

**A Linear Programming Approach for Synthesizing  
Origin-Destination (O-D) Trip Tables from Link Traffic Volumes**

by

**R. Sivanandan**

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

**Doctor of Philosophy**

in

**Civil Engineering**

APPROVED:

Dr. A. G. Hobeika (Co-Chairman)

Dr. H. D. Sherali (Co-Chairman)

Dr. D. R. Drew

Dr. S. C. Sarin

Dr. R. D. Walker

December 1991

Blacksburg, Virginia

**A Linear Programming Approach for Synthesizing  
Origin-Destination (O-D) Trip Tables from Link Traffic Volumes**

by

R. Sivanandan

Dr. A. G. Hobeika (Co-Chairman)

Dr. H. D. Sherali (Co-Chairman)

Civil Engineering

(ABSTRACT)

This research effort is motivated by the need to quickly obtain origin-destination (O-D) trip information for an urban area, without expending the excessive time and effort usually accompanying survey-based methods. The intent is to utilize this information to facilitate diversion of traffic in real time, in the event of congestion-causing incidents such as accidents. The O-D trip table information is a key to successful diversion planning, where user destinations are considered in developing the plans.

Traffic volumes on the links of the road network contain information which can be exploited advantageously to derive the trip patterns. This approach of synthesizing a trip table from link volumes, and perhaps using a prior trip table to guide the derivation, has useful applications in the context of diversion planning. Unlike conventional O-D surveys, it has the potential of yielding results quickly, a requisite for real-time applications.

This research work details a new methodology for synthesizing origin-destination (O-D) trip tables. The approach, which is based on a non-proportional assignment, user-equilibrium motivated, linear programming model, is the principal component of this dissertation. The model is designed to determine a traffic equilibrium network

flow solution which reproduces the link volume data, if such a solution exists. If such alternate solutions exist, then it is designed to find that which most closely resembles a specified target trip table. However, it recognizes that due to incomplete information, the traffic may not conform to an equilibrium flow pattern, and moreover, there might be inconsistencies in the observed link flow data, or there might be incomplete information. Accordingly, the model permits violations in the equilibrium conditions as well as deviations from the observed link flows, but at suitable incurred penalties in the objective function. A column generation solution technique is presented to optimally solve the problem. The methodology also accommodates a specified prior target trip table, and drives the solution toward a tendency to match this table using user controlled parameters. Implementation strategies are discussed, and an illustration of the proposed method is presented using some sample test networks. The results from the model are discussed vis-a-vis other relevant, available approaches. The quality of the results and the computational effort required are used as a set of criteria in the comparisons. The comparisons of test results demonstrate the superiority of the linear programming model over the other models considered.

The model is also applied to a real network of Northern Virginia, where congestion problems present a serious concern. As a result of this experience, several implementation strategies relevant to the application of the model on a real network are presented, and some general conclusions are derived. The potential application of the model in real-time traffic diversion planning for the study area is discussed. Recommendations for further research are also presented.

## **ACKNOWLEDGEMENTS**

The author wishes to express his sincere thanks and appreciation to his major advisor and Co-Chairman of the doctoral committee, Dr. A. G. Hobeika, for proposing this research, and for his constant encouragement, continued advice and financial support. He offered timely suggestions and guidance throughout the course of the work. During the course of association with him, the author was exposed to several topics which are in the forefront of research activities in transportation. In addition to being a well-wisher, he offered useful insights on career choice decisions. The author expresses his gratitude to Dr. H. D. Sherali, who agreed to be the Co-Chairman of the committee. This research would not have reached fruition without his constant input, innovative ideas, understanding and thoughtful remarks, and sustained advice. He has been a source of great encouragement to the author. The author wishes to thank Dr. D. R. Drew for serving in the committee, and for being a source of inspiration throughout the author's course of study at Virginia Tech. Thanks are also due to Dr. R. D. Walker and Dr. S. C. Sarin for serving in the committee, and for their advice. The author expresses his thanks to the US Department of Transportation and

the Virginia Department of Transportation, which sponsored this research project under the University Transportation Centers Program.

Dr. W. A. O'Neill, who has also specialized in the topic of this research, deserves a special mention for her advice, and for kindly permitting the author to use related computer programs developed by her. Mr. D. F. Beagan and Mr. E. J. Bromage were of considerable help during the final phase of the research, by providing related computer programs and results. The author also wishes to thank many transportation departments for providing data.

The author derived the much needed emotional support from his kind, caring, thoughtful and well-wishing parents, grandfather and brother, who have been a constant source of mental strength. During the final phase of this research, the author was married to Kausalya, who also provided emotional support, standing by as a source of encouragement and hope, at a home away from home. She also helped in whatever ways she could. Dr. Sigon Kim, Mr. L. N. Manchikalapudi, Mr. V. R. R. Thummala, and a list of many other friends and well-wishers deserve due thanks for their timely help and cooperation.

#### **ACKNOWLEDGEMENTS**

# **Table of Contents**

<b>1.0 INTRODUCTION .....</b>	<b>1</b>
<b>1.1 Background .....</b>	<b>1</b>
<b>1.1.1 Cogestion Problem .....</b>	<b>3</b>
<b>1.1.1.1 Magnitude of the Problem .....</b>	<b>3</b>
<b>1.1.2 The Concern .....</b>	<b>5</b>
<b>1.1.3 Congestion Mitigation Measures .....</b>	<b>6</b>
<b>1.1.3.1 Traffic Diversion to Reduce Congestion During Incidents .....</b>	<b>7</b>
<b>1.1.3.2 Significance of Real-Time Diversion Planning .....</b>	<b>8</b>
<b>1.1.4 Need for O-D Trip Tables in Diversion Planning .....</b>	<b>9</b>
<b>1.2 Establishing Origin-Destination (O-D) Trip Tables, and the Use of Link Traffic Volumes</b>	<b>10</b>
<b>1.3 Organization of the Dissertation .....</b>	<b>11</b>
<b>2.0 RESEARCH OBJECTIVES AND PROBLEM STATEMENT .....</b>	<b>13</b>
<b>2.1 Overall Research Goal .....</b>	<b>13</b>
<b>2.1.1 User-optimal Approach to Diversion Planning .....</b>	<b>14</b>
<b>2.1.1.1 Estimating Traveller Destinations .....</b>	<b>14</b>
<b>2.2 Relevancy of Research .....</b>	<b>15</b>

2.2.1 Application in Traffic Management Systems .....	16
2.2.2 Application in Traffic Management During Reconstruction .....	16
2.2.3 Working for an IVHS Research & Development Need .....	18
2.2.4 Implementation in Real Time .....	20
2.3 Specific Objectives of This Research .....	22
2.4 Problem Definition .....	24
2.4.1 Issues Related to the Problem of Estimating Origin-Destination (O-D) Trip Tables from Link Volumes .....	25
2.4.1.1 Types of Trips .....	25
2.4.1.2 Stability of O-D Matrix Over Time .....	26
2.4.1.3 Inconsistency in Link Volumes .....	26
2.4.1.4 Traffic Assignment .....	27
2.4.1.5 Underspecification of the Problem .....	28
<b>3.0 SURVEY OF APPROACHES FOR ESTIMATING O-D TRIP TABLES FROM TRAFFIC COUNTS .....</b>	<b>29</b>
3.1 Introduction .....	29
3.1.1 Types of Models .....	30
3.1.2 Parameter Calibration Models .....	31
3.1.2.1 Conceptual Framework .....	31
3.1.2.2 Linear Parameter Calibration Models .....	32
3.1.2.3 Non-linear Parameter Calibration Models .....	34
3.1.3 Matrix Estimation Models .....	35
3.1.3.1 General .....	35
3.1.3.2 Types of Models .....	37
3.2 Summary of Literature Review .....	46

<b>4.0 LINEAR PROGRAMMING APPROACH .....</b>	<b>49</b>
4.1 The Approach .....	49
4.2 Notation and Model Formulation .....	50
4.3 Modifications for Accommodating Prior Trip Tables .....	59
4.4 Illustration of Model Formulation .....	62
<b>5.0 SOLUTION TECHNIQUE .....</b>	<b>65</b>
5.1 Limitations of the Standard Simplex Solution Technique .....	66
5.2 Adopted Solution Technique .....	68
5.3 Column Generation Technique .....	68
5.3.1 Theoretical Background .....	69
5.3.2 Development of a Column Generation Scheme for Problem LP(TT) .....	71
5.3.3 Summary of Column Generation Algorithm (CGA) for Solving LP(TT) .....	77
<b>6.0 MODEL TESTS, COMPARISONS AND VALIDATION .....</b>	<b>82</b>
6.1 Introduction .....	82
6.2 Sample Test Network .....	83
6.3 Computer Resource for Tests .....	85
6.4 Initial Tests of Linear Programming Model LP With No Prior Trip Table Information ..	85
6.5 Target Trip Table .....	87
6.6 Comparison of Test Results .....	89
6.6.1 Replication of Observed Link Volumes on User-Equilibrium Assignment of the Estimated Trip Table .....	90
6.6.2 Closeness of Estimated Trip Table to the Target Table .....	92
6.6.3 Comparison of Computer Run Times .....	93
6.7 Test Results .....	93
6.7.1 Case 1: Use of a "No-Prior-Information" Target Trip Table .....	94
6.7.1.1 Linear Programming Model .....	94

6.7.1.2 Maximum Entropy Model .....	97
6.7.1.3 Network Equilibrium Approach .....	99
6.7.2 Case 2: Use of a "Relatively Small Errors" Target Trip Table .....	102
6.7.2.1 Linear Programming Model .....	103
6.7.2.2 Maximum Entropy Model .....	103
6.7.2.3 Network Equilibrium Approach .....	104
6.7.3 Case 3: Use of the Correct Trip Table as a Target .....	107
6.7.3.1 Linear Programming Model .....	108
6.7.3.2 Maximum Entropy Model .....	108
6.7.3.3 Network Equilibrium Approach .....	110
6.8 Discussion of Results .....	113
6.9 Tests on Additional Networks .....	120

<b>7.0 APPLICATION OF THE LINEAR PROGRAMMING MODEL TO A NORTHERN VIRGINIA NETWORK .....</b>	<b>123</b>
7.1 Introduction .....	123
7.2 The Case Study .....	124
7.2.1 The Traffic Management System (TMS) .....	124
7.2.2 Study Network .....	126
7.2.3 Network Aggregation, Coding and Data Reduction .....	129
7.2.3.1 Network Aggregation .....	129
7.2.3.2 Link Impedances .....	129
7.2.3.3 Network Coding .....	131
7.2.3.4 Data Reduction .....	132
7.2.4 Location of Zones .....	132
7.3 Overall Approach for Diversion Planning .....	134
7.3.1 The Scenario .....	134

7.4	The Approach .....	137
7.5	Implementation Strategies and Model Refinements .....	140
7.5.1	Accountability of Centroid Connector Links with Unknown Volumes .....	140
7.5.2	Prevention of Degeneracy and Cycling .....	141
7.5.3	Circumvention of Numerical Difficulties .....	142
7.6	Model Application .....	143
7.7	General Conclusions on Preliminary Results of the Models .....	144
<b>8.0</b>	<b>CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH .....</b>	<b>148</b>
8.1	Conclusions .....	148
8.2	Recommendations for Further Research .....	150
<b>REFERENCES</b>	.....	<b>151</b>
<b>APPENDICES</b>	.....	<b>157</b>
<b>Appendix A. Backtracing Algorithm</b>	.....	<b>158</b>
<b>Appendix B. INITSOLN Algorithm</b>	.....	<b>161</b>
<b>Appendix C. COMPUTER PROGRAM DOCUMENTATION AND IMPLEMENTATION STRATEGIES</b>	.....	<b>166</b>
C.1	Introduction .....	166
C.2	Overall Program Structure .....	167
C.3	MAIN Program .....	170
C.4	Subroutine INITSONL .....	173
C.5	Subroutine ARTPRICE .....	176
C.6	Subroutine NETFLO .....	177

C.7 Subroutine BACKTRACE .....	180
C.8 Subroutine LEAVE .....	183
C.9 Subroutine UPDATE .....	185
<b>Appendix D. Details of Additional Test Networks .....</b>	<b>187</b>
<b>VITA .....</b>	<b>192</b>

# List of Illustrations

Figure 1. Causes of Urban Congestion .....	2
Figure 2. The Two Approaches for Developing Diversion Strategies .....	14
Figure 3. Relevancy of Research in Traffic Management Systems .....	17
Figure 4. Relevancy of Research in IVHS .....	19
Figure 5. Overall Research Approach .....	21
Figure 6. Example Network for Illustrating Problem Formulation .....	63
Figure 7. "Corridor Network" (Sample Test Network) (Source: Gur et al., 1980) .	84
Figure 8. Plot of Cell Values of "No-Prior-Information" Target and Modelled Trip Tables .....	115
Figure 9. Plot of Cell Values of "Relatively Small Errors" Target and Modelled Trip Tables .....	117
Figure 10. Plot of Cell Values of Correct Target and Modelled Trip Tables .....	118
Figure 11. Computer Run Times for Test Network (Case 2) .....	119
Figure 12. A Variable Message Sign Informing Motorists About a Major Accident (Source: VDOT) .....	125
Figure 13. Northern Virginia Study Area .....	128
Figure 14. The Study Network .....	130
Figure 15. Impact Assessment Districts, as Defined by the Metropolitan Washington Council of Governments (1987), for the Study Area .....	133
Figure 16. The Scenario (Partially Disturbed Network) .....	135
Figure 17. Proposed Approach for User-Optimal Diversion Planning .....	139

Figure 18. The Smaller Study Network .....	146
Figure 19. Overall Program Structure for the Linear Programming Model LP(TT) .....	168
Figure 20. Flowchart for the MAIN Program .....	171
Figure 21. Flowchart for the INITSOLN Subroutine .....	175
Figure 22. Flowchart for the ARTPRICE Subroutine .....	178
Figure 23. Flowchart for the BACKTRACE Subroutine .....	181
Figure 24. Flowchart for the LEAVE Subroutine .....	184
Figure 25. Flowchart for the UPDATE Subroutine .....	186
Figure 26. Test Network 2 (Source: Nguyen, 1977) .....	187
Figure 27. Test Network 3 .....	189
Figure 28. Test Network 4 .....	190
Figure 29. Test Network 5 .....	191

# List of Tables

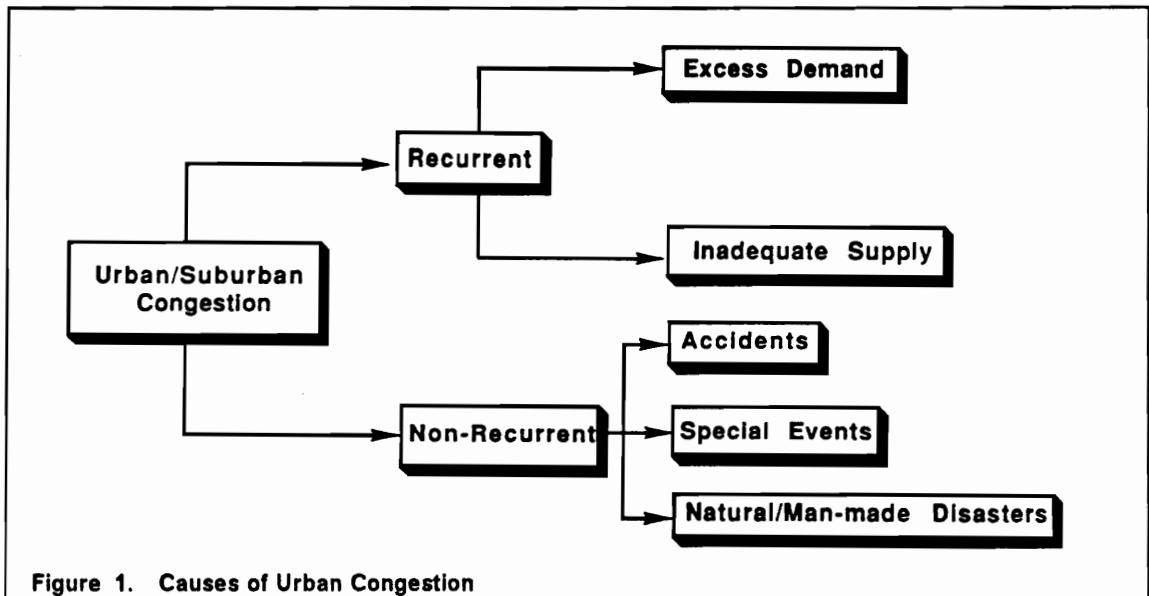
Table 1. Urban Freeway Congestion in U.S. (Source: Lindley, 1989) .....	4
Table 2. "Corridor Network" Characteristics (Source: Gur et al., 1980) .....	84
Table 3. Solution through Linear Programming Model LP (Without Prescribing a Target Trip Table) - Advanced-Start Procedure .....	86
Table 4. Solution through Linear Programming Model LP (Without Prescribing a Target Trip Table) - Artificial-Start Procedure .....	86
Table 5. "No-Prior-Information" Target Trip Table .....	88
Table 6. "Relatively Small Errors" Target Trip Table .....	88
Table 7. Correct Target Trip Table .....	89
Table 8. Advanced-Start Basis for Linear Programming Model LP(TT) (Case 1) ..	96
Table 9. Final Trip Table Using Linear Programming Model LP(TT) - Advanced-Start Procedure .....	99
Table 10. Final Trip Table Using Linear Programming Model LP(TT) - Artificial-Start Procedure .....	100
Table 11. Final Trip Table Using Maximum Entropy Model .....	100
Table 12. Final Trip Table Using LINKOD Model (238 Iterations) .....	100
Table 13. Final Trip Table Using LINKOD Model (297 Iterations) .....	101
Table 14. Comparison of Equilibrium Assigned and Observed Link Volumes ..	101
Table 15. Closeness of Estimated Trip Table to the Target Table .....	102
Table 16. Comparison of Computer Run Times for Test Network .....	102
Table 17. Advanced-Start Basis for Linear Programming Model LP(TT) (Case 2) ..	104

Table 18. Final Trip Table Using Linear Programming Model LP(TT) - Advanced-Start Procedure .....	105
Table 19. Final Trip Table Using Linear Programming Model LP(TT) - Artificial-Start Procedure .....	105
Table 20. Final Trip Table Using Maximum Entropy Model .....	105
Table 21. Final Trip Table Using LINKOD Model (211 iterations) .....	106
Table 22. Final Trip Table Using LINKOD Model (393 iterations) .....	106
Table 23. Comparison of Equilibrium Assigned and Observed Link Volumes ..	106
Table 24. Closeness of Estimated Trip Table to the Target Table .....	107
Table 25. Comparison of Computer Run Times for Test Network .....	107
Table 26. Advanced-Start Basis for Linear Programming Model LP(TT) (Case 3)	109
Table 27. Final Trip Table Using Linear Programming Model LP(TT) - Advanced-Start Procedure .....	110
Table 28. Final Trip Table Using Linear Programming Model LP(TT) - Artificial-Start Procedure .....	110
Table 29. Final Trip Table Using Maximum Entropy Model .....	110
Table 30. Final Trip Table Using LINKOD Model (226 iterations) .....	111
Table 31. Final Trip Table Using LINKOD Model (477 iterations) .....	111
Table 32. Comparison of Equilibrium Assigned and Observed Link Volumes ..	112
Table 33. Closeness of Estimated Trip Table to the Target (Correct) Table ..	112
Table 34. Comparison of Computer Run Times for Test Network .....	112
Table 35. Run Times and Replication of Link Volumes for Additional Test Networks .....	121
Table 36. Test Network 2 Characteristics (Source: Nguyen, 1977) (Data Slightly Modified) .....	188
Table 37. Test Network 3 Characteristics .....	189
Table 38. Test Network 4 Characteristics .....	190
Table 39. Test Network 5 Characteristics .....	191

# 1.0 INTRODUCTION

## 1.1 *Background*

The need to address various problems associated with the growing traffic congestion flows in urban areas motivated this research effort. Traffic demand in most metropolitan areas is steadily increasing, but the transportation infrastructure has been unable to expand at the same pace. Thus, demand has outpaced supply. An inevitable consequence of such a trend is traffic congestion. While recurrent congestion has become an everyday event, non-recurrent congestion is also becoming a common phenomenon due to *incidents* ranging from accidents and chemical spills to disasters, further adding to the problem. Figure 1 depicts the major causes of congestion. Many congestion-reducing measures are being experimented with in most urban areas facing such problems. While these measures have succeeded partially, the concern for the problem still remains, and is growing.



**Figure 1. Causes of Urban Congestion**

This research effort was initiated with the intention of contributing to a particular aspect of alleviating congestion through traffic diversions, realizing that developing and implementing effective and efficient diversion strategies can reduce congestion costs significantly. A crucial information in developing a successful diversion plan is the knowledge of destinations of travellers to be diverted at the time of implementation. This information enables the diversion of motorists through best alternative routes to their destinations. In this research work, a model is developed to synthesize this information from traffic volumes present in the road network. The advantage of such an approach is its potential for application during real-time diversion planning. The developed model and methodology have been applied to a network of Northern Virginia, a region facing severe congestion problems. Below, we discuss some of these motivating issues, before presenting our methodology and results. This research was supported by a grant under the US DOT/VDOT University Transportation Centers Program, under the auspices of Mid-Atlantic Universities Transportation Center.

### **1.1.1 Cogestion Problem**

The "Statement of National Transportation Policy", issued by the U.S. Department of Transportation, underscores the need to "maintain and expand the nation's transportation system", with particular commitment "to reducing congestion in the aviation and highway systems" (US DOT, 1990). Congestion is also identified as a major concern of urban transportation users by the "Transportation 2020 Program" (2020 Transportation Program, 1988). Several other professional bodies have expressed similar sentiments. Clearly, urban traffic congestion is one of the major problems facing the transportation community. The gravity of the problem is further highlighted by the interest it draws from academic, private and public institutions alike.

#### ***1.1.1.1 Magnitude of the Problem***

Many estimates are available from different agencies of the magnitude and seriousness of congestion problems. While the parameters for measuring congestion may differ for various estimates, the message is the same - that the problem is costing heavily, and deserves serious attention. Though it may be of interest to summarize all the statistics on the problem, a few indicators as listed below are considered sufficient in pronouncing the magnitude of the problem.

Table 1 provides a statistical summary and future projections for urban freeway congestion in U.S., followed by some more indicators of rising congestion.

**Table 1. Urban Freeway Congestion in U.S. (Source: Lindley, 1989)**

Type	Unit	1987	2005
Recurring delay	(million vehicle-hours)	728	3,030
Delay due to incidents	(million vehicle-hours)	1,287	7,978
Total delay due to congestion	(million vehicle-hours)	2,015	11,008
Total wasted fuel	(million gallons)	2,206	11,638
Total user costs	(billion dollars)	15.9	88.2

- Traffic volumes on America's highway network are predicted to double from 1.9 trillion vehicle miles of travel (VMT) in 1988 to 3.8 trillion VMT in 2020 (US DOT, 1989), with probable increase in urban congestion.
- Commuting to work by car has increased, and they account for over four-fifths of all work trips (Meyer et al., 1989), adding to traffic congestion.
- Since 1970, travel on urban areas has doubled and travel on urban arterials has tripled (FHWA, 1985).
- 65 percent of peak hour travel on urban interstates occurred under congested conditions in 1987. This figure rose from 61 percent in 1985 and 54 percent in 1983. Also, urban travel increased by 9.4 percent between 1985 and 1987 (FHWA, 1989).
- The percent of urban interstates facing severe congestion (operating at levels of service D and E) rose from 16% in 1981 to 30% in 1988 (Stowers, 1988 & FHWA, 1988).

- Traffic incidents account for roughly 60% of the vehicle-hours lost to congestion.

The consequences of highway congestion are reduced mobility, economic losses, and decreased productivity. Thus, the congestion problem is assuming a larger dimension and poses an ever increasing challenge to traffic engineers and managers.

