows with Elastic Travel Demand, presented at the 70th Annual Meeting of the Transportation Research Board, Washington DC, January 1991.

[Hobeika, A.G., Sivanandan, R., Zhang, Y., Ozbay, K., 90] Optimal Diversion Strategies for a Congested Urban Network, Phase 2 Project Report.

[Janson, B.N., 90] A Convergent Algorithm for Dynamic Traffic Assignment, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6366.

[Hammerslag, R., 90] Dynamic Assignment in the Three-Dimensional Time Space, Transportation Research Board, 68th Annual Meeting, January 22-26, 1988, Washington, DC.

[Hammerslag, R., 90] Dynamic Assignment in the Three-Dimensional Time Space, Transportation Research Board, 68th Annual Meeting, January 22-26, 1988, Washington, DC.

[Koutsopoulos, H.N., Lotan, T., 90] Role of Motorist Information Systems in Reducing Traffic Congestion, Final Report to University for Transportation Studies, January 1990, MIT.

[Koutsopoulos, H.N., 89] Vehicle Navigation & Information Systems Conference, Toronto, Ontario, Canada - September 11-13, 1989.

[Hobeika, A.G., Sivanandan, R., 89] Optimal Diversion Strategies for a Congested Urban Network, Phase 1 Project Report.

[Friesz, T.L., Luge, F.J., Tobin, R.L., Wie, B.W., 89] Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem, Operation Research 37(6) pp 893-901.

[Janson, B.N., 89] Estimating Departure Times for Traffic Counts Using Dynamic Assignment, to be presented at the International Conference on Dynamic Traffic Behavior Analysis, Japan, July 18-19, 1991.

[Mahmassani H.S., Kyriacos M.C., 88] Guidelines and Computational Results for Vector Processing of Network Assignment Codes on Supercomputers, Prepared for Presentation at the 68th Annual meeting of Transportation Research Board and Publication in Transportation Research Record, August 1988.

[Mahmassani, H.S., 88] Network Traffic Simulation and Assignment: Super Computer Applications, "Res. Report CRAY-SIN-1988-F, Ctr for Transportation Research, University of Texas, Austin, Tx

[Carey, M., 87] Optimal Time-Varying Flows on Congested Networks Operation Research, Vol. 35, 1987 pp 58-63.

[Mahmassani, H.S., Chang, G.L., Herman, R., 85] Macroparticle Traffic Simulation to Investigate Peak-Period Commuter Decision Dynamics, Transportation Research Rec. 1005, 1984, pp.107-120

[Sheffi, Y., 85] Urban Transportation Networks, Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632.

[Ben-Akiva, M., Cyna, M., DePalma, A., 84] Dynamic Model of Peak Period Congestion, Transportation Research, Vol.18B, 1984, pp.339-355.

[Hendrickson, C., Nagin D., Plank, E., 83] Characteristics of Travel Time and Dynamic User Equilibrium for Travel to Work, in Proceedings of Eight International Symposium on Transportation and Traffic Theory (V.F Hurdle et al. eds.), University of Toronto Press, Toronto 1983, pp.321-347.

[Lisco, 83] Procedure for Predicting Queues and Delays on Expressways in Urban Core Area, Trans Research Record 944, pp.148-154.

[Hendrickson, C., and Plank, E., 81] Characteristics of Travel Time and Dynamic User Equilibrium for Travel to Work Proceedings of the Eight International Symposium on Transportation and Traffic Theory pp 321-347.

[Ho, J.K., 80] A Successive Linear Optimization Approach to the Dynamic Traffic Assignment, *Transportation Science*, Vol. 14, 1980 pp 295-305.

[Leboeuf, J., Tajima, T., Dawson, J., 79] A Magnetohydrodynamic Particle Code for Fluid Simulation of Plasmas. *Journal of Computational Physics*, Vol.31, No 3, June 1979, pp. 379-408

[Merchant, D.K., Nemhauser, G.L., 78] A Model and an Algorithm for the Dynamic Traffic Assignment Problems, *Transportation Science*, Vol. 12, 1978 pp 183-199.

[Merchant, D.K., Nemhauser, G.L., 78] Optimality Conditions for a Dynamic Traffic Assignment, *Transportation Science*, Vol. 12, 1978 pp 200-207.

[Nguyen, S., 74] Algorithm for the Traffic Assignment Problem, *Transportation Science* 8(3), 1974, pp.203-216.

[Beckman, M.J., McGuire, C.B., Winston C.B., 56] *Studies in the Economy of Transportation*, Yale University Press, 1956, New Haven, Connecticut.

## VITA

Kaan Ozbay was born in Istanbul, Turkey on October 19, 1964. He graduated from Lycee de Saint Joseph High School in June, 1983. He attended Bosphorus University (Robert College) in Istanbul, where he graduated with a Bachelor of Science in Civil Engineering in June, 1988. Then he entered the Master of Science program in Geotechnical Engineering at Bosphorus University in September, 1988. He completed his course work in this program at the end of 1989 spring semester. He married Ferzen Dumanoglu on August 10, 1989.

In fall 1989, he came to Virginia Polytechnic Institute and State University in Blacksburg, Virginia and joined the Master of Science program in Civil Engineering with emphasis in Transportation Engineering. He has been a graduate research assistant at the Center for Transportation Research since May, 1990. He worked on a project that attempts to provide different solutions to congestion problems faced in urban areas. He defended his Master thesis at the end of the summer semester, 1991 and will continue his studies as a Ph.D. student.