### **1.1.2 The Concern**

In many instances, traffic congestion results in closure of streets or reduction in their traffic carrying capabilities, disabling the road network in meeting the traffic demand. This results in traffic chaos, delays, and uncertainties. Based on a study of congestion levels in major urban areas, travel delay constitutes about 65% of congestion costs (Hanks and Lomax, 1991). Concern for this problem is growing in many urban areas around the world. Attempts are underway in many cities to execute mitigation measures. Success has been only partial, because of the complex nature of the problem. A variety of potential solutions are being investigated to relieve the situation. Some of the measures include: 1) demand restriction policies, 2) increasing transportation supply, and 3) improving transportation system management strategies. Another solution of recent focus is the ambitious Intelligent Vehicle/Highway Systems (IVHS). U.S., Europe, and Japan have on-going research efforts in this field. It is noteworthy to mention here that in 1988, the U. S. congress directed the Secretary of Transportation to submit a comprehensive report on advanced vehicle highway technology research and development initiatives (U. S. Congress, 1988). Subsequently, the Federal Highway Administration produced a report elaborating on the various components of IVHS technologies, progress of

research and development, both in the U.S. and elsewhere, timeframe for reaching different stages of progress for U.S. efforts, along with financial needs. It also underscored the fact that U.S. is lagging behind some other nations in this field. The momentum to pursue research and development activities in IVHS is picking up in U.S. since then. The Intelligent Vehicle Highway Society of America (IVHS America) has recently been formed to coordinate the efforts of various organizations, including those engaged in designing and manufacturing products related to IVHS. Transportation authorities concur that IVHS is essential in alleviating traffic congestion, in addition to bringing about other benefits.

### **1.1.3 Congestion Mitigation Measures**

The obvious solution to the congestion problem may appear to be the construction of new facilities to accommodate the increased demand. However, the common experience in the past has been that, as the supply is increased, the demand also increases, soon outpacing the supply, thereby further leading to congestion. Moreover, increasing supply is infeasible in most cases, due to exorbitant costs, lack of construction space, and environmental considerations. On the other hand, demand restriction policies are an attractive solution to the problem. Telecommuting has a vast potential in reducing the demand considerably. However, it entails implementation difficulties in linking homes with offices through communication channels. Car pooling and the designation of high occupancy vehicles (HOV) lanes are other demand restriction measures. Although some of these have been partially successful, implementing such policies to restrict the demand are relatively difficult tasks. Solutions like Intelligent Vehicle/Highway Systems (IVHS)

are generally long term in nature and immediate benefits are difficult to be achieved through these technologies. An efficient use of the highways that are already in place is an effective strategy that can reduce the need for new capital investments. It can also bring about reduction in congestion levels. The effective and efficient diversion of traffic during an incident is one such contributing factor.

#### ***1.1.3.1 Traffic Diversion to Reduce Congestion During Incidents***

Among the various measures for congestion reduction, the potential of traffic diversions is yet to be fully tapped. Diversion planning is a key element in incident management. Since incidents account for causing about 60% of congestion, the importance of effecting diversions need not be stressed. This is a complex task, since it involves diverting users with different destinations in dynamically changing traffic conditions. Complicating the task further is the degree of compliance of diversion directions, by the motorists.

**Diversion Planning:** Effective and efficient diversion planning could minimize user travel time and cost. Many state departments of transportation engage in such diversion exercises. However, it appears that their efforts are focused on freeway corridors where alternate routes for diversion are limited. Also, diversion plans are predetermined, and in the event of an incident, the appropriate plan is chosen for implementation. However, when an area-wide diversion is attempted, pre-planning of diversion routes may be difficult due to the presence of multiple, alternate routes. The behavior of motorists to diversion will also impact the effectiveness of pre-determined alternate routes in minimizing the congestion. Thus, the need is to

plan for diversion in real-time, by providing the motorists with timely information on best alternate routes.

#### ***1.1.3.2 Significance of Real-Time Diversion Planning***

Pre-determined diversion plans are of limited use since the time and location of most incidents 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, that is in real-time conditions. Also, most incident sites are cleared in a few hours, creating an additional requirement that these diversion plans must be devised quickly, to be of use in diverting traffic during the period while the disruption lasts. This poses a challenge to traffic engineers and managers. The task is made difficult on two accounts: (1) the complexities of prediction of alternate diversion routes due to the dynamic nature of traffic flow, and (2) the computational difficulties associated with evolving plans in a short period of time. Hence, there is a need for developing algorithms/techniques that would facilitate this task. One of the sub-tasks is to develop a model for estimating traveller destinations in real-time. The sole aim of this research is to develop such a model. The basis for such a model is to establish an O-D trip table that could be used to determine user-optimal diversion plans. O-D trip tables could also be pre-constructed for regions, based on recent traffic volume data, for later use in the event of an incident in the region.

#### **1.1.4 Need for O-D Trip Tables in Diversion Planning**

When a congestion causing event like a major accident occurs, the basic functions of the network in the region are placed in jeopardy. Under such situations, affected links or an area is cordoned off to traffic by authorities. This upsets the prevailing flow in the network, thereby causing further disturbances in the network. The disturbances might be felt not only in the vicinity of the cordoned area, but also at distant locations.

A disturbance could manifest in the form of stoppages of vehicles, delays, and detours. Whatever the case, the impact is of great concern not only to users of the network, but also to traffic authorities who are responsible for traffic management. Under such a situation, a principal decision issue that authorities will address is how to minimize traffic chaos, delays, and detours, thus saving overall cost to users of the network. The key answer to the question is the development of *effective and efficient diversion strategies* in real time. This issue can be addressed effectively if the traffic engineer/planner has knowledge of destinations of affected traffic. Estimating trip destinations will facilitate the formulation of efficient diversion strategies. This, in turn, will result in considerable savings to users, in the form of time and fuel, by optimal rerouting within imposed network constraints.

## **1.2 Establishing Origin-Destination (O-D) Trip Tables, and the Use of Link Traffic Volumes**

An origin-destination (O-D) trip table is a two-dimensional matrix of elements whose cell values represent the number of trips made between various O-D zone pairs in a given region. The interest here is on the highway trips made in an urban area. Conventionally, the information presented in the O-D trip table are established through extensive surveys of the travellers. These surveys include home interviews, license plate surveys, road side surveys etc. These survey techniques are expensive, time consuming, and labor intensive. Most of the these methods are conducted through sampling. Thus, it is impossible to determine a "real" trip table. Even if all the trips on a particular day are recorded, the O-D table so determined may not be stable over time, due to variations from day to day (Willusen, 1978). There are other inherent drawbacks associated with conventional techniques. One common problem, not peculiar to trip table estimation, is ~~the changes in pattern due to changes in influencing factors.~~ For instance, as the land use develops or changes rapidly, so will the trip table. Thus, the previously established trip table becomes outdated. This will necessitate re-surveying, leading to further expenditures and efforts. Another resulting setback of these techniques is that planning agencies of small urban areas are often unable to conduct such surveys due to budgetary and time restrictions. Nevertheless, these agencies do require trip table information for many planning and management purposes. This necessitates obtaining trip tables through cheaper and quicker means, and is the motivating factor leading to the development of theoretical approaches for synthesizing trip tables using available information in the form of link

traffic counts. While a trip matrix obtained through a conventional survey relies on only a small sample of tripmakers, the link counts method constitutes nearly the entire sample, thus utilizing greater information. The central idea in all these approaches is to use the information available in the link volume data to derive a trip table that will replicate the observed link volumes as closely as possible. Recent advances in trip table synthesis has opened a new avenue for its potential application in real-time incident management, when the O-D information pertaining to the motorists on the network aids in developing effective diversion plans. It also has applications in the development of Intelligent Vehicle/Highway Systems (IVHS), which has evoked considerable interest lately.

This dissertation presents a new linear programming based approach for synthesizing a trip table from observed link flows on the network. In contrast to complex nonlinear approaches like maximum entropy and network equilibrium, the linear programming model developed here achieves the same solution in a more efficient and straightforward manner, and promises potential computational gains in application.

### ***1.3 Organization of the Dissertation***

The organization of this dissertation is as follows. Chapter 2 presents the overall research goal, relevancy of the research, specific objectives of this research, and the problem statement. Chapter 3 reviews the existing approaches and models for trip table estimation using link counts. Chapter 4 presents the linear programming model and formulation of the problem. Chapter 5 details the solution technique and

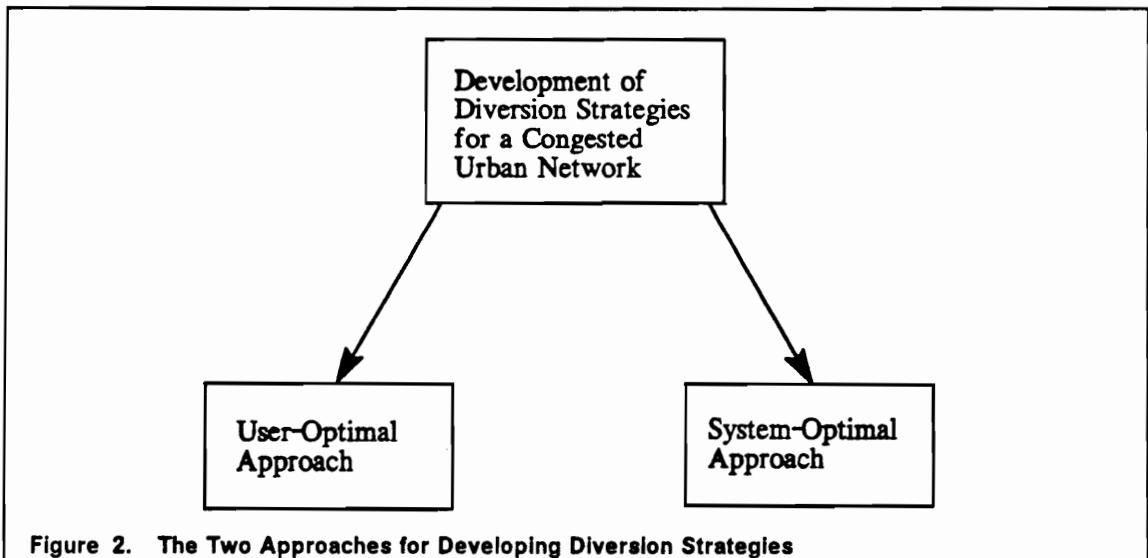
implementation strategies developed to solve the problem. Model tests, comparisons, and validation through the use of test networks are presented in Chapter 6. An application of the model to a real network of Northern Virginia, certain computer implementation strategies, and general conclusions on preliminary results are detailed in Chapter 7, followed by conclusions and recommendations for further research in Chapter 8. Computer program documentation and implementation strategies are contained in Appendix C.

## **2.0 RESEARCH OBJECTIVES AND PROBLEM STATEMENT**

### **2.1 *Overall Research Goal***

The overall goal of this research project is to develop a tool to assist the traffic managers in formulating strategies in real time, for effectively and efficiently diverting traffic during congestion-causing events (Hobeika et al., 1989, 1990, 1991). To achieve this, a two-pronged research approach, as shown in Figure 2, is being adopted.

In the user-optimal approach, the objective is to minimize the delay due to diversion for individual users. On the other hand, in the system-optimal approach, an attempt is made to effect diversions that optimize the utility of the system facility. The research work reported in this dissertation adopts the user-optimal approach, because of the independent nature of route selection by the users.



### 2.1.1 User-optimal Approach to Diversion Planning

The underlying philosophy in this approach is to supply diversion information with a view to route each of the motorists through the best alternate route to his or her destination, bypassing the incident location. This necessitates acquiring a crucial information - that of motorist destinations. Once this information is at hand, devising alternate routes can be accomplished through a traffic assignment process. Thus, the two major objectives in this research approach are the estimation of traveller destinations, and an evaluation of diversion alternatives.

#### 2.1.1.1 *Estimating Traveller Destinations*

To accomplish the task of rerouting, the destinations of motorists near the incident and those approaching the incident location need to be determined. This will

enable advance rerouting notice to the motorists so they can take alternate routes avoiding the incident. The simple-minded approach is to enquire each motorist about his/her travel plan, and divert him/her accordingly. This is, however, impractical and will lead to greater confusion and delay. The simplest way then is to acquire existing O-D survey results for the area, which can then be used to establish the traveller destinations. However, this has many drawbacks. Firstly, the O-D trip table may not be available for the area in question. Secondly, even if available, it may be an outdated one, thus making it worthless. Thirdly, the table may represent a whole-day period travel data, which may not be able to identify the travel patterns for the specific short periods of interest covering the incident duration. Under such circumstances, estimating traveller destinations by utilizing normal traffic flow volumes on links is of greatest value. This method has the advantage of presenting information about the currently existing traffic in the network that needs to be diverted, and forms the basis for the user-optimal approach adopted herein.

## **2.2 *Relevancy of Research***

This research has three relevant and important applications. These pertain to operation of traffic management systems, traffic management during reconstruction, and to the Intelligent Vehicle/Highway Systems (IVHS), a topic of recent and major focus in highway transportation.

## **2.2.1 Application in Traffic Management Systems**

There are many state-of-the-art traffic management systems in U.S. that are under development or in operation. One of the management facilities is located in Arlington, Virginia, and is considered to be among the most advanced centers (VDOT). In the event of an incident, the tasks of this traffic control center are to alert the police and the media, to control ramp meters, and to display locations of congestion to the motorists through variable message signs. However, no diversion information is relayed to inbound traffic. Few control centers are known to display alternate route plans. One technique for determining alternate routes is to assess the status of traffic on unaffected streets, and scan them for available capacities that can be used for diversion purposes. Although this might satisfy the operator in utilizing the road system optimally, motorists may not be taking the best alternate routes to their destinations. The deficiency of this procedure lies in the fact that the rerouting plans are charted out without considering motorist destinations!

This research project proposes to address this deficiency by utilizing real-time traffic counts collected continuously by control centers. Traffic volume information constitutes an input for estimating motorist destinations, based on which diversion choices can be assessed on a real-time basis. The potential application of this research in traffic management is shown conceptually in Figure 3.

## **2.2.2 Application in Traffic Management During Reconstruction**

This research assumes further importance in the context of major U. S. highway reconstruction plans. Diversion planning comprises an integral part of highway

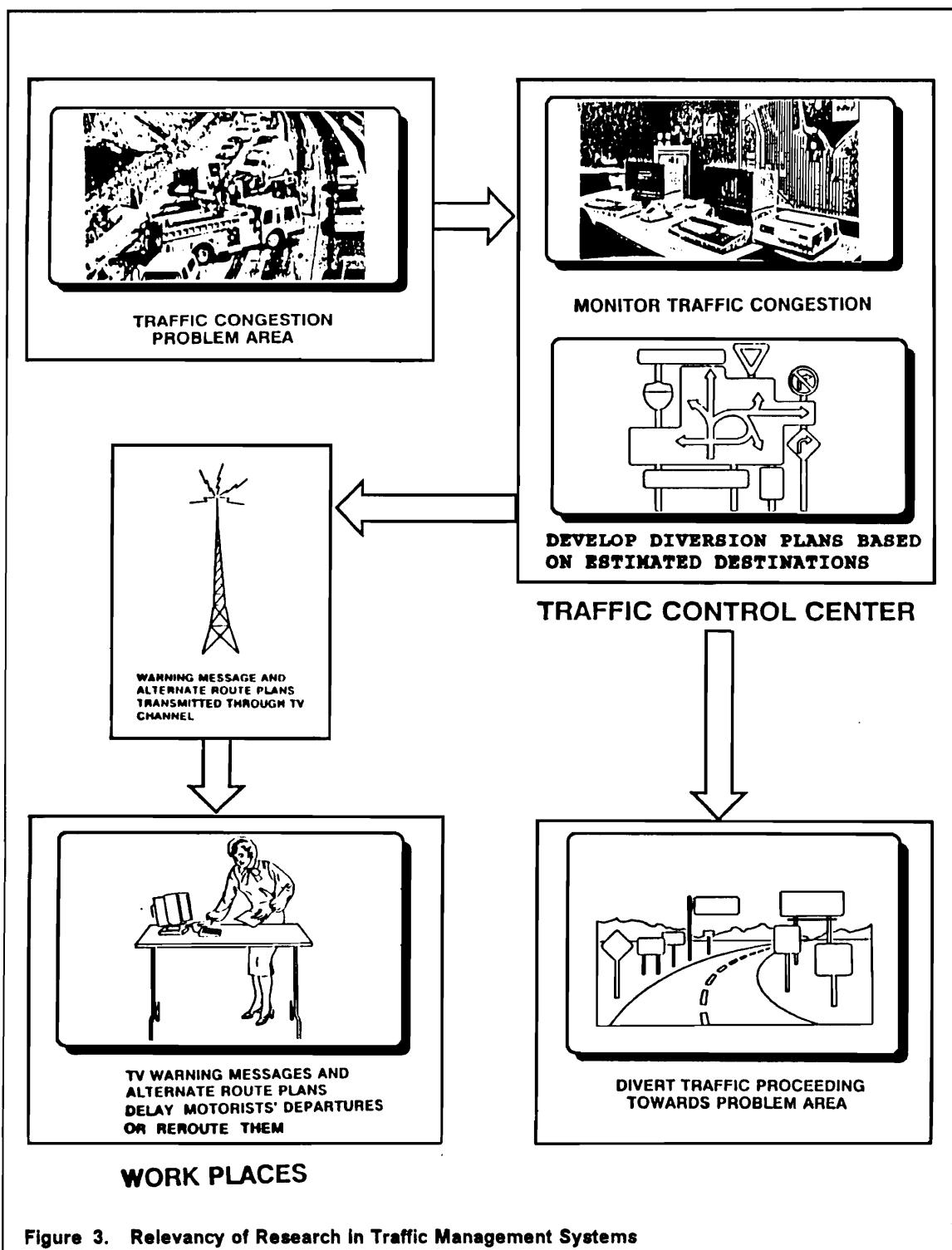


Figure 3. Relevancy of Research in Traffic Management Systems

reconstruction. A tool for developing appropriate diversion plans during reconstruction would be very valuable. Thus, the product of this research would be of significant aid to the traffic engineer in charge of reconstruction.

### **2.2.3 Working for an IVHS Research & Development Need**

The subject of Intelligent Vehicle/Highway Systems (IVHS) has recently attracted major attention of the transportation community, the public and the politicians, due to its potential for enhancing safety, improving energy efficiency, and reducing environmental and economic impacts of automobile transportation. These technologies will also greatly contribute to reducing urban traffic congestion through advanced traffic management strategies, and by increasing road capacities many fold by automatic control of vehicles.

Four categories of elements have been identified for the Intelligent Vehicle/Highway Systems (IVHS) (US DOT, 1989). They are:

1. Advanced Traffic Management Systems (ATMS)
2. Advanced Driver Information Systems (ADIS)
3. Freight and Fleet Control Operations
4. Automated Vehicle Control Systems (AVCS)

One major research task that has been identified for traffic management systems is the development and testing of models and algorithms for optimal routing and diversion. This is precisely the objective of this dissertation research. Figure 4 illustrates the different components of IVHS, and the relevancy of this research in the

overall IVHS context. Hence, this research will contribute to both short- and long-term efforts to mitigate congestion problems.

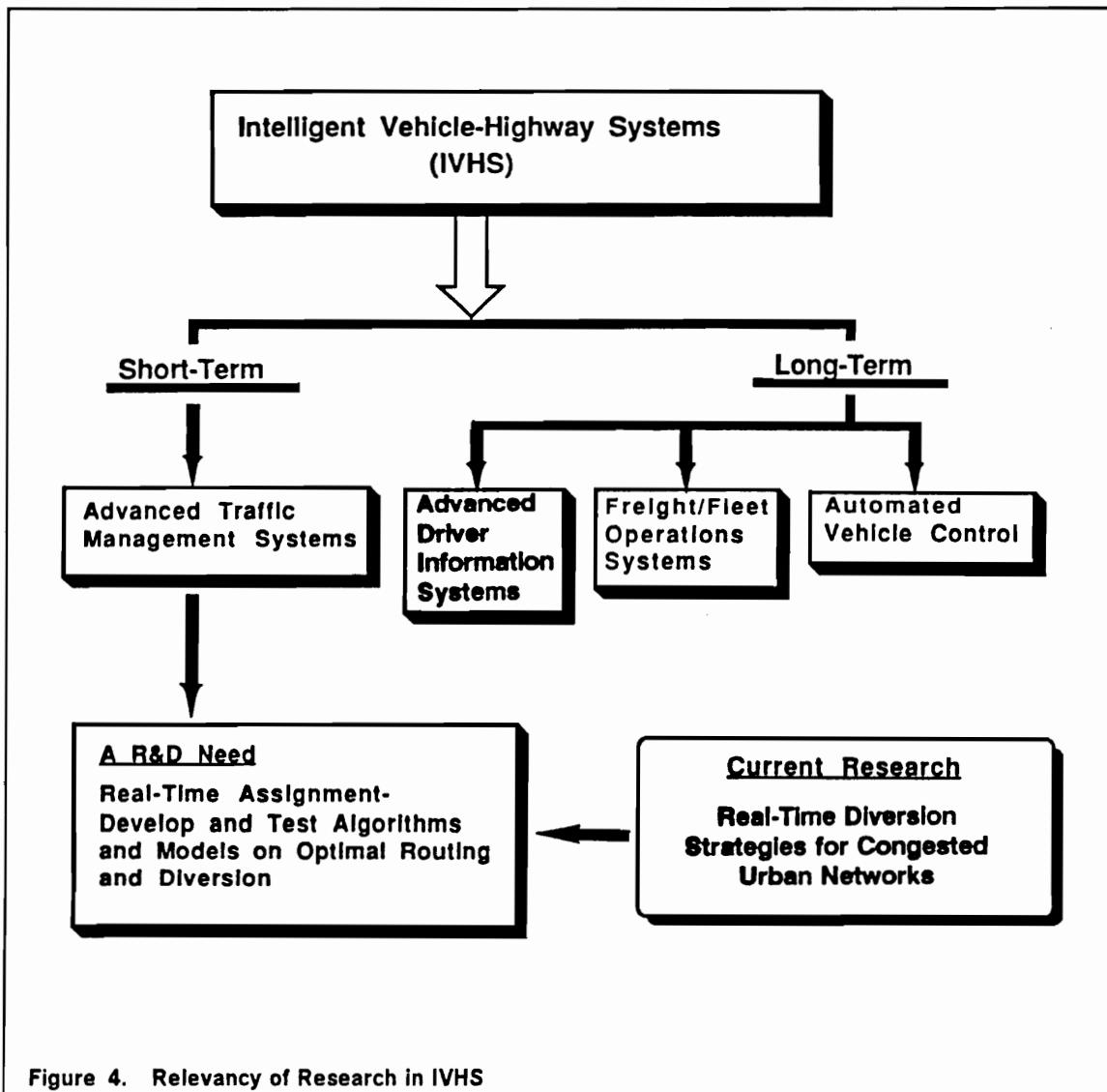


Figure 4. Relevancy of Research in IVHS

## **2.2.4 Implementation in Real Time**

The goal here is to implement diversion in real time. This entails a constant updating of traffic volumes. Since this is currently being done in most traffic control centers, these data are put to good use here. In most circumstances, prior knowledge of network failure does not exist. To implement a quick and efficient diversion, complete information on link volumes, capacity, and other characteristics of the network, must form the computerized data base. Details of affected links can be immediately input, and the destination estimation and diversion assessment models can then be run to quickly formulate diversion plans. A graphics capability that displays the network and the disturbed region on the computer screen will greatly enhance the use of the models, and aid quick decision making.

The diversion plans and strategies chosen can then be implemented by posting signs to direct traffic, by sending radio and television advisories, and by deploying personnel to monitor traffic. Appropriate Transportation Systems Management (TSM) actions can be taken, if warranted. As diversion progresses, the network conditions must be constantly assessed to evolve further plans appropriately. With the growing popularity of traffic monitoring and control through centralized control centers in more and more urban areas, changeable message signs can be employed to advise traffic of alternate routes.

The overall approach is shown in Figure 5.

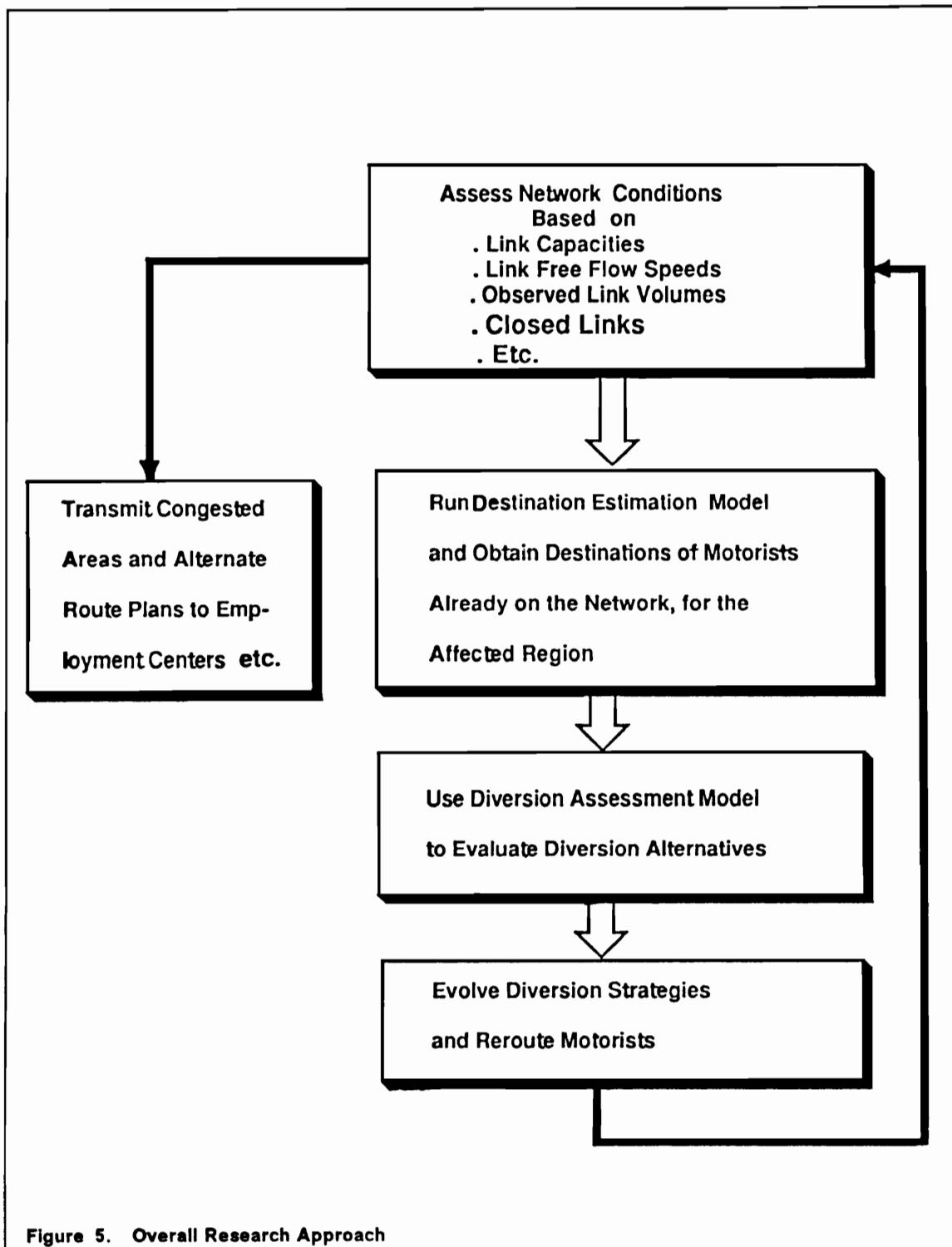


Figure 5. Overall Research Approach

## **2.3 Specific Objectives of This Research**

Within the framework of the project goal, and with an intent to partially fulfil the overall objectives, the task of building a model to estimate the destinations of motorists was taken up as the major objective of this research. Chapter 1 exposed the need for synthesizing a trip table utilizing traffic counts on links, to enable diversion planning in the event of congestion-causing incident. The information contained in the link volumes can be exploited to arrive at a trip table that is representative of trip patterns in the region. Consequently, the interest is in determining a trip table that replicates the observed link flows as closely as possible, when assigned to the network. However, it is possible that more than one trip table that are significantly different might conform to the condition of replicating link volumes. In such a case, the choice of a trip table as the true trip table becomes problematic. However, this can be overcome by utilizing information in the form of a prior or target trip table. The strategy here is to guide the solution trip table to conform to the trip patterns of this target table, in addition to replicating link flows as closely as possible. In fact, many of the existing approaches use this concept in their model development. However, these models are computationally demanding, which makes their potential application to real-time situations questionable.

Another issue pertains to the route choice behavior of the motorists. Do they always choose the least time path route from their origins to destinations ? Or do they have other preferences in their route choice ? Since this issue is a factor in the assignment step of the model development, an appropriate assumption must be made beforehand. It is accepted that most commuters in urban areas attempt to take the least-time path, given the status of the system traffic. Also, usually there are

multiple paths between O-D pairs. The selection of a least-cost path becomes a simple problem if there is no congestion, and all the trips between an O-D pair can be accommodated in one shortest route. However, congestion is an ever-present phenomenon in urban areas. Consequently, the trips between an O-D pair may be using multiple routes, all attempting to minimize their respective travel times. Thus an equilibrium state is achieved. To capture this behavior of the users, the equilibrium principle of traffic assignment needs to be incorporated in the modelling concept. It should, however, be pointed out that some of the existing models do not incorporate this principle. Yet another issue is data requirements. Again, some models have restrictive requirements in this regard that limit their usage.

Thus, there is a need for developing a simpler model to suit the intended application. With the above issues in mind, the specific objectives of this research are set as follows:

1. To formulate and develop a simpler mathematical model that would utilize link traffic volumes to synthesize a trip table that replicates the observed link volumes, when assigned to the network based on the user-equilibrium principle.
2. The developed model must be able to accommodate prior trip table information for the region, if required, to synthesize a table that is also consistent with, or as close as possible to, this prior information.
3. The model must be designed for quick computer execution, so as to be of practical use during real-time diversion situations.

The problem of trip table synthesis from link traffic volumes is defined below.

## 2.4 Problem Definition

Consider a road network with origin and destination zones connected by links representing the roadway sections, and assume that information regarding the traffic volumes during any period of time is known. In addition, assume that these volumes are in a steady state condition. These volumes are the result of trips from origin zones to destination zones, and therefore contain information on the number of trips between various origin-destination pairs. The interest is in extracting this information.

The problem of trip table estimation from traffic counts hinges on the general relationship:

$$v_a = \sum_i \sum_j p_{ij}^a t_{ij}, \quad [2.1]$$

where,

$v_a$  is the volume of traffic on link  $a$ ,

$p_{ij}^a$  is the proportion of trips between origin  $i$  and destination  $j$  using link  $a$ ,

$t_{ij}$  is the number of trips between zone  $i$  and zone  $j$ .

The above equation states that the volume of traffic on a link is equal to the sum of the number of trips between all the origin-destination pairs using this link. The volume  $v_a$  is known through link count data. The parameters  $p_{ij}^a$ 's are dependent on the tripmakers' choice of routes. An assumption needs to be made on this behavior to determine the proportions. If proportions are to be determined independent of the trip table, then assignment techniques such as all-or-nothing, or Dial's (1971)

multipath assignment, can be adopted. However, if congestion effects are to be incorporated in the assignment process, proportions cannot be determined before estimating the trip table, since there is an interdependence between congestion and demand. In such instances, the user-equilibrium principle may be applied. While the proportional assignment assumption simplifies the solution, it is deficient with respect to accuracy in situations when congestion exists. On the other hand, the consideration of congestion effects usually leads to added analytical complexity.

The interest in the above problem is to determine the unknowns, i.e.,  $t_{ij}$ 's, which constitute the cell values of the required trip table. In determining the  $t_{ij}$ 's, different approaches attempt to achieve different chosen objectives, such as the minimization of the total system travel cost, or the maximization of the system entropy.

The various issues related to the above problem are enumerated below.

## **2.4.1 Issues Related to the Problem of Estimating Origin-Destination (O-D) Trip Tables from Link Volumes**

Issues related to the problem of O-D trip table synthesis, including certain general comments made by Willumsen (1978) on various aspects of the problem are relevant here.

### **2.4.1.1 Types of Trips**

The concern in estimating the O-D trip table from traffic counts deals exclusively with car trips. In other words, mass transit trips, walk trips, etc., are not of interest

here. This is evident from the fact that only motor vehicle volumes on network links are considered in estimating the trip table. This is consistent with the overall research goal which is concerned with only highway traffic. It is to be noted that when automatic traffic measuring devices are used, the volume counts also include trucks and other non-autos. However, these vehicles constitute only a small percentage of total volume.

#### **2.4.1.2 Stability of O-D Matrix Over Time**

Another noteworthy point is the stability of the O-D matrix over time. This issue still poses a problem. In other words, as time changes, the O-D trip pattern also changes. However, in this research, the interest is in estimating the trip table for a short period, ideally one hour. This is intended to capture the trip destinations during the time while the incident lasts. Appropriately, assuming steady state conditions, the observed link volumes during the hour preceding the incident can be used in the estimation process.

#### **2.4.1.3 Inconsistency in Link Volumes**

Link volumes are said to be inconsistent, if at an intersection node, the conservation of flow property is not satisfied. In other words, the total volume of traffic flowing into the intersection is not equal to total volume flowing out. The reasons for this could be due to the abstraction of the network, errors in measurement, or due to asynchronous counting. This inconsistency is most likely in real networks. A good model must be able to accommodate this inconsistency.

#### **2.4.1.4 Traffic Assignment**

As stated earlier, a major aspect of the problem of O-D trip table estimation is traffic assignment. This procedure enables the allocation of O-D trips to the different routes in the network. From this allocation, the link volumes can be known. The two general alternative approaches for traffic assignment are: proportional assignment and capacity restrained assignment. In proportional assignment, not all drivers are assumed to "perceive" route costs to be the same. These differences are accounted for in the assignment process. Drivers and route characteristics determine the proportion of tripmakers choosing different near optimum routes. Congestion effects are not considered and hence proportions are treated as independent of link flows. A special case of proportional assignment is the all-or-nothing assignment, where all trips are assigned to the (apparently) cheapest route, independent of traffic volumes. Capacity restrained assignments are more realistic, since they consider the effect of traffic volume and capacity on travel time, and make appropriate adjustments.

Equilibrium Assignment, which is a special case of capacity restrained assignment, takes into account congestion and satisfies Wardrop's (1952) first principle, which states that, "Traffic on a network distributes itself in a way that the travel costs on all used routes between any origin and any destination are equal, while all unused routes have equal or greater costs". To achieve equilibrium assignment, heuristic or mathematical programming techniques can be used. This research work proposes to incorporate user-equilibrium concepts in the modelling approach.

#### **2.4.1.5 Underspecification of the Problem**

The general relationship (equation [2.1]) for trip table estimation translates to one equation for each link of the network with known observed volume. In order to uniquely determine the  $t_{ij}$ 's, there must be as many independent equations as there are  $t_{ij}$ 's. However, in most cases, the number of links will be less than the number of interchanges. Thus, the problem is underspecified. This necessitates assumptions to be made on the trip making behavior. This is usually accomplished by targeting the solution to a known prior trip table.

With this general background of the problem, we review in the next chapter, various existing approaches for estimating a trip table from link volume data.

## **3.0 SURVEY OF APPROACHES FOR ESTIMATING O-D TRIP TABLES FROM TRAFFIC COUNTS**

### **3.1 *Introduction***

The early 1970's saw the dawn of theoretical approaches for estimating O-D trip tables using link traffic counts. The interest in these approaches was kindled due to a need for a shift in planning philosophy from long term to intermediate and short term, required by limitations in budget, time and manpower resources. Since then, different approaches to accomplish this task have evolved, incorporating various desirable features and refinements. Many prominent approaches are nonlinear, and are based on the general framework of trip table estimation utilizing link volumes, as defined in Chapter 2.

Early approaches to this problem relied on linear or nonlinear regression analysis to construct demand models assuming a gravity-type flow pattern, in order to estimate trip table entries. These models, however, required data on zone-specific

variables, like population, employment etc. A later group of models based their estimation of trips on the network traffic equilibrium approach, with the aim of accounting for congestion effects. Yet another group attempted to extract a most likely trip table consistent with link volumes, through entropy maximizing or information minimizing approaches. A more recent group of models, based on statistical estimation approaches, employ statistical techniques to produce future estimates based on prior information. The interest in the problem is continuing, as evidenced by reports of enhancements and refinements to the above approaches.

Willumsen (1978) presents a detailed review of methods for estimating an origin-destination matrix from traffic counts. A comprehensive and more recent review of these models is provided by O'Neill (1987) in her Ph. D. dissertation. Hence, we adopt O'Neill's classification scheme, and confine our remarks to a brief review of the relevant literature, providing details from more recent papers, wherever necessary. For further details and a critical discussion of these models, the reader is referred to O'Neill's dissertation.

### **3.1.1 Types of Models**

O'Neill (1987) classifies the different types of models to estimate trip tables from link volumes, as follows:

1. Parameter Calibration Models
  - a. Linear
  - b. Non-Linear
2. Matrix Estimation Models

- a. LINKOD Type
- b. Maximum Entropy - Minimum Information Type
- c. Statistical Estimation Type

Different model variations have been proposed in the sub-classes of the above categorization. The details of these models are provided in the following sections.

### **3.1.2 Parameter Calibration Models**

#### **3.1.2.1 Conceptual Framework**

The underlying concept in these models is to combine an assignment assumption with a distribution assumption to produce a consistent trip table. If a travel demand model is assumed to be of the form:

$$t_{ij} = f(O_i, D_j, c_{ij}, \beta) \quad [3.1]$$

where,

$O_i, D_j$  are explanatory variables,

$c_{ij}$  is a measure of travel cost or impedance,

$\beta$  is a vector of parameters,

then, by substitution, equation [2.1] (general equation for trip table estimation) takes the form:

$$v_a(\beta) = \sum_i \sum_j p_{ij}^a f(O_i, D_j, c_{ij}, \beta) + e_a \quad [3.2]$$

where  $e_a$  is an error term, and other notation are as defined earlier.

These modelled link volumes are then compared with observed volumes through regression analyses. When applying regression techniques, the aim is to calibrate parameters to minimize the difference between the observed and estimated volumes. In notational form, the intent is to solve a problem of the following form.

$$\min_{\beta} F(v, v(\beta)), \quad [3.3]$$

where  $F$  is some measure of distance between observed and estimated volumes.

These parameters are then substituted in equation [3.1] to estimate the trip matrix.

As cited earlier, these models are either linear or non-linear, based on the regression type employed.

### **3.1.2.2 Linear Parameter Calibration Models**

O'Neill (1987) reviews models formulated by Low (1972), by Overgaard, by Jensen and Neilsen (reviewed in Bendtsen (1974)), by Holm et al. (1976), and by Gaudry and Lamarre (1979). All models except the one by Holm et al. (1976) adopt proportional all-or-nothing assignment, with other differences among the models attributable to variable definition and parameter selection.

Low's (1972) model obtains trip probabilities based on the simple gravity model, according to

$$t_{ij}^n = O_i D_j / c_{ij}^s, \quad [3.4]$$

where,

$O_i, D_j$  are zone specific explanatory variables,

$1 / c_{ij}^s$  is a separation factor with parameter  $s$ ,

$t_{ij}^n$  is a trip probability using definition  $n$  for  $O_i, D_j$ , and  $s$ .

Here,  $O_i, D_j$  and  $s$  can be defined in different ways to yield different trip tables. Each trip matrix is assigned to the network using the all-or-nothing assignment to produce trip probability factor volumes,  $x_n$ . These volumes are then compared to selected existing traffic counts with the aim of calibrating the following multiple regression equation:

$$v_a = c + \sum_i b_i x_i, \quad [3.5]$$

where  $c$  and  $b_i$  are parameters to be estimated. These equations can then be used as a volume forecasting model. In addition to forecasting, the model could also be used for replicating the present pattern of trips.

The models due to Overgaard, and Jensen and Nielsen, when compared to Low's model, offer slight differences with regard to zonal generation coefficients, explanatory variables, etc. Following Jensen and Nielsen's model of iteratively determining the exponent in the gravity model, Holm et al. (1976) further develop this procedure by adopting Smock's (1962) iterative traffic assignment algorithm, thus

incorporating capacity restraints into the solution. Gaudry and Lamarre (1979) developed a simple linear model accounting for heteroscedasticity of residuals. Based on a procedure parallel to Low's, Neumann et al. (1983) have developed areawide, all-purpose trip production rates from traffic counts. The drawback of Low's, and most other methods, is the requirement of estimation of external trips to the study area which are to be subtracted from observed counts.

### **3.1.2.3 Non-linear Parameter Calibration Models**

Robillard (1975) and Hogberg (1976) investigate this type of model. Robillard's model uses the proportional assignment technique and hence does not consider capacity restraints. The problem of estimating an O-D table is shown to be equivalent to a general nonlinear problem. The measure of closeness suggested in choosing an O-D matrix is  $C(v_a, v_a(\beta))$  defined as:

$$C(v_a, v_a(\beta)) = \sum_a (v_a - v_a(\beta))^2, \quad [3.6]$$

where,

$v_a$  is observed flow on arc  $a$ ,

$v_a(\beta)$  is assigned traffic volume on arc  $a$

By employing a distribution assumption, the O-D matrix is then evaluated. A trip table so obtained is viewed as a nonlinear least squares estimate of an O-D matrix. However, this class of method is shown to be equivalent to a linear regression

problem. It is also shown through an example, that the solution is not unique. Consequently, a distribution assumption is used to evaluate the O-D matrix.

Hogberg's (1976) model is a parameter calibration model of the nonlinear regression type, solved by means of a least-squares algorithm to get estimates of trips between origin-destination interchanges. The all-or-nothing assignment is employed here. In this model, the distance between districts, the number of inhabitants, and the number of employees in each district are used as exogenous variables. Only a partial set of link counts is mandatory. Three journey types are used in the gravity-type distribution model. Wills (1977) proposes an alternative nonlinear model for estimating an O-D matrix. This model accommodates a wider range of variables.

### 3.1.3 Matrix Estimation Models

#### 3.1.3.1 General

Here, trip tables are extracted directly from known information without calibrating parameters as in parameter calibration models. Assumptions on how trips are distributed are utilized to overcome the problem of underspecification that arises when the number of known volume counts on links are fewer than the number of unknown trip interchanges.

The general relationship for the determination of O-D trip table entries as shown in equation [2.1] is used here to form a feasible region over which a functional form

of a distribution assumption is optimized. O'Neill gives a general framework for these types of models, as below:

$$\min F(T, T^*), \quad [3.7]$$

Subject to:

$$v_a = \sum_i \sum_j p_{ij} t_{ij} \quad [3.7a]$$

where,  $T$  represents the matrix of interest, and  $T^*$  represents a reference matrix.

The reference matrix may be a target matrix, an outdated matrix, or even a structural matrix. Additional constraints could be added to the above problem, if required. In some matrix estimation approaches, in order to obtain a feasible solution, the constraints must be consistent. This is violated when the link volumes do not conserve flow. For various reasons, it is very likely that observed link volumes may not conserve flow. However, ways of eliminating inconsistencies in traffic counts have been provided by Van Zuylen and Willumsen (1980), Van Zuylen (1981), Van Zuylen and Branston (1982), and Barbour and Fricker (1989). The differences among the matrix estimation models can be attributed to the source of their derivation - whether derived as mathematical programming optimization models or developed from principles of econometric analysis. Another source of difference is the combination of distributional assumption, assignment assumption, and measure of distance employed. Data on target or prior trip table and observed link volumes on some or all network links are required by these models.

### **3.1.3.2 Types of Models**

O'Neill (1987) classifies papers relating to matrix estimation techniques into three groups as follows:

1. Group I: Those dealing with LINKOD and related models
2. Group II: Those that deal with modelling approaches employing minimum information/maximum entropy theory
3. Group III: Those that employ statistical estimation techniques, covering the most recent research.

The primary aim in LINKOD-type models is to derive a solution satisfying equilibrium assignment concepts. Maximum entropy/minimum information models, on the other hand, emphasize distribution assumptions. While trip distribution and traffic assignment are treated separately in these models, Erlander et al. (1979), and Fisk and Boyce (1983) propose combined distribution and assignment models. The Group III models attempt to consider the stochastic nature of the data in future estimates. Prior information in the form of a trip table is used by all the above models to guide the solution.

**Group I: LINKOD and Related Models:** A comprehensive review of the development of this group of models, and a critical discussion of the various related aspects is provided by O'Neill. This group of models is suitable for congested area analysis. In these models, the solution tends to approach the given prior information, which is made the target. However, adjustments are made to reproduce observed link

volumes. Here, the main interest is in extracting an equilibrium trip table, that is as close to the target as possible. Presented below are the formulations proposed by Nguyen, that form the bases for this type of models.

Nguyen (1977) exploits Wardrop's user-equilibrium principle for route choice and formulates a deterministic network equilibrium approach. He proposes two models stated as optimization problems.

In the first model, an O-D trip matrix is obtained by solving the following problem:

$$\min F(v) = \sum_a \int_0^{v_a} f_a(x) dx \quad [3.8]$$

subject to :

$$t_{ij} - \sum_r h_{rij} = 0, \quad \forall i, j \quad (\text{zonal conservation of flow}) \quad [3.8a]$$

$$h_{rij} \geq 0, \quad \forall r, i, j \quad (\text{nonnegativity constraint}) \quad [3.8b]$$

$$t_{ij} \geq 0, \quad \forall i, j \quad (\text{nonnegativity constraint}) \quad [3.8c]$$

$$v_a = \sum_r \sum_i \sum_j d_{arij} h_{rij}, \quad \forall a \quad (\text{analogous to general relationship}) \quad [3.8d]$$

$$\sum_a f_a(v_a) v_a - \sum_i \sum_j u_{ij} t_{ij} = 0 \quad (\text{obs. total travel cost} = \text{est. total travel cost}) \quad [3.8e]$$

where,

$f_a(x)$  = volume - delay function for link  $a$ ,

$t_{ij}$  = travel demand between origin  $i$  and destination  $j$ ,

$h_{rij}$  = flow on route  $r$  connecting origin  $i$  and destination  $j$ ,

$d_{arij} = \begin{cases} 1 & \text{if link } a \text{ is on route } r \text{ between } i \text{ and } j, \\ 0 & \text{otherwise,} \end{cases}$

$v_a^*$  = observed link flow, and

$u_{ij}^*$  = observed inter-zonal accessibility (travel time on any used route between zone  $i$  and zone  $j$  due to the equilibrium assumption).

Here, the observed link flows are explicitly considered. Giving a modified Frank-Wolfe algorithm for solving the problem, Nguyen suggests that this model is more appropriate for small networks, where observed traffic counts on all links can be easily obtained. Where this is not the case, or where large number of origins and destinations may render the solution procedure less efficient, he proposes an alternate formulation.

The alternative formulation, as given below, considers only observed inter-zonal accessibilities, thus relaxing the requirement on the need for all link counts.

$$\min F(v, T) = \sum_a \int_0^{v_a} f_a(x) dx - \sum_i \sum_j u_{ij}^* t_{ij}, \quad [3.9]$$

subject to :

Constraints in equations [3.8a], [3.8b], [3.8c], [3.8d]

This model is suitable for large networks, and there is substantial reduction in input data, as claimed by Nguyen. Based on tests on a small problem, the author concludes that both the above models produce trip tables that closely reproduce observed volumes when assigned to the network using user-equilibrium principle. A major deficiency of this approach is that the problem can have more than one solution that can reproduce the traffic patterns. This necessitates the use of a distribution assumption to identify the most likely trip table among the alternative solutions.

Nguyen's theoretical model was operationalized during the course of a Federal Highway Administration project in which the LINKOD system of models (Turnquist and Gur, 1979; Gur et al., 1980) were developed. The LINKOD system is comprised of two major components: SMALD and ODLINK. SMALD (Kurth et al., 1979), a small area trip distribution model, determines a trip table for a sub-area. This table is used to overcome the underspecification problem of Nguyen's formulation. It is used as the initial (target) table, which is corrected by the ODLINK model, such that the corrected table when assigned using equilibrium principle, will replicate observed volumes on network links. An alternate approach for selecting the most likely trip table among optimal solutions is proposed by Nguyen (1984).

Turnquist and Gur (1979) and Gur et al. (1980) propose an enhanced and efficient iterative heuristic solution procedure for Nguyen's second model, again employing a modified Frank-Wolfe algorithm. This procedure, incorporated in the LINKOD system, also includes a heuristic method for correcting the initial trip table. Gur et al. (1980) have conducted tests of the LINKOD system on a sample network. The same network has been used in Chapter 6 of this research work for a comparison of models. The authors of the LINKOD model conclude that the algorithm has many desirable properties, including the capability to move towards a reasonable solution even when

the starting table does not contain information about the solution table. Also, the table produced by the model approximates observed flows very closely. The computer effort required is reported to be only modest. The LINKOD model has been extensively tested and verified by Han et al. (1981), Han and Sullivan (1983), and Dowling and May (1984). A modification of the LINKOD model to handle partial traffic counts has been considered by many authors, including Nguyen (1984). The issue is identified as a case for further research. Most researchers of LINKOD models conclude that the performance of the model depends primarily on data requirements.

**Group II: Minimum-Information/Maximum Entropy Models:** The above approaches for estimating a trip table from link counts, like the parameter calibration models and the network equilibrium based approaches, force the solution trip table to conform to a gravity type pattern, or cause them to be as close to a prior trip table as possible. Van Zuylen and Willumsen (1980) criticize this by saying that these approaches do not make full use of the information contained in the link volumes. Since prior trip table information is used by other approaches to also take care of the underspecification problem, Van Zuylen and Willumsen suggest that this could be solved by introducing minimum external information. Following this idea, the authors have put forward two concepts - information minimization and entropy maximization approaches. The original models in these categories were proposed by Van Zuylen (1978) and Willumsen (1978), respectively. In the information minimizing approach, an attempt is made to choose a trip table that adds as little information as possible to the knowledge contained in the general equation [2.1] for trip table estimation from link counts.

Based on the theory of information minimization, Van Zuylen (Van Zuylen and Willumsen, 1980) has derived a multiproportional model for estimating a trip table, as follows:

$$\min F(v) = \sum_a \sum_i \sum_j t_{ij} p_{ij}^a \ln [(t_{ij} v_a^*) / (v_a t_{ij}^*)] \quad [3.10]$$

subject to the general relationship for trip table estimation, namely, Eqn. [2.1],

where,

$t_{ij}^*$  = prior (or old) trip matrix,

$$v_a^* = \sum_i \sum_j t_{ij}^* p_{ij}^a.$$

and other notation are as defined earlier.

On similar lines, Willumsen (Van Zuylen and Willumsen, 1980) proposed a maximum entropy approach to solve the problem. This approach is more popular, and is detailed below. The method is based on Wilson's (1970) application of the concept of entropy to the O-D trip matrix. Here, the most likely trip matrix is defined as the one having the greatest number of micro-states associated with it. Attempting to maximize the number of ways of selecting a trip matrix, Willumsen formulates the problem as:

$$\max F(T) = - \sum_i \sum_j (t_{ij} \ln t_{ij} - t_{ij}) \quad [3.11]$$

subject to eqn [2.1], the general O-D matrix estimation relationship.

Where information contained in a prior matrix is to be used, the above formulation becomes,

$$\max F(T, T^*) = - \sum_i \sum_j t_{ij} (\ln t_{ij} / t_{ij}^* - 1), \quad [3.11a]$$

subject to eqn [2.1].

The derived table would be the most likely that is consistent with information contained in the link flows.

Both the above approaches are shown to reduce to a multiproportional problem. In particular, the maximum entropy approach reduces to solving the following optimality conditions:

$$t_{ij} = t_{ij}^* \prod_a X_a^{p_{ij}}, \quad [3.12a]$$

where,

$$X_a = \left( \sum_{ij} t_{ij} \right)^{1/L} \cdot e^{-\lambda_a}, \quad [3.12b]$$

and where  $L$  denotes the number of counted links, and  $\lambda_a$  is the Lagrange multiplier corresponding to the count on link  $a$  constraint. Van Zuylen and Willumsen (1980)

also give an algorithm for solving the above problem [3.12], based on Murchland's (1977) algorithm for the multiproportional problem. It is further indicated that the convergence of the problem has not been satisfactorily proved. Counts on all the links are not necessary. However, a complete set of counts is expected to yield better results. It is to be noted that both the above formulations require information on link usage proportions.

O'Neill (1987) summarizes the conditions to be satisfied by the maximum entropy model in order for an estimated trip table to reproduce observed volumes fully, as: (1) consistency of link volumes (flow conservation), (2) consistency of prior trip table with observed flows and route choice proportions, and (3) consistency of route choice assumption with observed flows.

Many researchers (Hall et al. (1980), Van Zuylen (1981), Van Vliet and Willumsen (1981), Willumsen (1982), Bell (1983), and Nguyen (1984) etc.) have conducted tests or have proposed improvements on this type of model. Of particular interest is the attempt to consider the effects of congestion, through equilibrium assignment. Willumsen (1982) proposed and tested a heuristic model that includes the equilibrium principle. Bromage (1988), while at Central transportation Planning Staff (CTPS), Boston, programmed the maximum entropy model, incorporating a capacity restraint procedure for the assignment step. This program was enhanced recently by Beagan (1990), also of CTPS, to include an equilibrium assignment option, as proposed by Hall et al. (1980). These two improved models were in fact used in this research to test the maximum entropy approach. The results are reported in Chapter 6. Hamerslag and Immers (1988) suggest some possibilities for improvement for this group of models. Fisk (1988) has shown how to combine the maximum entropy model and the user-optimal assignment into one problem. It is also shown by Fisk (1989), that the network equilibrium approach, the maximum entropy approach, and

the combined distribution-assignment formulation can be expected to produce the same results under congested network conditions, when observed link volumes satisfy equilibrium flow pattern. In a recent approach, Brenninger-Gothe, Jornsten, and Lundgren (1989) present a multiobjective programming formulation using entropy function, for estimating O-D matrices, where efficient points are sought which compromise between the separation of the solution from the observed traffic counts versus that from the prior target O-D matrix. As concluded by O'Neill (1987), the maximum entropy models require further verification and refinements.

**Group III: Statistical Estimation Models:** The models in this category attempt to estimate trip tables directly from prior estimates, using statistical techniques. The two techniques employed to achieve the objectives in this group of models are the Bayesian inference methods, and least squares estimation techniques. These models consider the stochastic nature of data.

Criticizing the maximum entropy or minimum information approaches as being prone to giving little weight to the prior information, Maher (1983) points out the need for incorporating the measure of degree of belief in prior estimates. Based on Bayesian statistical inference, he proposes a method that allows the flexibility of placing different degrees of belief in the prior information, for estimating a posterior trip table. This method, however, is based on a proportional assignment assumption, and requires that these proportions be known beforehand. No detailed tests are known to have been conducted on this method. In the category of least squares estimation approaches, Carey et al. (1981), McNeill and Hendrickson (1985), and Cascetta (1984) have proposed model formulations. Cascetta proposes a generalized least squares estimator for the trip table matrix, by combining the estimation with traffic counts, through an assignment model. The hypothesis here is that the direct

or model estimators used for this purpose are unbiased, and that there are negligible misspecification errors in the assignment model. Tests on real data have not been carried out.

Thus, the advantage of Group III models lies in their consideration of stochastic nature of data and the problem. However, these models need to be further tested, especially on real data. Also, the equilibrium assignment principle needs to be incorporated into these models, to account for congestion. O'Neill (1987) has combined the equilibrium assignment concept, that is applicable in urban analysis, with the incorporation of stochastic nature of data, into the trip table estimation process to produce a heuristic model. Here, the equilibrium proportion estimation method, and the assignment model have been combined with the trip matrix estimation process through statistical techniques. The two steps, one associated with each component technique, are iteratively solved to get an acceptable solution. O'Neill has concluded based on tests on a small network, that the model cannot be rejected, but that further tests need to be conducted.

### **3.2 *Summary of Literature Review***

The above review of different approaches to synthesize a trip table from link counts has exposed the complexities associated with the problem, and the attempts by various approaches to overcome them. However, the models reviewed have certain inherent weaknesses that render them either inappropriate or questionable for use in the context of real-time diversion planning. The reasons for these are stated below.

Parameter calibration models use zone-specific variables such as information on population, employment, etc., in formulating the demand model. All of these models employ all-or-nothing traffic assignment, with the exception of one model. The use of these models is unattractive, since they require considerable data, which, furthermore, may become outdated as changes in land use take place. This raises the issue of reliability of data for use in the model. Another drawback of parameter calibration models is that they fail to incorporate the equilibrium assignment principle, which is relevant in congestion related contexts.

In the category of matrix estimation models, although the LINKOD type models incorporate the desired equilibrium assignment concept, their nonlinear nature renders them infinitely convergent, leading to the issue of the computational effort required for deriving acceptable solutions. The minimum information/maximum entropy models, on the other hand, pose restrictions on data requirements (such as consistency of flows), in addition to being theoretically complex. The maximum entropy model also needs refinements for incorporating the equilibrium principle. The computational effort required may also pose restrictions in its use for real-time implementation.

The statistical estimation models possess the desirable quality of considering the stochastic nature of the data and the problem. However, they have not been well-tested, and further work needs to be done in accounting for congestion considerations.

Thus, there is a need for developing a simpler approach, that will accomplish the problem objectives, without demanding excessive computational effort. With this motivation, a linear programming model is proposed in the next chapter. In contrast to *nonlinear* network equilibrium approaches, the proposed model is a *linear programming model*, but one that also employs the nonproportional-assignment

assumption, and that finds a user-equilibrium solution which reproduces the observed link flows, whenever such a solution exists, and is as close to a prior trip table as possible.

## **4.0 LINEAR PROGRAMMING APPROACH**

### ***4.1 The Approach***

This chapter is concerned with a linear programming-based optimization approach for solving the problem of determining a trip table using traffic counts on links. The philosophy of the approach rests on the observation that user equilibrium solutions, if they exist for the given problem, can be determined through an appropriate system oriented model. In other words, given a network, origin-destination (O-D) trips are estimated by minimizing a weighted sum of travel times of all motorists between different origin-destination points, subject to constraints that link volumes arising from estimated O-D trips replicate observed volumes. This problem can be formulated as a linear program, as follows. The approach is further extended to accommodate prior trip table information to guide the solution trip table.

## 4.2 Notation and Model Formulation

Given an urban road network for a particular region, let  $G(N, A)$  represent the underlying digraph. Here,  $N$  is the set of nodes representing either traffic intersection points where flow is conserved, or zones where trips are generated and/or where trips terminate, and  $A$  denotes the set of corresponding directed links or arcs representing the roadways existing between designated pairs of nodes. Let  $OD$  denote the set of origin-destination (O-D) pairs which comprise the trip table to be estimated, where  $O$  is the set of possible source or origin nodes, and  $D$  is the set of potential sink or destination nodes. It is to be noted that a node representing a given zone is typically both an origin as well as a destination node.

The problem addressed requires the estimation of flows  $x_{ij}$  between each O-D pair  $(i, j) \in OD$  comprising the trip table, given observed (integral) link flows  $\bar{f}_a$  on each link  $a \in A$ . Accordingly, consider an *implicit* enumeration of all possible O-D paths. Let  $p_{ij}^k$ ,  $k = 1, \dots, n_{ij}$ , represent all the paths between each O-D pair  $(i, j) \in OD$ , where  $p_{ij}^k$  is a vector which has one component for each link  $a \in A$ , this component being unity if the corresponding link belongs to the particular path, and being zero otherwise. Let  $x_{ij}^k$  represent the flow on path  $p_{ij}^k$  for each  $k = 1, \dots, n_{ij}$ ,  $(i, j) \in OD$ , and let  $x$  denote the vector of components  $x_{ij}^k$ . Then, in order to determine the flow  $x$  which reproduces the vector  $\bar{f}$  of the observed link counts ( $\bar{f}_a$ ,  $a \in A$ ), a solution to the following system needs to be determined.

$$\sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} (p_{ij}^k) x_{ij}^k = \bar{f}, \quad x \geq 0 \quad [4.1]$$

It is to be noted that the O-D flows  $x_{ij}$  are related to the path-decomposed flows  $x_{ij}^k$  via the relationship,

$$x_{ij} = \sum_{k=1}^{n_{ij}} x_{ij}^k, \quad \text{for each } (i, j) \in OD \quad [4.2]$$

Based on the observed flows  $\bar{f}_a$ ,  $a \in A$ , a corresponding link impedance  $\bar{c}_a \equiv c_a(\bar{f}_a)$ , can be computed, where  $c_a(\cdot)$  is a link travel time/cost function of the following form, as suggested by the Bureau of Public Roads (BPR) (1964):

$$c_a(v_a) = c_a^F \left[ 1 + 0.15 \left( \frac{v_a}{u_a} \right)^4 \right] \quad [4.3]$$

where,

$c_a(v_a)$  = travel time/cost on link  $a$  with volume  $v_a$

$c_a^F$  = free-flow travel time/cost on link  $a$

$v_a$  = flow on link  $a$

$u_a$  = flow capacity of link  $a$

Since it is only needed to consider links  $a \in A$  for which  $\bar{f}_a > 0$ , and since multiplying  $\bar{c}_a$  for all  $a \in A$  by a constant leaves the problem invariant, we assume for the sake of development that  $\bar{c}_a \geq 1$  and is integral for each  $a \in A$ . Let  $\bar{c}$  denote the vector of components  $\bar{c}_a$ ,  $a \in A$ .

Now, if the observed flow pattern indeed represents a network user-equilibrium solution corresponding to some interchange of traffic between the designated O-D pairs, then by Wardrop's (1952) (first) principle, all the routes between any O-D pair

which have positive flows should have equal travel costs, and this cost must not exceed the travel cost on any other unused route between this O-D pair. Notationally, let  $c_{ij}^k \equiv \bar{c} \cdot p_{ij}^k$  denote the impedance or cost on route  $k$  between O-D pair  $(i, j)$  for each  $k = 1, \dots, n_{ij}$ ,  $(i, j) \in OD$ , and let  $c_{ij}^* = \min\{c_{ij}^k, k = 1, \dots, n_{ij}\}$ . Furthermore, let:

$$K_{ij} = \{k \in \{1, \dots, n_{ij}\} : c_{ij}^k = c_{ij}^*\} \quad \text{and} \quad \bar{K}_{ij} = \{1, \dots, n_{ij}\} - K_{ij} \quad [4.4]$$

Then according to Wardrop's principle, if an equilibrium solution exists which reproduces the observed flows, then it should be able to find a solution to [4.1] which satisfies the condition that  $x_{ij}^k > 0$  only if  $k \in K_{ij}$ , for each  $(i, j) \in OD$ . With this motivation, the following linear programming model may be considered.

$$\text{Minimize} \quad \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} \hat{c}_{ij}^k x_{ij}^k \quad [4.5a]$$

$$\text{Subject to} \quad \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} (p_{ij}^k) x_{ij}^k = (\bar{f}) \quad [4.5b]$$

$$x \geq 0 \quad [4.5c]$$

where for each  $(i, j) \in OD$ ,

$$\hat{c}_{ij}^k = \begin{cases} c_{ij}^k & \text{if } k \in K_{ij} \\ M_1 c_{ij}^k & \text{if } k \in \bar{K}_{ij} \end{cases} \quad [4.5d]$$

and where  $M_1 > 1$  is some scalar parameter. The principal feature of this linear program [4.5] is stated by the following result.

Lemma 1:

Let  $\bar{C}_{\text{total}} \equiv \sum_{a \in A} \bar{c}_a \bar{f}_a$  represent the total observed system cost. Then, the optimal objective value of the linear program [4.5] is at least  $\bar{C}_{\text{total}}$ . Moreover, a solution  $x^*$  represents a user equilibrium solution which reproduces the observed flow vector  $\bar{f}$  if and only if  $x^*$  solves the linear program [4.5] with optimal objective value  $\bar{C}_{\text{total}}$ .

Proof:

For any feasible solution  $x$  to the linear program [4.5],

$$\sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} c_{ij}^k x_{ij}^k = \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} (\bar{c} \cdot p_{ij}^k) x_{ij}^k = \bar{c} \cdot \bar{f} \equiv \bar{C}_{\text{total}} \quad [4.6]$$

Noting [4.5d] and [4.6], it follows that the optimal value of the linear program [4.5] is at least  $\bar{C}_{\text{total}}$ . Moreover, the objective value is  $\bar{C}_{\text{total}}$  at an optimal solution  $x^*$  if and only if  $(x^*)_{ij}^k = 0$  for all  $k \in \bar{K}_{ij}$ , for each  $(i, j) \in OD$ , and noting Wardrop's

principle, this holds if and only if  $x^*$  is a user equilibrium solution that is feasible to the linear program [4.5]. This completes the proof.  $\square$

Remark 1:

In the light of Lemma 1, a value of  $M_1 = 2$  can be assumed in [4.5d].

Remark 2:

Consider any feasible solution  $x$  to [4.5], and let  $C_1(x)$  and  $C_2(x)$  represent the total contributions to the system cost with respect to the cost coefficients  $c_{ij}^k$  for the routes  $k \in K_{ij}$  and  $k \in \bar{K}_{ij}$ , respectively, over all O-D pairs  $(i, j) \in OD$ . Then,  $C_1(x) + C_2(x) \equiv \bar{C}_{\text{total}}$ , while the objective function [4.5a] attempts to minimize  $C_1(x) + M_1 C_2(x)$ . Hence, if there exists a feasible solution  $x^*$  such that  $C_1(x^*) = \bar{C}_{\text{total}}$  and  $C_2(x^*) = 0$ , then as in Lemma 1, the linear program [4.5] will determine such a solution since  $M_1 > 1$ . On the other hand, if such a solution does not exist, then the linear program will attempt to allocate as much of the total observed cost  $\bar{C}_{\text{total}}$  as possible to the term  $C_1(x)$  representing total cost on the routes in  $K_{ij}$ . By varying the parameter  $M_1$ , it is possible to manipulate, to an extent, the distribution of flows (as opposed to costs) on the routes in  $K_{ij}$  versus that on the routes  $\bar{K}_{ij}$ .

Until now, it was assumed that the observed flow data  $\bar{f}$  is consistent with respect to [4.1]. However, due to errors in measurement and due to approximations in the network representation, there might be internal inconsistencies which might render [4.1], and therefore the linear program [4.5], infeasible. While some

preprocessing may be done to manipulate the data in order to make [4.1] feasible, as in Van Zuylen and Branston (1982) for example, it is more appealing to let the model and the optimization process itself accommodate this feature. Therefore, two vectors of nonnegative artificial variables  $y^+$  and  $y^-$  are introduced, with respective components  $y_a^+$  and  $y_a^-$ , for each  $a \in A$ , into the constraints [4.5b], in order to permit positive or negative deviations in the link flows. Accordingly, the objective function is penalized with the term  $M e \cdot (y^+ + y^-)$ , where  $e$  is a vector of  $|A|$  ones, and  $M$  is a suitable penalty parameter, whose value is addressed below. This revises the linear programming model (LP) as follows.

$$LP: \text{Minimize} \quad \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} \hat{c}_{ij}^k x_{ij}^k + M e \cdot (y^+ + y^-) \quad [4.7a]$$

$$\text{Subject to} \quad \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} (p_{ij}^k) x_{ij}^k + (y^+ - y^-) = (\bar{f}) \quad [4.7b]$$

$$x \geq 0, \quad y^+ \geq 0, \quad y^- \geq 0 \quad [4.7c]$$

Henceforth, we will denote  $y \equiv y^+ + y^-$ .

#### Lemma 2:

Assuming that the linear program LP has a feasible solution  $(x, y)$  with  $y \equiv y^+ + y^- = 0$ , let the following alternative values for  $M$  be considered, where  $\bar{c}_a = \text{maximum } \{\bar{c}_a : a \in A\}$ :

$$(i) \quad M \geq 1 + \bar{c}_* + (|A|!) \bar{C}_{\text{total}} \quad [4.8a]$$

$$(ii) \quad M \geq 1 + \bar{c}_* + \bar{C}_{\text{total}} \quad [4.8b]$$

$$(iii) \quad M \geq 1 + \bar{c}_* \quad [4.8c]$$

Then, if  $M$  is selected as in [4.8a], it is ensured that  $y^* = 0$  in any optimal solution  $(x^*, y^*)$  to LP, while if  $M$  is selected as in [4.8b],  $e \cdot y^* < 1$  in any optimal solution  $(x^*, y^*)$  to LP. Moreover, if there exists a feasible solution  $(x, y)$  to the LP with  $y = 0$  and with  $x$  conforming to a user equilibrium, then if  $M$  is chosen as in [4.8c], a solution is optimal to LP if and only if it is such an equilibrium solution.

Proof:

First of all, note that LP is feasible with nonnegative objective values, and hence has an optimal solution. Moreover, for any basic feasible solution (BFS) to LP with  $y_a > 0$  for some  $a \in A$ , we have upon applying Cramer's rule to the basic system of equations that  $y_a \geq 1/(\det B)_{\max}$  where  $(\det B)_{\max}$  is the maximum absolute value of the determinant for any basis of the equation system [4.7b] (Bazaar et al., 1990). Since any basis of [4.7b] is a square matrix of size  $|A|$  with zero-one elements, by the definition of determinants, it readily follows that  $(\det B)_{\max} \leq |A|!$ . Hence,

$$y_a \geq \frac{1}{|A|!}, \text{ whenever } y_a \geq 0 \text{ at any BFS to LP.} \quad [4.9]$$

Now, for any feasible solution  $(x, y)$  to LP, the objective value in [4.7a] is given by

$$\begin{aligned}
\hat{c} \cdot x + M e \cdot y &\geq \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} c_{ij}^k x_{ij}^k + M e \cdot y = \bar{c} \cdot \left[ \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} p_{ij}^k x_{ij}^k \right] + M e \cdot y \\
&= \bar{c} \cdot [\bar{f} - y^+ + y^-] + M e \cdot y \geq \bar{C}_{\text{total}} + (M e - \bar{c}) \cdot y \\
&\geq \bar{C}_{\text{total}} + (M - \bar{c}) e \cdot y \quad [4.10]
\end{aligned}$$

Hence, if  $M$  is given by [4.8a], then for any basic feasible solution to LP with  $y_a > 0$  for some  $a \in A$ , from [4.8a], [4.9], and [4.10], the objective value (strictly) exceeds  $2\bar{C}_{\text{total}}$ . Similarly, if  $M$  is given by [4.8b], then for any (basic) feasible solution to LP with  $e \cdot y \geq 1$ , from [4.10] it follows that the objective value (strictly) exceeds  $2\bar{C}_{\text{total}}$ . However, for any feasible solution  $(x, y)$  to LP with  $y = 0$ , the objective value in [4.7a] is no more than  $2\bar{C}_{\text{total}}$  noting [4.5d], [4.6], and Remark 1. Hence, since the optimal objective value of LP must be lesser than or equal to  $2\bar{C}_{\text{total}}$  under the hypothesis of the lemma, the proof of the second part follows from the foregoing argument.

Finally, it is to be noted that any solution  $(x, y)$  with  $y = 0$  and with  $x$  conforming to a user equilibrium solution has an objective value of  $\bar{C}_{\text{total}}$  in LP. Moreover, the first inequality in the string [4.10] is strict if the feasible solution  $(x, y)$  to LP has  $x_{ij}^k > 0$  for any  $k \in \bar{K}_{ij}$  and  $(i, j) \in OD$ , while  $(M - \bar{c}) e \cdot y \geq 0$  with the inequality being strict if  $y_a > 0$  for any  $a \in A$ . Hence, the objective value of any feasible solution  $(x, y)$  to LP with  $y \neq 0$  or with  $x$  not conforming to an equilibrium solution strictly exceeds  $\bar{C}_{\text{total}}$ . This completes the proof.  $\square$

Remark 3:

For computational purposes, the value of  $M$  as given by [4.8b] is recommended. Experience shows that this value provides the same practical effect as that guaranteed theoretically by the value [4.8a] in Lemma 2, while the value [4.8a] is too large to be computationally usable in practice. Moreover, if it is known *a priori* that there exists a user-equilibrium solution feasible to [4.1], then the value of  $M$  as suggested by [4.8c] may be used.

Remark 4:

From Remark 2, the objective function in [4.7a] is given by

$$C_1(x) + M_1 C_2(x) + M(e \cdot y) \quad [4.11]$$

This function is therefore comprised of three objective terms  $C_1(x)$ ,  $C_2(x)$ , and  $e \cdot y$ , aggregated with suitable coefficients in a nonpreemptive fashion, while Lemmas 1 and 2 enforce certain preemptive priorities such as feasibility to [4.1] with priority one (Lemma 2), and a user-equilibrium solution with priority two (Lemma 1), if such solutions exist. Of course, the user-equilibrium interpretation is accessible in the linear programming framework only when the observed flows, and hence the observed impedances, are reproduced. Dafermos and Sparrow (1969) present an analysis in a similar spirit. By varying the parameters  $M_1$  and  $M$  in the formulation of LP, one can manipulate the foregoing effects to derive various O-D trip tables.

### 4.3 Modifications for Accommodating Prior Trip Tables

The model formulated thus far is mainly concerned with determining an O-D trip table that conforms to a user-equilibrium solution which reproduces the link flows to as close an extent as possible. The principal advantage of the above procedure is that this task can be accomplished through a purely linear programming based approach, by adopting an appropriate column generation technique using shortest path sub-problems as detailed in Chapter 5. However, if there exist alternative optimal O-D trip tables to the linear programming model LP, then there is no specified mechanism for discriminating among these, as yet. Moreover, by the nature of extreme point optimal solutions to LP, no more than  $|A|$  O-D paths that carry positive flows are identified by the foregoing method, leading to an unduly sparse O-D matrix. In order to rectify these shortcomings, while preserving the advantages of the linear programming approach, the proposed methodology is extended as follows.

Suppose that a partial prior (target) trip table (TT) is specified with associated O-D flows  $Q_{ij} > 0$  for  $(i, j) \in \overline{OD} \subseteq OD$ , where  $\overline{OD}$  might represent some significant or key O-D pairs. The corresponding modification LP(TT) of LP given in [4.7] is then as follows.

$$LP(TT): \text{Minimize} \quad \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} \hat{c}_{ij}^k x_{ij}^k + M \epsilon \cdot (y^+ + y^-) + M_\sigma \sum_{(i,j) \in \overline{OD}} (Y_{ij}^+ + Y_{ij}^-) \quad [4.12a]$$

$$\text{Subject to} \quad \sum_{(i,j) \in OD} \sum_{k=1}^{n_{ij}} (p_{ij}^k) x_{ij}^k + (y^+ - y^-) = (\bar{f}) \quad [4.12b]$$

$$\sum_{k=1}^{n_{ij}} x_{ij}^k + (Y_{ij}^+ - Y_{ij}^-) = Q_{ij} \quad \forall (i, j) \in \overline{OD} \quad [4.12c]$$

$$x \geq 0, \quad y^+ \geq 0, \quad y^- \geq 0, \quad Y^+ \geq 0, \quad Y^- \geq 0 \quad [4.12d]$$

Here, for each  $(i, j) \in \overline{OD}$ , the deviation of the O-D flow  $x_{ij} \equiv \sum_{k=1}^{n_{ij}} x_{ij}^k$  from the target trip table value  $Q_{ij}$  is measured by the difference of two nonnegative ("artificial") variables  $Y_{ij}^+$  and  $Y_{ij}^-$ . The effective absolute value  $(Y_{ij}^+ + Y_{ij}^-)$  of this deviation is penalized in the objective function via a penalty parameter  $M_\sigma \geq 0$ . The value chosen for  $M_\sigma$  must naturally tradeoff the penalty imposed for deviations from the targeted trip table values with the remainder of the objective function. Since the parameter  $M$  penalizes similar deviations of the accounted link flows from the observed ones, the value of  $M_\sigma = \sigma M$ , where  $0 \leq \sigma \leq 1$  is recommended. Hence, we can select a value of  $\sigma \in [0, 1]$  to reflect the relative degree of importance in minimizing the trip table deviations versus the link flow deviations, from unimportant ( $\sigma = 0$ ) to equally important ( $\sigma = 1$ ). Also, note that the additional constraints [4.12c] admit additional basic variables at an extreme point optimum, thereby possibly admitting additional nonzero O-D flows.

#### Remark 5:

In some situations, the interest may be in the preemptive priority problem of matching the prior trip table O-D flows as closely as possible among the alternative optimal solutions to LP. This can be handled by first finding the optimal objective

function value  $v^*$  for LP. Then, the problem of minimizing  $\sum_{(i,j) \in OD} (Y_{ij}^+ + Y_{ij}^-)$  subject to the constraints of LP(TT), in addition to the constraint that restricts the objective function value to be lesser than or equal to  $v^*$ , can be solved.

The theory and algorithm developed for Problem LP extends readily for handling Problem LP(TT). Lemma 1 holds identically, and the value of  $M_1 = 2$  can still be used. Moreover, by modifying [4.10] to include the new penalty term introduced into the objective function, it can be shown similar to Lemma 2 that if there exists a feasible solution to LP(TT) with all artificial variables equal to zero, then by selecting  $M_\sigma = M (\sigma = 1)$  according to [4.8b], the sum of all artificial variables will be less than unity at optimality. Furthermore, the statement in Lemma 2 concerning the recovery of a user equilibrium solution by using [4.8c], if one exists, along with  $\sigma \geq 0$  sufficiently small, continues to hold. Hence, the selection of a value of  $M$  as in Remark 3 is again recommended.

The intent here, therefore, is in determining the  $x_{ij}^k$ 's (O-D trips) that will approach user-equilibrium values, replicate observed flows on links, and concur with the specified prior trip table values to as much of an extent as possible. As can be seen from the formulation, each link has a constraint, which essentially says that sum of different O-D flows passing through that link must equal the observed volume on the link. In addition, there is a constraint for each prior trip table value supplied. These constraints attempt to ensure that the total sum of flows ( $x_{ij}^k$ 's) for each  $(i, j) \in \overline{OD}$  is equal to or as close to  $Q_{ij}$  as possible. The formulation can be better illustrated by considering the following simple, hypothetical network.

## 4.4 Illustration of Model Formulation

For the sake of illustrating the model formulation, consider the three zone, three link example shown in Figure 6. Let it be assumed that trips to a zone can pass via another zone. Consider trips from zone 1. These trips may be destined to zone 2 or to 3, or to both of them. For trips from 1 to 3, two different paths can be utilized, namely, the direct path of 1 to 3 through link 2, or the path passing through zone 2 via links 1 and 3. The cost (travel time) of these paths is 4 minutes each. Similarly, all feasible interchanges and their travel times (weighted as per equation [4.5d]) may be identified. In addition to forming the link constraints for links with known link volumes, a prior trip table information may be used for different O-D pairs to guide the solution trip table to conform as closely as possible to the prior table. Alternatively, instead of utilizing prior information for all O-D pairs, only key pairs may be identified for guiding the solution. The corresponding trip table constraints may then be written. In the example illustration, let O-D interchanges 1-3 and 2-3 be the guiding cells. The linear programming formulation LP(TT) can now be written as follows.

$$\begin{aligned} \text{Minimize } & 2x_{12}^+ + 4x_{13}^+ + 4x_{23}^+ + 2x_{23}^- + M(y_{12}^+ + y_{13}^+ + y_{23}^+ + y_{12}^- + y_{13}^- + y_{23}^-) \\ & + M_o(Y_{13}^+ + Y_{23}^+ + Y_{13}^- + Y_{23}^-) \end{aligned}$$

Subject to,

$$\begin{array}{llll} x_{12}^+ + & x_{13}^+ & + (y_{12}^+ - y_{12}^-) & = 10 & \text{(Link 1 constraint)} \\ x_{13}^+ & & + (y_{13}^+ - y_{13}^-) & = 20 & \text{(Link 2 constraint)} \\ x_{13}^+ + x_{23}^+ & & + (y_{23}^+ - y_{23}^-) & = 30 & \text{(Link 3 constraint)} \end{array}$$

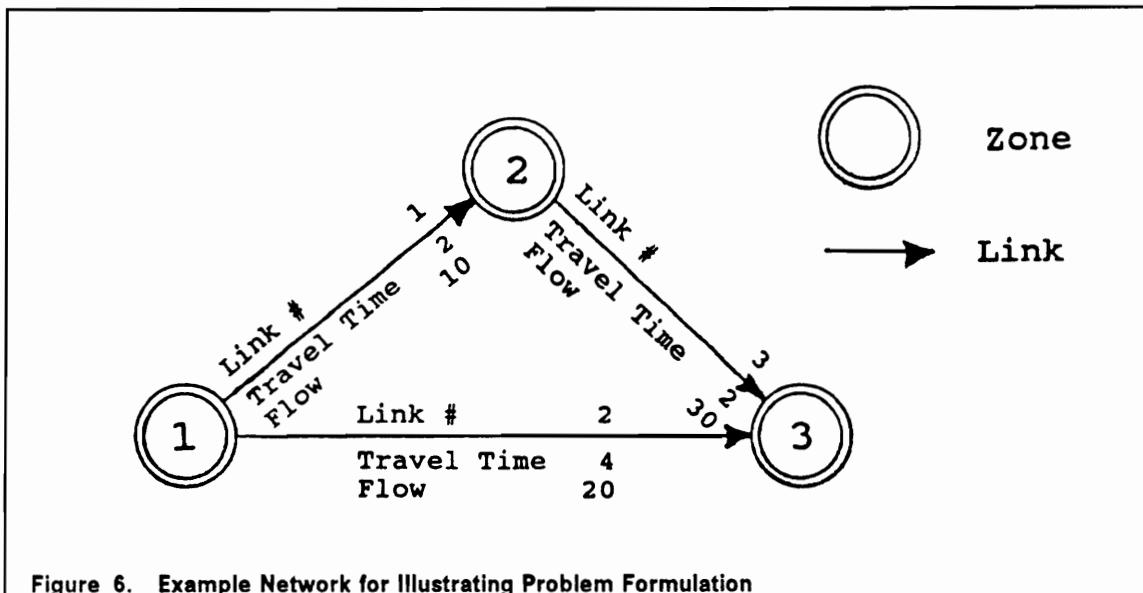


Figure 6. Example Network for Illustrating Problem Formulation

$$x_{13}^1 + x_{13}^2 + (Y_{13}^+ - Y_{13}^-) = 25 \quad (\text{Trip Table Constraint})$$

$$x_{23}^1 + (Y_{23}^+ - Y_{23}^-) = 25 \quad (\text{Trip Table Constraint})$$

$$x_{12}^1, x_{13}^1, x_{13}^2, x_{23}^1, y_{12}^+, y_{13}^+, y_{23}^+, y_{12}^-, y_{13}^-, y_{23}^-, Y_{13}^+, Y_{23}^+, Y_{13}^-, Y_{23}^- \geq 0$$

where, for example,

$x_{12}^1$  = trips from zone 1 to zone 2 via path 1,

$x_{13}^2$  = trips from zone 1 to zone 3 via path 2, etc.

$y_{12}^+$  = excess of observed flow over the modelled value on link (1,2), etc.

$y_{12}^-$  = deficit in observed flow over the modelled value on link (1,2), etc.

$Y_{13}^+$  = excess of prior trip table value over the modelled value for the O-D pair (1,3), etc.

$Y_{13}^-$  = deficit in prior trip table value over the modelled value for the O-D pair (1,3), etc.

As seen above, the formulation of the linear program requires that *all* different paths between *all* feasible O-D pairs be known, so that the objective function and the constraints can be written. When the number of origins and destinations are large, there may be numerous combinations of O-D flows that may pass through a link. The formulation in its simplest form will require that all these O-D pairs be taken into account. In addition, each O-D pair may be decomposed into several path flows. Once all the paths between all O-D pairs are enumerated, and their corresponding links are identified, enumerated, the constraint for each link can be written. Clearly, this is untenable for larger sized problems. Hence, in the sequel, we develop a suitable column generation technique which generates such paths only in an as-and-when-needed, or on-the-fly, fashion. Also, note that the number of trip table constraints will depend on the number of key O-D pairs that are specified for guiding the solution trip table. In forming these constraints, the path decomposed flows of the corresponding O-D pairs need to be considered. The objective function is formed using weighted cost coefficients (path costs) for various O-D pair path flows. The values of penalty parameters  $M$  and  $M_o$  for deviations from observed link flows and prior trip values respectively, are chosen as addressed earlier.

## 5.0 SOLUTION TECHNIQUE

The linear programming problem formulated in Chapter 4, for estimating a trip table that reproduces observed link volumes when assigned to the network, and whose cell values are as close as possible to the specified prior values, can be solved by the standard simplex procedure, provided an explicit formulation is available. However, this can be prohibitively expensive, in terms of the computational effort needed to explicitly generate the entire model. Fortunately, a more efficient implicit generation method lends itself amenable to the solution of the problem. This method is known in operations research terminology as the column generation technique. Although our linear programming model does not directly admit a standard column generation procedure, we demonstrate below how its general concept can be adapted in devising a suitable specialized approach. This chapter presents the details of such an algorithm. But before this, certain remarks on the standard application of the simplex method is at order at this point.

## **5.1 Limitations of the Standard Simplex Solution**

### **Technique**

A simple-minded approach for solving the linear programming formulation LP/LP(TT) in order to estimate an O-D trip table from link traffic volumes, would be a straightforward application of the popular linear programming solution algorithm, namely, the simplex method. But this method has a number of limitations that discouraged its use in solving the problem under consideration. These limitations are briefly enumerated below.

1. An explicit statement of the linear programming model LP/LP(TT) requires the enumeration of all possible paths between each O-D pair  $(i, j) \in OD$ . While this can be readily accomplished in theory, it is computationally prohibitive except for small sized networks, since as the network size increases, the computer effort (memory and execution time) needed to track all paths increases and may be prohibitively expensive. This factor is a potential practical limitation, since the overall goal of this research is to develop methods for real-time applications, where speed of execution is of primary importance. Also, when all possible paths between each O-D pair are to be considered, the number of decision variables to be stored in memory increases manyfold, thus taxing the computer.
2. In addition to the burden of enumerating all the paths between each O-D pair, excessive computer storage will be necessary to store the path information as required by the simplex procedure. This may lead to a practical limitation for large-sized networks.

3. In order to avoid enumerating all paths, an attempt was made to solve the linear program by considering only some efficient paths. For this purpose, Dial's (1971) definition of efficient paths was adopted. Dial's "first definition" states that a path in which each of the links has its initial ("From") node closer to the origin than its final ("To") node, and has its "To" node closer to the destination than its "From" node, is an "efficient path". Whereas such an approximation might be acceptable in certain applications, some caution must be exercised. For example, an attempt was made to solve the linear program [4.5] for the sample test case of Chapter 6, while permitting the use of only such efficient paths (determined exclusively). The LINDO (Linear Interactive aNd Discrete Optimizer) software package was employed to solve the problem. However, the problem turned out to be infeasible, whereas the original non-restricted problem is in fact feasible. This is due to not considering *all* path decompositions of O-D pairs in the formulation. Feasibility could be achieved if inefficient paths also are taken into consideration. This entails a potential enumeration of all paths.

The foregoing observation highlights the fact that one cannot ignore the possibility of having to assign trips to inefficient paths. This is because the solution has to satisfy link volume constraints, and if a link is not on any efficient path, but still carries positive flow, then the solution is forced to allot trips via inefficient paths to satisfy that link's constraint. Also, one of the objectives in developing the model is to incorporate the user-equilibrium principle of traffic assignment. Dial's method does not conform to this principle.

These are the weaknesses of using Dial's efficient paths in the formulation. Consequently, a new approach to solve the problem more efficiently has been developed, the details of which are presented below.

## **5.2 Adopted Solution Technique**

The above weaknesses of the simplex approach have been overcome by devising an algorithm in which the columns of the linear program are selectively generated, as and when needed, within the framework of the revised simplex method, in order to optimally solve the problem. Such an algorithm is known as a column generation procedure, but as explained below, requires a non-standard modification for the problem in question. This method does not require that all paths between the different O-D pairs be enumerated to formulate and solve the linear program. The neat structure of link and trip table constraints is exploited by this technique to solve the problem, bringing about considerable savings in computer effort.

## **5.3 Column Generation Technique**

"Column Generation" is a technique where the simplex method for solving linear programming problems is carried out without tabulating all columns. Here, columns are generated when the pricing operation of the simplex method finds it necessary to do so. The algorithm that is used to execute the above technique iterates between

a set of independent subproblems, whose objective functions contain variable parameters, and the master problem. Simplex multipliers or prices are sent from the master problem to the subproblem, which in turn sends its solution or proposal to the master problem. These solutions are combined by the master problem with previous solutions in an optimal way, and then new prices are computed. These are again passed on to the subproblem. The iterations stop when optimality is reached (Lasdon, 1970). Because of its nature, such a method is also sometimes called a price directive approach.

### 5.3.1 Theoretical Background

Consider the following linear program.

$$\text{Minimize } z = \sum_{j=1}^n c_j x_j \quad [5.1a]$$

subject to:

$$\sum_{j=1}^n p_j x_j = b \quad [5.1b]$$

$$x_j \geq 0, \quad j = 1, \dots, n \quad [5.1c]$$

where  $p_j$  and  $b$  are  $m$ -component vectors,  $m < n$ . Let  $x_B$  be the available initial basic feasible solution with associated basis matrix  $B$  and cost coefficients  $c_B$ . The simplex multipliers (dual vector) associated with this basis may be found as,

$$\pi = c_B B^{-1} \quad [5.2]$$

In order to improve the available basic feasible solution, all columns corresponding to the non-basic variables can be "priced" by finding their reduced cost coefficients

$$\bar{c}_j = c_j - \pi p_j \quad [5.3]$$

Let

$$\min_j \bar{c}_j = \bar{c}_s < 0 \quad [5.4]$$

Then, barring degeneracy, improvement in the current solution can be achieved by introducing  $x_s$  into the basis through a pivot transformation. When there are hundreds of columns, finding  $\min_j \bar{c}_j$  by computing each  $\bar{c}_j$  and comparing these may be a tedious process and may be computationally expensive. However, in some cases, the columns can be identified as vertices of another polyhedron  $S$ . In such a case, the column to be entered into basis can then be chosen by solving the linear program subproblem

$$\underset{p_j \in S}{\text{minimize}} \quad c(p_j) - \pi p_j \quad [5.5]$$

where  $c(p_j) = c_j$  is a linear function of  $p_j$ . The subproblem solutions are then sent to the master problem for pivoting and updating. Iterations continue until optimality is reached.

In the model developed in this research, the above approach is exploited advantageously. The subproblems arising out of adoption of the column generation technique, using certain non-standard modifications, are solved using a shortest path procedure. The details of this procedure are discussed next.

### 5.3.2 Development of a Column Generation Scheme for Problem LP(TT)

To initialize the procedure, we need a basic feasible solution to LP(TT). Artificial variables  $y^+$  and  $Y^+$  may be used as the set of initial basic variables. Alternatively, the following procedure can be adopted. First, examining the conservation of flow with respect to the observed volumes  $\bar{f}$  at the nodes in  $O \cup D$ , the sets,

$$O' = \{i \in O : \text{node } i \text{ is a net source of traffic}\}, \text{ and}$$

$$D' = \{i \in D : \text{node } i \text{ is a net receiver of traffic}\}$$

are determined. Using a backtracing algorithm (Appendix A), a path of positive flow to some  $j \in D'$  from some  $i \in O'$  can be traced. This path is recorded along with its corresponding flow, and the network data adjusted by extracting this path flow. This procedure is repeated until no such augmenting paths can be found. At this point, a feasible solution is obtained by letting  $y^+$ ,  $Y^+$ , and  $Y^-$  take up any slack, but this need not necessarily be a basic feasible solution to LP(TT). By permitting the flow

to be nonzero only on the paths explicitly generated thus far and considering all the artificial variables, a procedure as described in Bazaraa et al. (1990) is used to purify this solution to a basic feasible solution  $(\bar{x}, \bar{y}, \bar{Y})$  to LP(TT) having at least as good an objective function value (Appendix B). Also note that  $\bar{y}_a = 0$  implies that  $\bar{y}_a^+ = \bar{y}_a^- = 0$ , while  $\bar{y}_a > 0$  corresponds either to the value of  $\bar{y}_a^+$  or  $\bar{y}_a^-$ , whichever happens to be basic. An identical remark holds for  $\bar{Y}_{ij}$ .

At any stage of the algorithm, suppose that a basic feasible solution  $(\bar{x}, \bar{y}, \bar{Y})$  to LP(TT) is available. Let this be represented by a revised simplex tableau with  $(\bar{\pi}, \bar{\mu})$  being the corresponding set of simplex multipliers or complementary dual basic solution values  $c_B B^{-1}$  associated with the constraints [4.12b] and [4.12c] respectively, where  $B$  is the available basis matrix, and  $c_B$  is the vector of basic variable cost coefficients. (For convenience in presentation, let  $\bar{\mu}_{ij} \equiv 0 \forall (i, j) \in OD - \bar{OD}$ .) In order to ascertain the optimality of  $(\bar{x}, \bar{y}, \bar{Y})$ , the nonbasic variables need to be priced, that is, their reduced costs need to be computed. For the artificial variables  $y_a^+, y_a^-, a \in A$ , and  $Y_{ij}^+, Y_{ij}^-, (i, j) \in \bar{OD}$ , this reduced cost is given by  $(M - \bar{\pi}_a)$ ,  $(M + \bar{\pi}_a)$ ,  $(M_a - \bar{\mu}_{ij})$ , and  $(M_a + \bar{\mu}_{ij})$ , respectively. If any of these is negative, then the corresponding artificial variable is pivoted into the basis, and the main step is repeated. Hence, let it be supposed that no  $y$  or  $Y$ -variables are enterable in the basis.

Now, the  $x$ -variables need to be priced. However, the columns of the  $x$ -variables are not available in explicit form. So, there is a need to implicitly determine if any  $x_{ij}^k$  variable has a negative reduced cost. If it can be concluded that no such variable exists, then optimality can be declared.

Toward this end, consider the following two shortest (simple) path problems defined for each  $(i, j) \in OD$ .

$$SP_{ij}': \text{Minimize } \{(M_2 \bar{c} - \bar{\pi}) \cdot p_{ij}^k : k = 1, \dots, n_{ij}\} \equiv (M_2 \bar{c} - \bar{\pi}) p_{ij}^{k^*} \quad [5.6a]$$

$$\text{where } M_2 = \sum_{a \in A} |\bar{\pi}_a| + 1, \text{ and let } v_{ij}^1 = (\bar{c} - \bar{\pi}) p_{ij}^{k^*} - \bar{\mu}_{ij} \quad [5.6b]$$

$$SP_{ij}'': \text{Minimize } \{(M_1 \bar{c} - \bar{\pi}) \cdot p_{ij}^k : k = 1, \dots, n_{ij}\} \equiv (M_1 \bar{c} - \bar{\pi}) p_{ij}^{k''} \quad [5.6b]$$

$$\text{and let } v_{ij}^2 = (M_1 \bar{c} - \bar{\pi}) p_{ij}^{k''} - \bar{\mu}_{ij} \quad [5.7]$$

It is to be noted that  $SP_{ij}'$  and  $SP_{ij}''$  seek the shortest simple paths from  $i$  to  $j$  using respective link cost vectors  $(M_2 \bar{c} - \bar{\pi})$  and  $(M_1 \bar{c} - \bar{\pi})$ . Lemmas 3 and 4 below provide the motivation for these problems.

### Lemma 3:

$SP_{ij}'$  defined by [5.6] is a shortest path problem with nonnegative cost coefficients, and its solution yields a path  $k^* \in K_{ij}$ . Moreover, if  $v_{ij}^1 \geq 0$  in [5.6b], then no  $x_{ij}^k$  for  $k \in K_{ij}$  is enterable in the current basis for LP(TT). Otherwise,  $x_{ij}^{k^*}$  is enterable with a reduced cost of  $v_{ij}^1 < 0$ .

### Proof

Since  $\bar{c}_a \geq 1 \forall a \in A$ , by the definition of  $M_2$  in [5.6b],  $(M_2 \bar{c} - \bar{\pi}) > 0$ . Furthermore, for any  $k \in K_{ij}$  and  $\bar{k} \in \bar{K}_{ij}$ , since  $\bar{c}$  is integral, we have from [4.4] that  $\bar{c} \cdot (p_{ij}^{\bar{k}} - p_{ij}^k) \geq 1$ . Hence, we get

$$\begin{aligned}
(M_2 \bar{c} - \bar{\pi}) \cdot p_{ij}^k &= (M_2 \bar{c} - \bar{\pi}) \cdot p_{ij}^k \geq M_2 - \bar{\pi} \cdot (p_{ij}^k - p_{ij}^*) \\
&\geq M_2 - \sum_{a \in A} |\bar{\pi}_a| \geq 1
\end{aligned}$$

and so  $k^* \in K_{ij}$  in [5.6a]. Now, for any  $k \in K_{ij}$ , the reduced cost for  $x_{ij}^k$  is given from [4.5d] and [4.12] by

$$c_{ij}^k - \bar{\pi} \cdot p_{ij}^k - \bar{\mu}_{ij} = (\bar{c} - \bar{\pi}) \cdot p_{ij}^k - \bar{\mu}_{ij} \quad [5.8]$$

Hence, if  $v_{ij}^k < 0$ , then  $x_{ij}^{k^*}$  is enterable in the current basis for LP(TT). On the other hand, if  $v_{ij}^k \geq 0$ , then from [5.6], [5.8], and the fact that  $\bar{c} \cdot p_{ij}^k = \bar{c} \cdot p_{ij}^{k^*}$  by the definition of  $K_{ij}$  in [4.4], the reduced cost for  $x_{ij}^k$ ,  $k \in K_{ij}$ , is given by

$$\begin{aligned}
(\bar{c} - \bar{\pi}) \cdot p_{ij}^k - \bar{\mu}_{ij} &= (\bar{c} - \bar{\pi}) \cdot p_{ij}^k - (\bar{c} - \bar{\pi}) \cdot p_{ij}^{k^*} + v_{ij}^k \\
&= (M_2 \bar{c} - \bar{\pi}) \cdot p_{ij}^k - (M_2 \bar{c} - \bar{\pi}) \cdot p_{ij}^{k^*} + v_{ij}^k \geq v_{ij}^k \geq 0,
\end{aligned}$$

and hence no  $x_{ij}^k$ ,  $k \in K_{ij}$ , is enterable into basis. This completes the proof.  $\square$

#### Remark 6:

Lemma 3 provides a mechanism for *implicitly* pricing the nonbasic variables  $x_{ij}^k$  for  $k \in K_{ij}$ , for each  $(i, j) \in OD$ , and either finding an enterable variable, or else concluding that none exists. Moreover, since the link costs are all nonnegative for the shortest path problem  $SP_{ij}$ , the shortest path from any  $i \in O$  to all  $j \in D$  can be found in polynomial time of complexity  $O(|N|^2)$  (Bazaraa et al., 1990). Furthermore, suppose that there exists a user-equilibrium solution  $(x, y, Y)$  to LP(TT), where  $y \equiv 0$ . Then, if we are interested in finding such a solution, a sufficiently large value

of  $M$  and a sufficiently small value of  $M_\sigma$  can be used in LP(TT) to ensure that  $y \equiv 0$  at optimality. Then, beginning with an all-artificial basis, only  $SP_{ij}$  needs to be solved for pricing nonbasic variables. Since  $SP_{ij}$  provides a selective pricing of paths  $k \in K_{ij}$  by Lemma 3, this is equivalent to solving LP(TT) with the restriction that  $x_{ij}^k = 0$  for all  $k \in \bar{K}_{ij}$ ,  $(i, j) \in OD$ , and hence a user equilibrium solution can be found.

The following result addresses the pricing of the variables  $x_{ij}^k$  for  $k \in \bar{K}_{ij}$ ,  $(i, j) \in OD$ .

Lemma 4:

Suppose that  $v_{ij}^1 \geq 0$  in [5.6], and that  $SP_{ij}$  is solved and  $k^*$  and  $v_{ij}^2$  are obtained as in [5.7]. Then the following three cases apply.

(i) If  $k^* \in \bar{K}_{ij}$  and  $v_{ij}^2 < 0$ , then  $x_{ij}^{k^*}$  is enterable in the current basis.

(ii) If  $k^* \in \bar{K}_{ij}$  and  $v_{ij}^2 \geq 0$ , then no  $x_{ij}^k$  for  $k \in \bar{K}_{ij}$  is enterable in the current basis.

(iii) If  $k^* \in K_{ij}$ , then no  $x_{ij}^k$ ,  $k = 1, \dots, n_{ij}$ , is enterable in the current basis.

Proof:

Noting that the reduced cost for any  $x_{ij}^k$ ,  $k \in \bar{K}_{ij}$ , is given by

$$M_1 c_{ij}^k - \bar{\pi} \cdot p_{ij}^k - \bar{\mu}_{ij} = (M_1 \bar{c} - \bar{\pi}) \cdot p_{ij}^k - \bar{\mu}_{ij}, \quad [5.9]$$

cases (i) and (ii) follow by noting [5.7]. Hence, case (iii) needs to be considered.

Since  $v_{ij}^3 \geq 0$ , by Lemma 3 and [5.8],  $v_{ij}^3 \equiv (\bar{c} - \bar{\pi}) \cdot p_{ij}^{k^*} - \bar{\mu}_{ij} \geq 0$ . Hence, for any  $k \in \bar{K}_{ij}$ , from [5.7] and [5.9], the reduced cost for  $x_{ij}^k$  is given by

$$(M_1 \bar{c} - \bar{\pi}) \cdot p_{ij}^k - \bar{\mu}_{ij} \geq (M_1 \bar{c} - \bar{\pi}) \cdot p_{ij}^{k^*} - \bar{\mu}_{ij} \geq (\bar{c} - \bar{\pi}) \cdot p_{ij}^{k^*} - \bar{\mu}_{ij} = v_{ij}^3 \geq 0$$

Moreover, it is also known by Lemma 3 that no  $x_{ij}^k$ ,  $k \in K_{ij}$  is enterable in the basis, thus completing the proof.  $\square$

#### Remark 7:

In order to use Lemma 4, it is required to check whether the path  $k^*$  belongs to  $K_{ij}$  or to  $\bar{K}_{ij}$ . From [4.4], it is simply necessary to check if  $c_{ij}^{k^*}$  equals  $c_{ij}^*$  or exceeds it. The quantities  $c_{ij}^*$  may be computed once at the start of the algorithm by solving  $|O|$  shortest path problems with nonnegative link coefficients  $\bar{c}_a$ ,  $a \in A$ , one for determining the shortest paths from each  $i \in O$  to all  $j \in D$ . This can be accomplished in  $O(|O||N|^2)$  time.

#### Remark 8:

The shortest path problem  $SP_{ij}$  has mixed-sign cost coefficients, and so might contain negative cost circuits which are reachable from node  $i$ . In such a case, the problem of finding a shortest *simple* (no loop) path is NP-complete. Hence, an

enumerative branch-and-bound routine, as alluded in Bazaraa et al. [1990], would need to be adopted to solve  $SP_{II}$ , whenever the shortest path routine encounters a negative cost circuit. Alternatively, to avoid excessive effort for large sized networks, the following heuristic may be adopted. Consider the formulation of a minimum cost network flow programming problem with the link cost vector given by  $(M, \bar{c} - \bar{\pi})$ , with a supply of  $|D|$  at node  $i$ , and a demand of one at each node  $j \in D$ , and with an upper bound of  $|D|$  on the flow on each link  $a \in A$ . Assuming without loss of generality that all nodes in  $D$  are reachable from node  $i$ , since this is a feasible and bounded network flow programming problem, it has an optimum solution which can be obtained very efficiently. (The NETFLO routine from Kennington and Helgason (1980) was used for this purpose.) In the case of negative cost circuits, the solution might include loop flows within such circuits, in addition to path flows from  $i$  to each of the nodes  $j \in D$ , whence the corresponding paths may only be near optimal shortest paths. Nonetheless, these simple paths may be identified, along with any existing loop flows, using the backtracing routine as presented in Appendix A. Assuming these to be the shortest simple paths, Lemma 4 can now be applied.

The overall proposed algorithm for solving LP(TT) is summarized below.

### 5.3.3 Summary of Column Generation Algorithm (CGA) for Solving LP(TT)

#### Initialization (Starting Basic Feasible Solution):

For each  $i \in O$ , find the shortest paths to all  $j \in D$  using  $\bar{c}$  as the link cost vector, and hence determine  $c_{ij}^* \equiv \min\{c_{ij}^k : k = 1, \dots, n_{ij}\}$ , for each  $(i, j) \in OD$ .

Denote  $\bar{C}_{\text{total}} \equiv \bar{c} \cdot \bar{f}$ , and also define  $F_E(x, y) = \hat{c} \cdot x + M e \cdot (y^+ + y^-)$ . Note that a given solution  $(\bar{x}, \bar{y})$  represents an equilibrium solution that matches the link flows if and only if  $F_E(\bar{x}, \bar{y}) = \bar{C}_{\text{total}}$ . Furthermore, let  $M_1 = 2$ , and select a value for  $M$  as in [4.8] of Lemma 2 or as suggested in Remark 3, along with a sufficiently small value of  $\sigma \geq 0$ . Find an initial basic feasible solution  $(\bar{x}, \bar{y}, \bar{Y})$  to LP(TT) as outlined above. (Alternatively, an all-artificial basis may be used for this purpose.) Let  $(\bar{\pi}, \bar{\mu})$  be the corresponding set of simplex multipliers (complementary dual solution)  $c_B B^{-1}$ , where  $B$  is the basis matrix corresponding to  $(\bar{x}, \bar{y}, \bar{Y})$ , and  $c_B$  is the corresponding vector of cost coefficients for the basic variables. After constructing the initial revised simplex tableau, and initializing the iteration counter,  $K = 1$ , proceed to Step 1.

#### Step 1: (Pricing Artificial Variables $y^\pm$ and $Y^\pm$ ):

Compute  $v^{0+}$ ,  $v^{0-}$ ,  $N^{0+}$ , and  $N^{0-}$  as follows:

$$v^{0+} = M - \max_{a \in A} \{\bar{\pi}_a\},$$

$$v^{0-} = M + \min_{a \in A} \{\bar{\pi}_a\},$$

$$N^{0+} = M_\sigma - \max_{(i, j) \in \overline{OD}} \{\bar{\mu}_{ij}\}, \text{ and}$$

$$N^{0-} = M_\sigma + \min_{(i, j) \in \overline{OD}} \{\bar{\mu}_{ij}\}$$

If  $v^0 = \min\{v^{0+}, v^{0-}, N^{0+}, N^{0-}\} \geq 0$ , then proceed to Step 2. Otherwise, if  $v^0 = v^{0+} \equiv M - \bar{\pi}_a$ , then set  $z_K = y_a^+$ ,  $v_K = v^0$ ,  $Y_K = e_a$  (which is unit vector of length  $|A| + |\overline{OD}|$  comprised of zeros except for a one in the position  $a \in A$ ), and execute Step 4. If  $v^0 = v^{0-} \equiv M + \bar{\pi}_a$ , then set  $z_K = y_a^-$ ,  $v_K = v^0$ ,  $Y_K = -e_a$ , and go to Step 4. On the other hand, if  $v^0 = N^{0+} \equiv M_\sigma - \bar{\mu}_{(i,j)}$ , then set  $z_K = Y_{(i,j)}^+$ ,  $v_K = v^0$ ,  $Y_K = e_{(i,j)}$  (which is

again a unit vector of length  $|A| + |\overline{OD}|$  with the element one corresponding to position  $(i,j)^* \in \overline{OD}$ , and proceed to Step 4. Finally, if  $v^0 = N^{0-} \equiv M_\sigma + \bar{\mu}_{(i,j)^*}$ , then set  $z_K = Y_{(i,j)^*}^-$ ,  $v_K = v^0$ ,  $Y_K = -e_{(i,j)^*}$ , and go to Step 4.

#### Step 2: (Pricing $x_{ij}^k$ variables for $k \in K_{ij}$ , $(i, j) \in OD$ and User Equilibrium Check):

For each  $i \in O$ , solve the shortest path problem  $SP_{ij}^1$  given by [5.6], for all  $j \in D$ , as in Remark 6. Determine  $(ij)^* \in \operatorname{argmin} \{v_{ij} : (i, j) \in OD\}$ . If  $v_{(ij)^*}^1 \equiv (\bar{c} - \bar{\pi}) \cdot p_{(ij)^*}^{k^*} - \bar{\mu}_{(i,j)^*} < 0$ , then set  $z_K = x_{(ij)^*}^{k^*}$ ,  $v_K = v_{(ij)^*}^1$ , and execute Step 4. Otherwise,  $v_{(ij)^*}^1 \geq 0$ . Hence, if  $F_E(\bar{x}, \bar{y}) = \bar{C}_{\text{total}}$ , terminate the algorithm with the indication of an optimal user equilibrium solution to LP(TT). Otherwise, if this (partial) objective value exceeds  $\bar{C}_{\text{total}}$ , proceed to Step 3.

#### Step 3: (Pricing $x_{ij}^k$ variables for $k \in \bar{K}_{ij}$ , $(i, j) \in OD$ , and Termination Check):

For each  $i \in O$ , solve the shortest (simple) path problem  $SP_{ij}^2$  given by [5.7], for all  $j \in D$ , as in Remark 8. Using Remark 7, determine

$$(ij)^* \in \operatorname{argmin} \{v_{ij}^2 : (i, j) \in OD \text{ and } k^* \in \bar{K}_{ij} \text{ in [5.7]}\}$$

If  $(ij)^*$  does not exist, or if  $v_{(ij)^*}^2 \geq 0$ , terminate the algorithm; the current solution solves the problem LP(TT). Otherwise, if  $v_{(ij)^*}^2 \equiv (M_1 \bar{c} - \bar{\pi}) \cdot p_{(ij)^*}^{k^*} - \bar{\mu}_{(i,j)^*} < 0$ , where  $k^* \in \bar{K}_{ij}$ , then set  $z_K = x_{(ij)^*}^{k^*}$ ,  $v_K = v_{(ij)^*}^2$ ,  $Y_K = p_{(ij)^*}^{k^*}$ , and go to Step 4.

#### Step 4: (Updating Primal and Dual Solutions):

Compute the updated column  $B^{-1} Y_K$  of the (entering) nonbasic variable  $z_K$  with reduced cost  $v_K < 0$ , pivot it into the basis, and hence update the basic feasible

solution, the basis inverse, and the dual vector  $(\bar{\pi}, \bar{\mu})$  of simplex multipliers. Increment  $K$  by one, and return to Step 1.

Theorem 1:

Suppose that the algorithm CGA is implemented with a lexicographic cycling prevention rule at Step 4 (Bazaraa et al., 1990). Then the algorithm converges finitely with an optimal solution  $(x^*, y^*, Y^*)$  to LP(TT). Furthermore, suppose that  $M$  is chosen at least as in [4.8c]. Then the value of  $F_E(\bar{x}, \bar{y}) \equiv \hat{c} \cdot x + M\epsilon \cdot (y^+ + y^-)$  at optimality is at least  $\bar{C}_{\text{total}}$ . Moreover, there exists a user-equilibrium solution which reproduces the observed flows  $\bar{f}$  if and only if  $F_E(\bar{x}, \bar{y}) = \bar{C}_{\text{total}}$ . In the latter case, the set of optimal solutions to LP(TT) is given by such equilibrating solutions, and the algorithm will attain optimality without ever executing Step 3. In addition, for  $\sigma > 0$  and sufficiently small, LP(TT) will find that equilibrium solution (among possible alternatives) which most closely resembles the given prior trip table in the sum of total absolute deviations. Again, this will happen without resorting to Step 3.

Proof:

Since LP(TT) is feasible and bounded, by Lemmas 3 and 4, and by the finite convergence of the (revised) simplex algorithm under a lexicographic cycling prevention rule (Bazaraa et al., 1990), Algorithm CGA converges finitely to an optimal solution for Problem LP(TT). Furthermore, suppose that  $M$  is chosen as in [4.8c]. By [4.10], the value of  $F_E(\bar{x}, \bar{y})$  at optimality is at least  $\bar{C}_{\text{total}}$ , and by Lemmas 1 and 2, if there exists a user-equilibrium solution which reproduces  $\bar{f}$ , then any such solution is optimal to LP(TT) with  $F_E(\bar{x}, \bar{y}) = \bar{C}_{\text{total}}$ . Conversely, if an optimum to LP(TT) has

$F_E(\bar{x}, \bar{y}) = \bar{C}_{\text{total}}$ , then by [4.10] it must be feasible to the linear program [4.5], and so by Lemma 1, it must conform to a user-equilibrium solution which reproduces  $\bar{f}$ .

Finally, suppose that there exists a user equilibrium solution which reproduces  $\bar{f}$ . In such a case, if ever the algorithm enters Step 3, Lemma 3 asserts that the current solution  $(\bar{x}, \bar{y}, \bar{Y})$  is optimal to the linear program LP'(TT) which is obtained from LP(TT) by permanently restricting the current nonbasic variables  $y_a^+$ ,  $a \in A$ , and the variables  $x_{ij}^k$ ,  $k \in \bar{K}_{ij}$ ,  $(i, j) \in OD$ , to zeros. But if  $\bar{y}_a > 0$  for any  $a \in A$ , or if  $\bar{x}_{ij}^k > 0$  for any  $k \in \bar{K}_{ij}$ ,  $(i, j) \in OD$ , then via [4.10], the value of  $F_E(\bar{x}, \bar{y})$  strictly exceeds  $\bar{C}_{\text{total}}$  as in the proof of Lemma 2. However, any user equilibrium solution which reproduces  $\bar{f}$  is feasible to LP'(TT) with  $F_E(\bar{x}, \bar{y}) = \bar{C}_{\text{total}}$ , contradicting the optimality of  $(\bar{x}, \bar{y})$ . Hence,  $F_E(\bar{x}, \bar{y})$  must have the value  $\bar{C}_{\text{total}}$ . This completes the proof.  $\square$

#### Remark 9:

It is worth reiterating that by Theorem 1, if there exists a user equilibrium solution which reproduces  $\bar{f}$ , then so long as  $M \geq 1 + \bar{c}$ , any optimum to LP(TT) will conform to such a solution, and moreover, Step 3 will never need to be executed. Hence, all shortest path subproblems solved in the solution process will have nonnegative cost coefficients, and are hence executed in polynomial time. Furthermore, for the general situation, by checking if  $F_E(\bar{x}, \bar{y})$  value is equal to  $\bar{C}_{\text{total}}$  or exceeds it, it can be ascertained whether or not such an equilibrium solution exists.

## 6.0 MODEL TESTS, COMPARISONS AND VALIDATION

### 6.1 *Introduction*

In order to validate the linear programming model LP(TT) developed in this research, and to judge its merits in estimating O-D trips, test runs were made on a sample network. The results were then compared to those of two other existing models that also estimate trip tables from link traffic volumes, are well known, and are also available in the form of computer packages. These models are the maximum entropy and the LINKOD models, that respectively employ the maximum entropy and the network equilibrium approaches. Both these approaches have been tested by many researchers. The parameter calibration models were not included in the comparison, since they constitute a different class of models, owing to their data requirements on zone-specific variables. Also, since, computer programs for the largely untested statistical estimation models were not available for use in the

comparison, these were also not included. We first use a sample test network to compare the trip tables obtained through these different models, and their ability to replicate observed link volumes, using various statistical measures of fit. The behavior of the models in the light of prior information on trip pattern are also studied. The run time, the ability of the modelled trip table to replicate observed link volumes, and the closeness of the solution to the target trip table are used as a set of criteria in the comparisons. A critical discussion of results is then presented. Finally, the models are compared based on the tests on four more hypothetical networks. While these tests are conducted on sample networks, in the next chapter, we will apply the linear programming models LP and LP(TT) to a real network in Northern Virginia, for observing the behavior of the model on real data, and for further validation.

## 6.2 *Sample Test Network*

The sample network chosen for the tests is a hypothetical network called the "Corridor Network", reflecting a travel corridor (Gur et al., 1980). This network has been used extensively for tests while developing the LINKOD system of models. The network consists of 6 zones, 6 intersection nodes, and 18 links. Although this is a simple network, it offers multiple routes between O-D pairs, and multiple equilibrium solutions, thus presenting an opportunity to apply and test equilibrium concepts to account for congestion effects. Another desirable feature of this network is that the conservation of flow is satisfied at all intersection nodes, as required by some models. Finally, due its small size, it is an ideal network for performing several test runs of various models without expending excessive time and computer resources.

Its attributes are shown in Figure 7 and Table 2. All the three models were tested in turn on this sample network.

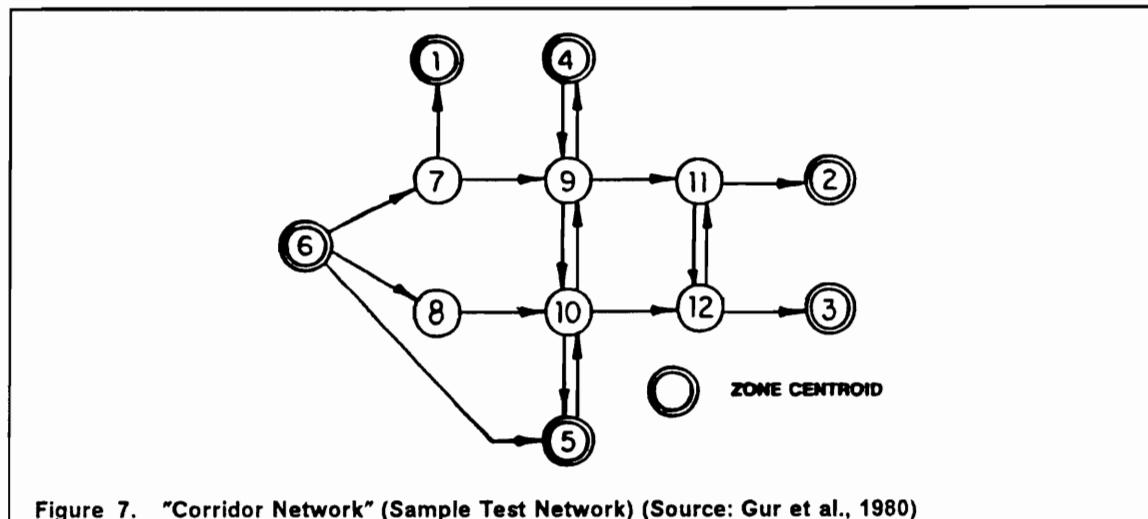


Table 2. "Corridor Network" Characteristics (Source: Gur et al., 1980)

Link Number	Beginning Node	Ending Node	Observed Impedance	Free-Flow Impedance	Observed Volume
1	4	9	10	7	2400
2	5	10	10	7	2000
3	6	5	40	30	100
4	6	7	10	7	5000
5	6	8	10	7	500
6	7	1	10	7	500
7	7	9	20	15	4500
8	8	10	20	15	500
9	9	4	10	7	2000
10	9	10	10	7	1500
11	9	11	20	15	4900
12	10	5	10	7	1600
13	10	9	10	7	1500
14	10	12	20	15	900
15	11	2	20	15	4800
16	11	12	10	7	300
17	12	3	20	15	1000
18	12	11	10	7	200

## **6.3 Computer Resource for Tests**

Trip table estimation procedures are known to be iterative in nature, incorporating complex algorithms. Hence, computer execution times and memory requirements to run these models are expected to be large. Mainframe computers with greater available memory and faster speed would inevitably be necessary to run these models for a large-sized network. The smaller size of the sample network chosen for the tests renders it possible to use a personal computer for test runs. This provides considerable flexibility in performing repeated tests. The IBM personal computer system PS/2 Model 50Z, with a math coprocessor, and a clock speed of 10MHz, was used for the tests. Run times were noted for all the runs.

## **6.4 Initial Tests of Linear Programming Model LP With No Prior Trip Table Information**

With an interest to test the linear programming model developed initially, the model LP (equation 4.7) was run for the sample test network described above, without utilizing any prior trip table information. This presented an opportunity to test the effect of an underspecified problem on trip table estimation. Both "advanced-start" and "artificial initial basis" procedures were employed to test the solutions. Both the approaches produced optimal trip table solutions, shown below in Tables 3 and 4, that replicated observed link volumes exactly. Also, both these trip tables were

equilibrium solutions, satisfying Wardrop's (first) principle. These two properties are confirmed directly from the algorithm, without the need for performing traffic assignments. In particular, the equilibrium status is verified by simply looking at the optimal objective value, and declaring compliance or non-compliance according to the satisfaction or non-satisfaction of the conditions set forth in Lemma 1 (Chapter 4). However, to verify the correctness of the computer code and the accuracy of the resulting solutions, a traffic assignment of the solution trip tables were performed using the equilibrium principle. The results were in total agreement with the above inferences.

**Table 3. Solution through Linear Programming Model LP (Without Prescribing a Target Trip Table)**  
**- Advanced-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	300	0	1500	2400
5	0	200	300	1500	0	2000
6	500	4000	400	500	200	5600
Total	500	4800	1000	2000	1700	10000

**Table 4. Solution through Linear Programming Model LP (Without Prescribing a Target Trip Table)**  
**- Artificial-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	700	0	1100	2400
5	0	1700	300	0	0	2000
6	500	2500	0	2000	600	5600
Total	500	4800	1000	2000	1700	10000

Observe that the cell values of the two trip tables in Tables 3 and 4 are significantly different. As a measure of this difference, the root mean square error for the deviation of these two table cell values is found to be 936.14, while the average cell value of the tables is only 909.09. The above experience endorses the issue of an underspecified problem. The above two tables, both of which are user-equilibrium solutions to the problem, indicate multiple optima, thus raising the issue of prescribing an appropriate solution. This motivated the need for targeting the solution to a prior trip table. The issue then is to prescribe an appropriate target table.

## 6.5 *Target Trip Table*

A target trip table is a useful device that resolves the underspecification of the O-D table estimation problem in the presence of multiple optima (alternate solution trip tables). Here, the solution table is guided toward that which is closest to the target table. Different target tables can, however, be used, leading again to different possible solutions. A prior table could be a structural table, with zeros and ones in the cells, to ensure that the final table retains the same structure. Or it could be a "no-information" prior table with all feasible interchanges assigned the same value of trips. Better still, the target table could be one that is somewhat close to the final solution, based on an older table or on some historical data or estimation process. The accuracy of the final solution will naturally depend on which target table is provided.

In the light of the above, three different target trip tables were tried in the following model comparisons. These tables were used by Gur et al. (1980) in testing

the LINKOD system, and denote various degrees of closeness to a "correct" trip table that reproduces observed link volumes exactly and is an equilibrium solution. The tables were derived using the SMALD (Small Area Trip Distribution) model of the LINKOD system, and are shown below in Tables 5, 6 and 7.

---

**Table 5. "No-Prior-Information" Target Trip Table**

To From	1	2	3	4	5	Total
4	0	983	983	0	983	2949
5	0	983	983	983	0	2949
6	983	983	983	983	983	4915
Total	983	2949	2949	1966	1966	10813

---



---

**Table 6. "Relatively Small Errors" Target Trip Table**

To From	1	2	3	4	5	Total
4	0	806	504	0	1109	2419
5	0	1512	504	0	0	2016
6	504	2520	0	2016	605	5645
Total	504	4838	1008	2016	1714	10080

---

Table 7. Correct Target Trip Table						
To From	1	2	3	4	5	Total
4	0	600	700	0	1100	2400
5	0	1700	300	0	0	2000
6	500	2500	0	2000	600	5600
Total	500	4800	1000	2000	1700	10000

Table 5 represents a "no-prior-information" trip table with all feasible interchanges carrying an equal number of trips. A better starting table is represented in Table 6, but it is still unsatisfactory, because it does not reproduce observed flows on links when assigned to the network. Table 7, however, lies at the other extreme, being a precise user-equilibrium solution that replicates link flows. This range of available target tables presents an opportunity to examine the sensitivity of model results to the kind of target table provided. Hence, the three models were tested in turn with the above tables, and the results analyzed.

## 6.6 Comparison of Test Results

In order to compare the test results from different models, two measures of closeness are used in judging the results. The first measure is based on the replication of link volumes by the solution trip table when assigned to the network. This is accomplished by comparing the output link volumes obtained from an equilibrium assignment program, with the observed volumes. This also enables to

verify if the trip table is an equilibrium solution or not. In the case of the linear programming model, this test is also conducted within the model itself. The second measure is the closeness of the solution table to the target table. These two criteria are obvious choices, since the objective of the problem of trip table estimation is to determine a table that replicates observed link volumes when assigned, and is as close to the target table as possible. The final interest is in the comparison of computer run times for the various models. This is of particular interest in the context of real-time applications, which is the final goal of the project.

### **6.6.1 Replication of Observed Link Volumes on User-Equilibrium**

#### **Assignment of the Estimated Trip Table**

The most important measure of the quality of the estimated trip table is its ability to replicate observed volumes on the network links when assigned using a traffic assignment procedure. While different assumptions can be made in assigning the traffic, it must be ensured that any assumption used is consistent with the assignment procedure that is incorporated within the trip table estimation approach. As stated earlier, since congestion is a common phenomenon in urban areas, the equilibrium principle is an appropriate assumption for making traffic assignments. While both the network equilibrium and the linear programming approaches are equilibrium based, the maximum entropy approach as incorporated in the original "THE" (The Highway Emulator) (Bromage, 1988) program utilizes a capacity restrained based proportional assignment technique. Although this capacity restrained assignment is a heuristic user-equilibrium method, its convergence is not guaranteed and Wardrop's principle may not be satisfied at termination. This introduces unfairness in the following

comparison, because the equilibrium principle of traffic assignment is used for all the models to test the property of replicating observed link volumes. However, a recent version of "THE" (Beagan, 1990) incorporates the user-equilibrium principle in its assignment step. This recent version was therefore used in order to make a fair comparison. Since the assignment algorithm is iteratively convergent, the trip table may not necessarily replicate exactly the observed volumes on the links when assigned to the network. In such cases, the degree to which the link volumes are replicated will depend on the number of iterations and other stopping criteria specified for the assignment algorithm.

For the purpose of assigning volumes to the network from the derived trip tables using the user-equilibrium principle, a traffic assignment program that incorporates this principle is used. This program is coded in FORTRAN by O'Neill (1986). The Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are chosen as measures of error rate to compare the assigned volumes with observed volumes. These measures are defined as follows.

$$RMSE = \sqrt{\frac{\sum (V_{assign} - V_{obs})^2}{n}}$$

$$MAE = \frac{\sum |V_{assign} - V_{obs}|}{n}$$

where,

$V_{assign}$  = equilibrium assigned link volume

$V_{obs}$  = observed link volume

$n$  = number of observations

### 6.6.2 Closeness of Estimated Trip Table to the Target Table

The target trip table is employed mainly to guide the solution to a known trend in trip pattern, and to make a choice among alternative optimal solutions, if they exist, in order to choose one that is closest to the target. An ideal solution is one that is the same as (or as close as possible to) the target table. There are various measures of closeness for comparing trip matrices. Smith and Hutchinson (1981) evaluate different goodness of fit statistics for trip distribution models and conclude that the phi-statistic ( $\phi$ ) is one of the most appropriate to test the goodness of fit of alternative trip distribution models. The mean absolute error statistic has also been reported as an useful indicator. Consequently, RMSE, MAE and  $\phi$  are used in the following analyses for trip table comparisons. These measures of closeness are defined below.

$$RMSE = \sqrt{\frac{\sum (t_{ij} - t_{ij}^*)^2}{n}}$$

$$MAE = \frac{\sum |t_{ij} - t_{ij}^*|}{n}$$

$$\phi = \sum_i \sum_j t_{ij} \left| \ln \frac{t_{ij}^*}{t_{ij}} \right|$$

where,

$t_{ij}$  = estimated number of trips for interchange ( $i, j$ )

$t_{ij}^*$  = targeted number of trips for interchange ( $i, j$ )

$n$  = number of feasible O-D interchanges

Since the above statistics are measures of error in estimation, the smaller the values of these measures, the closer are the tables under comparison to the target

table. Ideally, values of zero for each of these statistics would mean that the estimated table is the same as the target table.

### **6.6.3 Comparison of Computer Run Times**

As a final comparison, total run times taken by the three models to produce the solutions were determined on the IBM PS/2, Model 50Z, personal computer. These run times are expected to be indicators of the applicability of these models in real-time scenarios of diversion, where quick decision making is warranted.

## **6.7 Test Results**

All the three models, namely, the linear programming, the maximum entropy, and the LINKOD models were run using each of the three different target trip tables. The influence of the target table on the final table for each of the three models are observed. A FORTRAN program developed in this research was used to solve the linear programming model LP(TT). For the maximum entropy approach, "The Highway Emulator (THE)" package developed by Bromage (1988) and enhanced by Beagan (1990) was utilized. This is a highway traffic simulation model designed for modelling of individual communities, corridors, sections of counties, and for the analysis of small sections of major cities. In addition to the traditional four step planning algorithms, it has a trip table estimation algorithm. For the network equilibrium approach, the program (in Advanced BASIC) developed for personal

computers by O' Neill (1985) was used. This program executes the functions of the ODLINK module of the LINKOD system (Gur et al., 1980). In other words, given a prior trip table, it corrects it iteratively so as to replicate observed link volumes.

The solutions from the three models are presented and analyzed below, for each initial trip table used.

### **6.7.1 Case 1: Use of a "No-Prior-Information" Target Trip Table**

This table utilizes no prior information on the number of trips between interchanges. A uniform cell value of 983 trips for all the feasible O-D interchanges is considered as the target. This table is reported to match the total observed vehicle hours travelled on the network. The same table values were used for all the models to enable a common basis of comparison. The results are presented below.

#### **6.7.1.1 Linear Programming Model**

The linear programming model, LP(TT) of equation [4.12], which incorporates prior trip table information in the form of additional constraints, was run for the sample test problem. Both "advanced-start" and "artificial initial basis" procedures were employed. Parameter values of  $M_1 = 2$  and  $M = 1 + \bar{c} + \bar{C}_{\text{total}}$  were used as in [4.8b]. Since, a no-information prior trip table was being used,  $M_\sigma = 0.1 \bar{c}$  was used to permit the user-equilibrium driving component of the objective function to dominate. This led the solution to that user-equilibrium solution which more closely matches the prior trip table. When  $M_\sigma$  was increased beyond 0.32  $\bar{c}$ , a

non-equilibrium solution resulted, but one that grew closer to the prior trip table flows as  $\sigma$  was subsequently increased. Both the starting procedures, namely, the "advanced-start" and the "artificial initial basis" procedures were tested, as described below.

#### Advanced-Start Procedure:

For this case an initial basic feasible solution  $(\bar{x}, \bar{y}, \bar{Y})$  to LP(TT) was obtained using the backtracing method and the purification process as outlined in Chapter 5 (and in Appendices A and B). The basis comprised of some legitimate variables representing O-D pairs and other artificial variables, as shown in Table 8. The remaining variables were nonbasic at zero values. The column generation algorithm was then used to optimize the initial solution. Table 9 shows the optimal trip table obtained after 46 iterations of Algorithm CGA, 34 of which were associated with Step 1, and 12 iterations with Step 2. The following artificial variables, corresponding to the deviations from prior trip table constraints, remained in the basis:  $Y_{4,2}^+ = 383$ ,  $Y_{4,3}^+ = 283$ ,  $Y_{5,2}^+ = 266$ ,  $Y_{5,3}^+ = 683$ ,  $Y_{6,1}^+ = 483$ ,  $Y_{6,3}^+ = 983$ ,  $Y_{6,5}^+ = 383$ ,  $Y_{4,5}^- = 117$ ,  $Y_{6,2}^- = 2500$ ,  $Y_{6,4}^- = 34$ .

#### Artificial Initial Basis Procedure

Here, the Algorithm CGA was run by simply using an all-artificial initial basis. Hence, the initial solution  $(\bar{x}, \bar{y}, \bar{Y})$  is given by  $\bar{x} = 0$ ,  $\bar{y}_a^+ = \bar{f}_a$ ,  $\bar{y}_a^- = 0$ ,  $\bar{Y}_{ij}^+ = Q_{ij}$ ,

**Table 8. Advanced-Start Basis for Linear Programming Model LP(TT) (Case 1)**

**Legitimate Variables**

Basic Variable	Value	Links Used
$\bar{X}_{6,3}^1$	100	3,2,14,17
$\bar{X}_{6,3}^2$	300	4,7,10,14,17
$\bar{X}_{6,3}^3$	500	5,8,14,17
$\bar{X}_{6,3}^4$	100	4,7,11,16,17
$\bar{X}_{6,1}$	500	4,6
$\bar{X}_{6,2}$	4100	4,7,11,15
$\bar{X}_{5,2}$	700	2,13,11,15

**Artificial Variables**

Basic Variable	Value
$\bar{y}_{4,9}^+$	2400
$\bar{y}_{5,10}^+$	1200
$\bar{y}_{7,9}^+$	0
$\bar{y}_{8,10}^+$	0
$\bar{y}_{9,4}^+$	2000
$\bar{y}_{9,10}^+$	1200
$\bar{y}_{10,5}^+$	1600
$\bar{y}_{10,9}^+$	800
$\bar{y}_{11,12}^+$	200
$\bar{y}_{12,3}^+$	0
$\bar{y}_{12,11}^+$	200
$\bar{Y}_{4,2}^+$	983
$\bar{Y}_{4,3}^+$	983
$\bar{Y}_{4,5}^+$	983
$\bar{Y}_{5,2}^+$	283
$\bar{Y}_{5,3}^+$	983
$\bar{Y}_{5,4}^+$	983
$\bar{Y}_{6,1}^+$	483
$\bar{Y}_{6,4}^+$	983
$\bar{Y}_{6,5}^+$	983
$\bar{Y}_{6,2}^-$	3117
$\bar{Y}_{6,3}^-$	17

and  $\bar{Y}_{ij}^- = 0 \quad \forall a \in A, (i, j) \in \overline{OD}$ . Algorithm CGA went through 56 iterations (40 iterations associated with Step 1, and 16 iterations with step 2), before yielding the optimal solution shown in Table 10. This final solution was exactly the same as that given by the advanced-start procedure.

### **6.7.1.2 Maximum Entropy Model**

The trip table estimation procedure in "The Highway Emulator" (THE) computer model, used for testing the results of maximum entropy approach, is based on the algorithm defined by Van Zuylen and Willumsen (1980). This model estimates a trip table based solely on link volumes. One of the limitations, however, is that the traffic counts must be balanced. In other words, the volume of traffic entering a node (intersection) must equal the volume exiting. Although this requirement is difficult to obtain in real networks, the hypothetical network being tested here satisfies this condition. One other constraint in this program is that the estimated trip table must reproduce the observed volumes when assigned (Bromage, 1988).

The trip table estimation algorithm in this program performs a traffic assignment as the first step. A capacity restrained assignment based on the Bureau of Public Roads (BPR) (1964) link performance function is incorporated in the original THE model algorithm. The maximum number of iterations executed for the assignment algorithm needs to be provided by the user. The link use probabilities computed from the assignment step are then used to adjust the prior trip table, in order to obtain a new trip table. This is the second step in the algorithm. This step, because of its iterative nature and the requirement to process an extensive number of link probabilities, is very time consuming. Since the probabilities were developed based on a capacity restrained assignment, this new trip table may no longer produce the same link probabilities and assignment as the previous assignment iteration. Hence the new trip table must be reassigned and the process repeated (Bromage, 1988). Thus there are three iterative steps in the algorithm, namely, the assignment step, the trip table adjustment step, and the whole process of reassignment and readjustment of the trip table. Bromage recommends at least five iterations of each

step for good results. This was followed in the tests conducted. The resulting solution tables when assigned using equilibrium principle were unable to replicate observed volumes closely.

Citing the well known convergence problems of the iterative capacity restrained assignment technique incorporated in the above version, Beagan (1990) recently included the equilibrium assignment option by incorporating the algorithm proposed by Hall et al. (1980). Also, since the original version was sensitive to user-selected iteration limits for various steps (O'Neill and Sivanandan, 1989), the current version has removed the requirement of user specified values for these iterations, except for calibration. With certain other improvements also incorporated, the equilibrium assignment-based maximum entropy model presented an opportunity to compare the results of this approach at par with the other two approaches. This model was hence used.

The trip table estimation program in THE model does not always continually reduce the error rate of the assigned trip table, after completion of a calibration iteration. Thus, there is a possibility that a trip table generated in an interim iteration is better than the one generated by the final iteration. Hence, the model is programmed to save a trip table after an iteration, only if it is better than the one already saved, with regard to replicating observed flows on assignment (Bromage, 1988). Due to the absence of an appropriate termination criterion, we need to select a certain number of iterations, before enforcing termination. Hence, as suggested by Beagan, 10 calibration iterations were used to achieve good results. The model results for the "no-information" target case is shown in Table 11.

### **6.7.1.3 Network Equilibrium Approach**

This approach has been embodied in the LINKOD model system (Gur et al., 1980), which has been programmed for a mainframe computer. However, since one of the interests in this research is to compare the run times for the various models on a personal computer, the PC version of the LINKOD model, programmed by O'Neill (1985), was employed. Since the algorithm is iterative in nature, various stopping criteria need to be specified by the user for terminating the algorithm. The three parameters used for this purpose were the maximum number of iterations, accuracy of the final uncertainty interval in the golden section search, and the convergence test on volume error. Gur et al. (1980) report that the volume error criterion gives more insight. It is also recommended to prespecify the number of iterations for practical applications. Numerous experiments are reported to indicate that the first 10 iterations see most of the corrections, and that very small corrections take place after more than 25 iterations. The model was run using O'Neill's program to achieve better accuracy, by increasing the number of iterations, and reducing the tolerance on volume error. These convergence criteria were chosen so as to obtain solutions comparable to those of the maximum entropy and the linear programming models. The resulting matrices are shown in Tables 12 and 13.

**Table 9. Final Trip Table Using Linear Programming Model LP(TT) - Advanced-Start Procedure**

To From	1	2	3	4	.5	Total
4	0	600	700	0	1100	2400
5	0	717	300	983	0	2000
6	500	3483	0	1017	600	5600
Total	500	4800	1000	2000	1700	10000

---

**Table 10. Final Trip Table Using Linear Programming Model LP(TT) - Artificial-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	700	0	1100	2400
5	0	717	300	983	0	2000
6	500	3483	0	1017	600	5600
Total	500	4800	1000	2000	1700	10000

---

**Table 11. Final Trip Table Using Maximum Entropy Model**

To From	1	2	3	4	5	Total
4	0	622	552	0	1225	2399
5	0	1471	299	232	0	2002
6	500	2710	148	1767	477	5602
Total	500	4803	999	1999	1702	10003

---

**Table 12. Final Trip Table Using LINKOD Model (238 Iterations)**

To From	1	2	3	4	5	Total
4	0	630	475	0	1308	2413
5	0	1037	299	656	0	1992
5	501	3122	221	1359	379	5582
Total	501	4789	995	2015	1687	9987

---

---

**Table 13. Final Trip Table Using LINKOD Model (297 Iterations)**

To From	1	2	3	4	5	Total
4	0	606	480	0	1317	2403
5	0	1046	298	655	0	1999
5	500	3147	218	1347	382	5594
Total	500	4799	996	2002	1699	9996

---

A comparison of the equilibrium assigned volumes versus the observed volumes, and the closeness of the estimated tables to the target table are shown in Tables 14 and 15, respectively, for the various models. With respect to replicating observed link volumes, the LINKOD model did not converge to the same degree of accuracy, as obtained from the LP(TT) model, even after significantly greater number of iterations. In the calculation of  $\phi$  statistic in Table 15, a value of one has been assumed for cell values with zero trips in the estimated trip table, in order to overcome division by zero. Finally, computer run times for the models are given in Table 16.

---

**Table 14. Comparison of Equilibrium Assigned and Observed Link Volumes**

Trip Table From:	(Between Assigned and Obs. Volumes)	
	RMSE	MAE
LP(TT) Model (Advanced-Start)	0.74	0.43
LP(TT) Model (Artificial-Start)	0.74	0.43
Maximum Entropy Model	10.33	6.58
LINKOD Model (238 iterations)	9.62	7.82
LINKOD Model (297 iterations)	2.54	2.19

---

---

**Table 15. Closeness of Estimated Trip Table to the Target Table**

<b>Trip Table From:</b>	<b>RMSE</b>	<b>MAE</b>	$\phi$
<b>LP(TT) Model (Advanced-Start)</b>	872.57	555.91	11606.63
<b>LP(TT) Model (Artificial-Start)</b>	872.57	555.91	11606.63
<b>Maximum Entropy Model</b>	764.46	662.91	9028.81
<b>LINKOD Model (238 iterations)</b>	795.47	601.27	7573.97
<b>LINKOD Model (297 iterations)</b>	802.42	606.09	7628.66

---

**Table 16. Comparison of Computer Run Times for Test Network**

	<b>Run Time (sec)</b>
<b>LP(TT) Model (Advanced-Start)</b>	9
<b>LP(TT) Model (Artificial-Start)</b>	10
<b>Maximum Entropy Model</b>	318
<b>LINKOD Model (238 iterations)</b>	170
<b>LINKOD Model (297 iterations)</b>	213

---

### **6.7.2 Case 2: Use of a "Relatively Small Errors" Target Trip Table**

To determine the influence of providing a better target trip table on the solution, the "relatively small errors" trip table given in Table 6 was used as the target. The model parameters and other run settings were maintained the same as in case 1. The results of these test runs are given below, in a format similar to that for case 1.

### **6.7.2.1 Linear Programming Model**

#### **Advanced-Start Procedure:**

The initial basis is shown in Table 17. The remaining variables were nonbasic at values zero. Table 18 shows the optimal trip table obtained after 41 iterations of Algorithm CGA, 30 of which were associated with Step 1, and 11 iterations with Step 2. The following artificial variables, corresponding to the deviations from prior trip table constraints, remained in the basis:  $Y_{4,2}^+ = 206$ ,  $Y_{5,3}^+ = 204$ ,  $Y_{6,1}^+ = 4$ ,  $Y_{6,2}^+ = 20$ ,  $Y_{6,4}^+ = 16$ ,  $Y_{6,5}^+ = 14$ ,  $Y_{4,3}^- = 187$ ,  $Y_{5,2}^- = 188$ ,  $Y_{6,3}^- = 9$ .

#### **Artificial Initial Basis Procedure**

In this case, the Algorithm CGA went through 50 iterations (35 iterations associated with Step 1, and 15 iterations with step 2), before yielding the optimal solution shown in Table 19. This final solution was exactly the same as that given by the advanced-start procedure.

### **6.7.2.2 Maximum Entropy Model**

Model runs were made again with same number (10) of calibration iterations. The estimated final trip table is shown in Table 20.

**Table 17. Advanced-Start Basis for Linear Programming Model LP(TT) (Case 2)**

**Legitimate Variables**

Basic Variable	Value	Links Used
$\bar{X}_{6,3}^1$	100	3,2,14,17
$\bar{X}_{6,3}^2$	300	4,7,10,14,17
$\bar{X}_{6,3}^3$	500	5,8,14,17
$\bar{X}_{6,3}^4$	100	4,7,11,16,17
$\bar{X}_{6,1}^1$	500	4,6
$\bar{X}_{6,2}^1$	4100	4,7,11,15
$\bar{X}_{5,2}^1$	700	2,13,11,15

**Artificial Variables**

Basic Variable	Value
$\bar{y}_{4,9}^+$	2400
$\bar{y}_{5,10}^+$	1200
$\bar{y}_{7,9}^+$	0
$\bar{y}_{8,10}^+$	0
$\bar{y}_{9,4}^+$	2000
$\bar{y}_{9,10}^+$	1200
$\bar{y}_{10,5}^+$	1600
$\bar{y}_{10,9}^+$	800
$\bar{y}_{11,12}^+$	200
$\bar{y}_{12,3}^+$	0
$\bar{y}_{12,11}^+$	200
$\bar{Y}_{4,2}^+$	806
$\bar{Y}_{4,3}^+$	504
$\bar{Y}_{4,5}^+$	1109
$\bar{Y}_{5,2}^+$	812
$\bar{Y}_{5,3}^+$	504
$\bar{Y}_{5,4}^+$	0
$\bar{Y}_{6,1}^+$	4
$\bar{Y}_{6,4}^+$	2016
$\bar{Y}_{6,5}^+$	605
$\bar{Y}_{6,2}^-$	1580
$\bar{Y}_{6,3}^-$	1000

### 6.7.2.3 Network Equilibrium Approach

The model results using 211 and 393 iterations are given in Tables 21 and 22, respectively. These solutions are comparable to those of the maximum entropy and

the linear programming models, with respect to replicating observed link volumes on equilibrium assignment.

**Table 18. Final Trip Table Using Linear Programming Model LP(TT) - Advanced-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	691	0	1109	2400
5	0	1700	300	0	0	2000
6	500	2500	9	2000	591	5600
Total	500	4800	1000	2000	1700	10000

**Table 19. Final Trip Table Using Linear Programming Model LP(TT) - Artificial-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	691	0	1109	2400
5	0	1700	300	0	0	2000
6	500	2500	9	2000	591	5600
Total	500	4800	1000	2000	1700	10000

**Table 20. Final Trip Table Using Maximum Entropy Model**

To From	1	2	3	4	5	Total
4	0	589	720	0	1095	2404
5	0	1695	280	0	0	1975
6	500	2507	0	2000	585	5592
Total	500	4791	1000	2000	1680	9971

---

**Table 21. Final Trip Table Using LINKOD Model (211 Iterations)**

To From	1	2	3	4	5	Total
4	0	620	655	0	1121	2396
5	0	1681	339	0	0	2020
5	500	2499	0	1996	596	5591
Total	500	4800	994	1996	1717	10007

---

**Table 22. Final Trip Table Using LINKOD Model (393 Iterations)**

To From	1	2	3	4	5	Total
4	0	603	695	0	1102	2400
5	0	1698	299	0	0	1997
5	498	2501	0	1998	599	5596
Total	498	4802	994	1998	1701	9993

---

Tables 23 and 24 show a comparison of the equilibrium assigned volumes versus the observed link volumes, and the closeness of the estimated trip table to the target table, respectively. Table 25 shows the computer run times for this case.

---

**Table 23. Comparison of Equilibrium Assigned and Observed Link Volumes**

Trip Table From:	(Between Assigned and Obs. Volumes)	
	RMSE	MAE
<b>LP(TT) Model (Advanced-Start)</b>	2.53	1.31
<b>LP(TT) Model (Artificial-Start)</b>	2.53	1.31
<b>Maximum Entropy Model</b>	11.11	8.54
<b>LINKOD Model (211 iterations)</b>	11.20	8.75
<b>LINKOD Model (393 iterations)</b>	2.54	2.12

---

---

**Table 24. Closeness of Estimated Trip Table to the Target Table**

Trip Table From:	RMSE	MAE	$\phi$
LP(TT) Model (Advanced-Start)	118.83	77.09	889.93
LP(TT) Model (Artificial-Start)	118.83	77.09	889.93
Maximum Entropy Model	127.37	82.45	969.10
LINKOD Model (211 iterations)	101.92	67.00	769.83
LINKOD Model (393 iterations)	118.75	76.45	890.64

---

**Table 25. Comparison of Computer Run Times for Test Network**

	Run Time (sec)
LP(TT) Model (Advanced-Start)	8
LP(TT) Model (Artificial-Start)	10
Maximum Entropy Model	204
LINKOD Model (211 iterations)	153
LINKOD Model (393 iterations)	285

---

### 6.7.3 Case 3: Use of the Correct Trip Table as a Target

This starting table is known to be a correct final solution. Thus, it is expected that superior quality results will be produced by all the models, when this target is used. In addition to replicating observed volumes very closely, the solution table must also be the same or very close to the target. Model parameters and run settings were again maintained the same as in case 1. The results are presented below.

### ***6.7.3.1 Linear Programming Model***

#### **Advanced-Start Procedure**

The initial basis for this case is shown in Table 26. Table 27 shows the optimal trip table obtained after 45 iterations of Algorithm CGA, 34 of which were associated with Step 1, and 11 iterations with Step 2. There was no artificial variable with positive value in the final solution. This indicates that the final solution is a perfect match with the target trip table. The observed link volumes are also replicated exactly by the solution trip table.

#### **Artificial Initial Basis Procedure**

Here, the Algorithm CGA was run by simply using an all-artificial initial basis. Algorithm CGA went through 58 iterations (43 iterations associated with Step 1, and 15 iterations with step 2), before yielding the optimal solution shown in Table 28. This final solution is exactly the same as that given by the advanced-start procedure.

### ***6.7.3.2 Maximum Entropy Model***

The estimated final trip table for this approach after 10 calibration iterations is shown in Table 29.

**Table 26. Advanced-Start Basis for Linear Programming Model LP(TT) (Case 3)**

**Legitimate Variables**

Basic Variable	Value	Links Used
$\bar{X}_{6,3}^1$	100	3,2,14,17
$\bar{X}_{6,3}^2$	300	4,7,10,14,17
$\bar{X}_{6,3}^3$	500	5,8,14,17
$\bar{X}_{6,3}^4$	100	4,7,11,16,17
$\bar{X}_{6,1}^1$	500	4,6
$\bar{X}_{6,2}^1$	4100	4,7,11,15
$\bar{X}_{5,2}^1$	700	2,13,11,15

**Artificial Variables**

Basic Variable	Value
$\bar{y}_{4,9}^+$	2400
$\bar{y}_{5,10}^+$	1200
$\bar{y}_{7,9}^-$	0
$\bar{y}_{8,10}^+$	0
$\bar{y}_{9,4}^+$	2000
$\bar{y}_{9,10}^+$	1200
$\bar{y}_{10,5}^+$	1600
$\bar{y}_{10,9}^+$	800
$\bar{y}_{11,12}^+$	200
$\bar{y}_{12,3}^+$	0
$\bar{y}_{12,11}^+$	200
$\bar{Y}_{4,2}^+$	600
$\bar{Y}_{4,3}^+$	700
$\bar{Y}_{4,5}^+$	1100
$\bar{Y}_{5,2}^+$	1000
$\bar{Y}_{5,3}^+$	300
$\bar{Y}_{5,4}^+$	0
$\bar{Y}_{6,1}^+$	0
$\bar{Y}_{6,4}^+$	2000
$\bar{Y}_{6,5}^+$	600
$\bar{Y}_{6,2}^+$	1600
$\bar{Y}_{6,3}^+$	1000

### 6.7.3.3 Network Equilibrium Approach

The solution trip table using 226 iterations is shown in Table 30. When the number of iterations is more than doubled (477), a more accurate solution that is comparable to the maximum entropy solution is achieved (Table 31).

**Table 27. Final Trip Table Using Linear Programming Model LP(TT) - Advanced-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	700	0	1100	2400
5	0	1700	300	0	0	2000
6	500	2500	0	2000	600	5600
Total	500	4800	1000	2000	1700	10000

**Table 28. Final Trip Table Using Linear Programming Model LP(TT) - Artificial-Start Procedure**

To From	1	2	3	4	5	Total
4	0	600	700	0	1100	2400
5	0	1700	300	0	0	2000
6	500	2500	0	2000	600	5600
Total	500	4800	1000	2000	1700	10000

**Table 29. Final Trip Table Using Maximum Entropy Model**

To From	1	2	3	4	5	Total
4	0	593	713	0	1096	2402
5	0	1695	287	0	0	1982
6	500	2505	0	2000	592	5597
Total	500	4793	1000	2000	1688	9981

---

**Table 30. Final Trip Table Using LINKOD Model (226 iterations)**

To From	1	2	3	4	5	Total
4	0	572	748	0	1074	2394
5	0	1724	241	0	0	1965
5	498	2503	0	1997	603	5601
Total	498	4799	989	1997	1677	9960

---

**Table 31. Final Trip Table Using LINKOD Model (477 iterations)**

To From	1	2	3	4	5	Total
4	0	582	725	0	1088	2395
5	0	1713	269	0	0	1982
5	499	2504	0	1998	599	5600
Total	499	4799	994	1998	1687	9977

---

Tables 32 and 33 show a comparison of the equilibrium assigned volumes versus the observed link volumes, and the closeness of the estimated trip table to the target table (which is also a correct trip table), respectively. From Table 32, it is seen that the convergence rate obtained in RMSE and MAE values with respect to replicating observed volumes on equilibrium assignment is slow for the LINKOD model. Finally, computer run times for this case are shown in Table 34.

---

**Table 32. Comparison of Equilibrium Assigned and Observed Link Volumes**

<b>Trip Table From:</b>	<b>(Between Assigned and Obs. Volumes)</b>	
	<b>RMSE</b>	<b>MAE</b>
<b>LP(TT) Model (Advanced-Start)</b>	2.49	1.64
<b>LP(TT) Model (Artificial-Start)</b>	2.49	1.64
<b>Maximum Entropy Model</b>	6.74	4.78
<b>LINKOD Model (226 iterations)</b>	12.90	8.17
<b>LINKOD Model (477 iterations)</b>	6.91	5.16

---

**Table 33. Closeness of Estimated Trip Table to the Target (Correct) Table**

<b>Trip Table From:</b>	<b>RMSE</b>	<b>MAE</b>	$\phi$
<b>LP(TT) Model (Advanced-Start)</b>	0.0	0.0	0.0
<b>LP(TT) Model (Artificial-Start)</b>	0.0	0.0	0.0
<b>Maximum Entropy Model</b>	6.86	5.00	55.28
<b>LINKOD Model (226 iterations)</b>	26.72	17.82	201.94
<b>LINKOD Model (477 iterations)</b>	14.29	9.73	108.58

---

**Table 34. Comparison of Computer Run Times for Test Network**

	<b>Run Time (sec)</b>
<b>LP(TT) Model (Advanced-Start)</b>	9
<b>LP(TT) Model (Artificial-Start)</b>	10
<b>Maximum Entropy Model</b>	214
<b>LINKOD Model (226 iterations)</b>	165
<b>LINKOD Model (477 iterations)</b>	346

---

## **6.8 Discussion of Results**

In the light of the above results, two issues arise for critical discussion. The first issue is the degree to which the model fulfils its objectives. The second is the issue of computational effort required by a model, which has implications in real-time applications. These two issues are discussed below in detail, with reference to the results from the models under comparison.

Since a model is to be judged based on its ability to match the prior trip table as closely as possible, and at the same time its capability to replicate the observed flows when assigned to the network, a model will be judged as good if it can achieve both the above objectives. However, a prerequisite to this is that such a solution must exist in the first place. In reality, due to inconsistencies in link flows, and due to the invalidity of the assumption that all travellers are aware of the shortest paths, some of the assumptions such as consistency of link volumes, and existence of equilibrium traffic flow, are rendered invalid. In such a situation, a "correct" solution may not exist at all. A good model must be able to accommodate such inconsistencies, and be able to produce a solution that fulfils the objectives to the greatest extent possible. Both the linear programming and the LINKOD models have this required quality. The maximum entropy model, however, requires conservation of flow at intersections. Thus traffic volume inconsistencies may pose problems in the application of this model.

Examining the results of case 1 ("no-prior-information" target) runs, the solution tables from the LP(TT) models (both advanced-start and artificial-start cases in Tables 9 and 10, respectively) replicate the observed link volumes very closely on equilibrium assignment. In fact, the table conforms to an exact user-equilibrium

solution, replicating all observed volumes exactly. This is verified by examining the final objective value and the path-decomposed flows at optimality. However, the values of RMSE and MAE comparing the equilibrium assigned and observed volumes are nonzero due to convergence tolerances in the assignment routine. The table resulting from the maximum entropy model produces volumes fairly close to observed volumes on equilibrium assignment, with RMSE and MAE values of 10.33 and 6.58, respectively. The RMSE and MAE values for LINKOD model output are 9.62 and 7.82, respectively, for the table resulting from 238 iterations. This indicates that the solution is not in conformity with equilibrium. However, if the number of iterations is increased to 297, greater accuracy is achieved and the solution table is closer to equilibrium. Clearly, the linear programming model results are most superior in this respect, and represents a perfect fit.

Observing the closeness of solution tables to the target table (Table 15), all the models produce solutions that are distant from the target. This is to be expected, since the target is a "no-prior-information" table containing very little information about the trips. Thus, not much importance can be given to these results. The plots of the cell values of the different solution tables along with those of the target table (Figure 8) indicate clearly that the target is far removed from reality, and shows the existence of similarity in trip patterns as modelled by different approaches.

When a better quality ("Relatively Small Errors") target table is provided as input (case 2), the solutions from all the models improve. With regard to replicating observed volumes and extracting equilibrium solutions, the LP(TT) model retains its superiority. This fact is endorsed by statistics in Table 23, giving the deviations between the equilibrium-assigned and the observed volumes. Again, the LP(TT) model replicates link volumes exactly and is in perfect equilibrium, as verified similar to case 1. Maximum entropy result is, however, not close to equilibrium. While 211

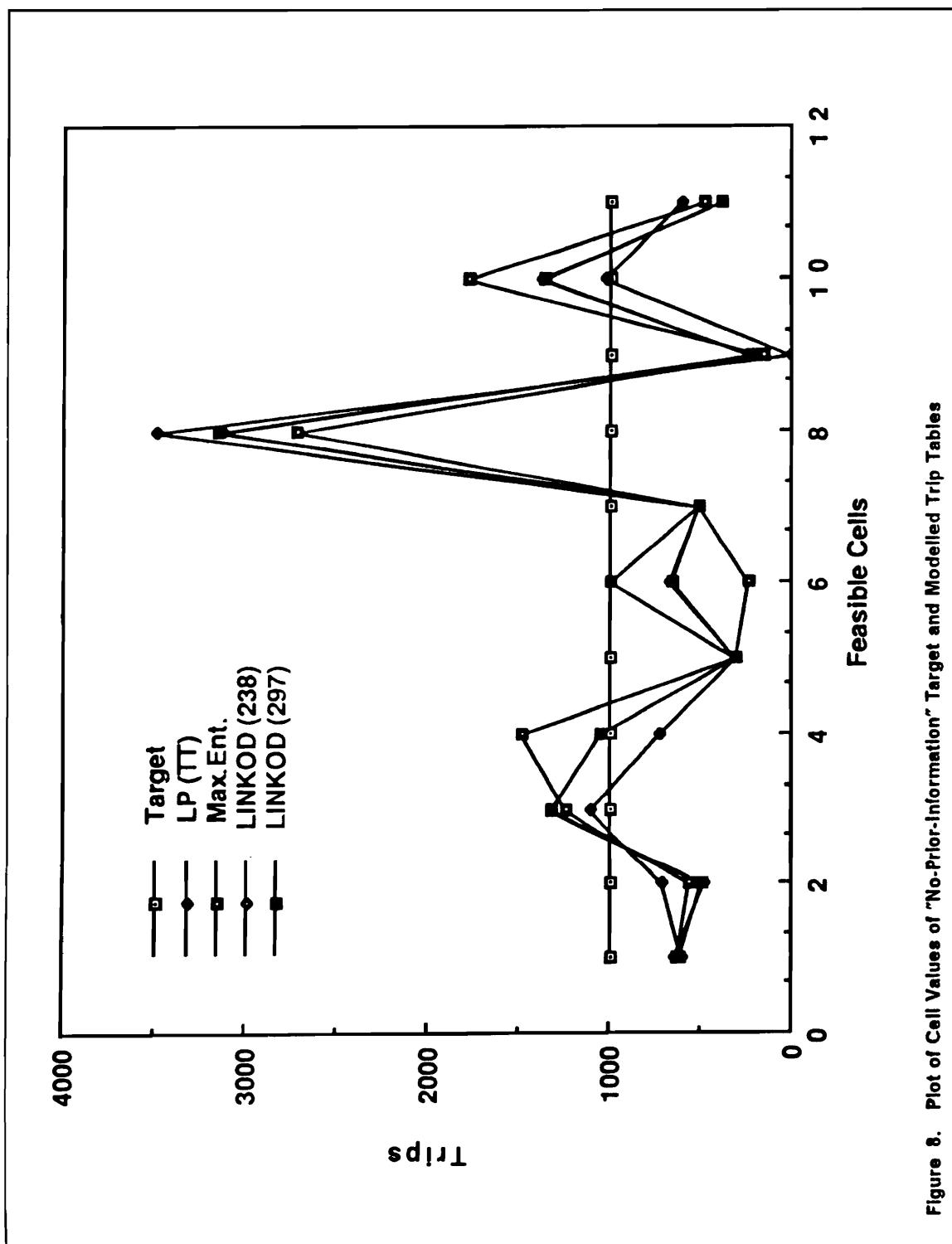


Figure 8. Plot of Cell Values of "No-Prior-Information" Target and Modelled Trip Tables

iterations of the LINKOD model produces a non-equilibrium solution, 393 iterations produce a solution that is very close to equilibrium. Examining the closeness of solution table to the target table, improvement is seen in the case of all the models. LP(TT) solution is as close to the target as the the LINKOD (393 iterations) solution, and better than that of the maximum entropy model (Table 24), an improvement over case 1 results. Figure 9 shows the cell value plots for the different models and the target table.

Case 3 targets the solution to the "correct" trip table itself. The LP(TT) model produces excellent results. In addition to perfect replication of link volumes under equilibrium conditions, as in the earlier cases, it also matches the target table perfectly, as indicated by zero values for the *RMSE*, *MAE*, and  $\phi$  statistics in Table 33. The results from the LINKOD and the maximum entropy models are not as competitive. A noteworthy point here is that the correct target table is modified by the maximum entropy and the LINKOD models to produce close but incorrect solutions. This is due to algorithmic deficiencies. Since, a correct table is given as seed/target table, strictly speaking, the maximum entropy and the LINKOD models must not alter them. But these algorithms assign the given table at the first step. This assignment does not replicate the observed volumes. Hence, the given table is subsequently altered, and ultimately the final solution is not as accurate as the initial table. Through personal communication, Dan Beagan has recently corrected this deficiency by providing an option to carry out equilibrium assignment for the first iteration in the maximum entropy model. This appears to give improved results, but tests are still underway. The linear programming model, on the other hand, does not modify the target table, and uses the cell values only as constraints. The cell value plots for the different models and the target table are shown in Figure 10.

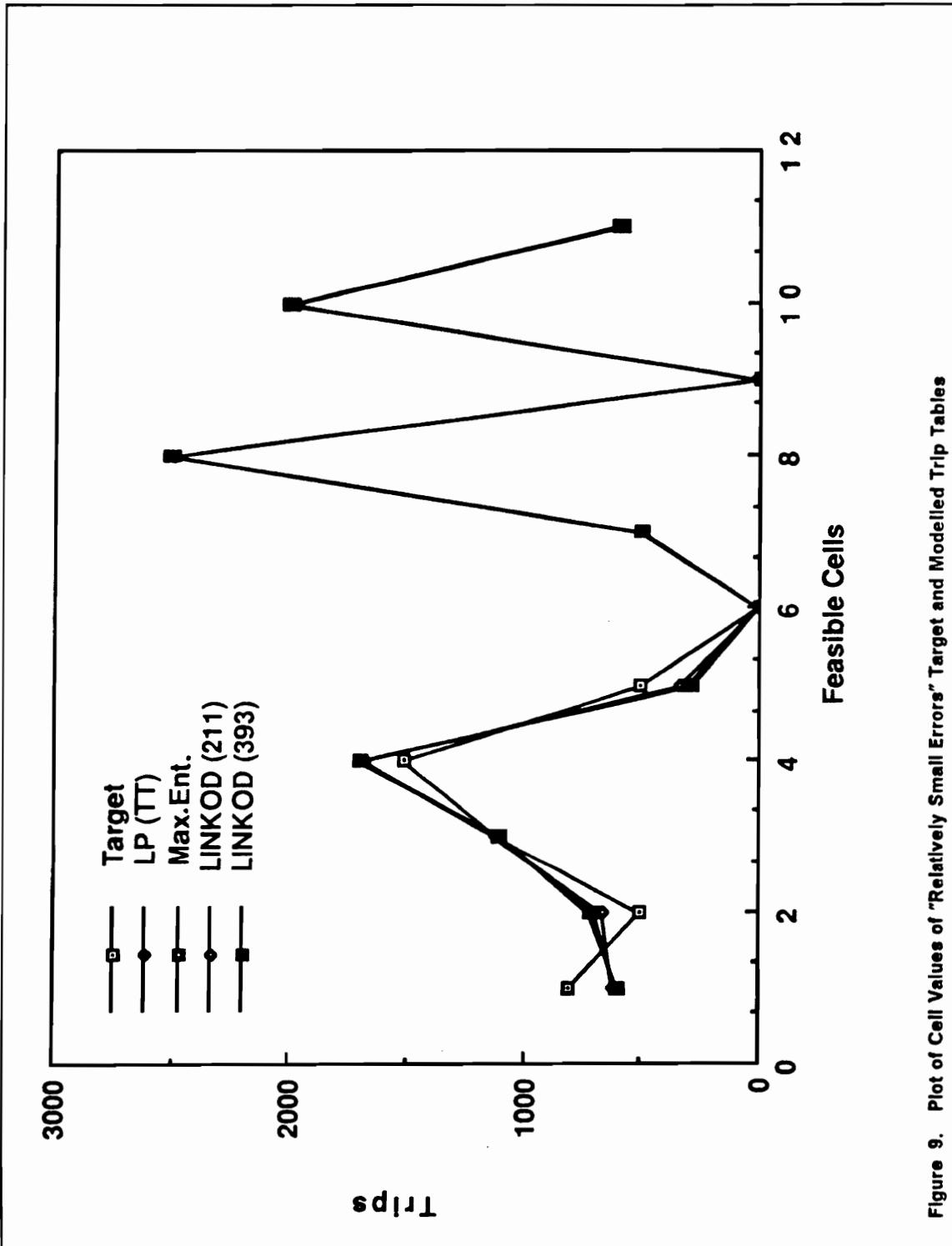


Figure 9. Plot of Cell Values of "Relatively Small Errors" Target and Modelled Trip Tables

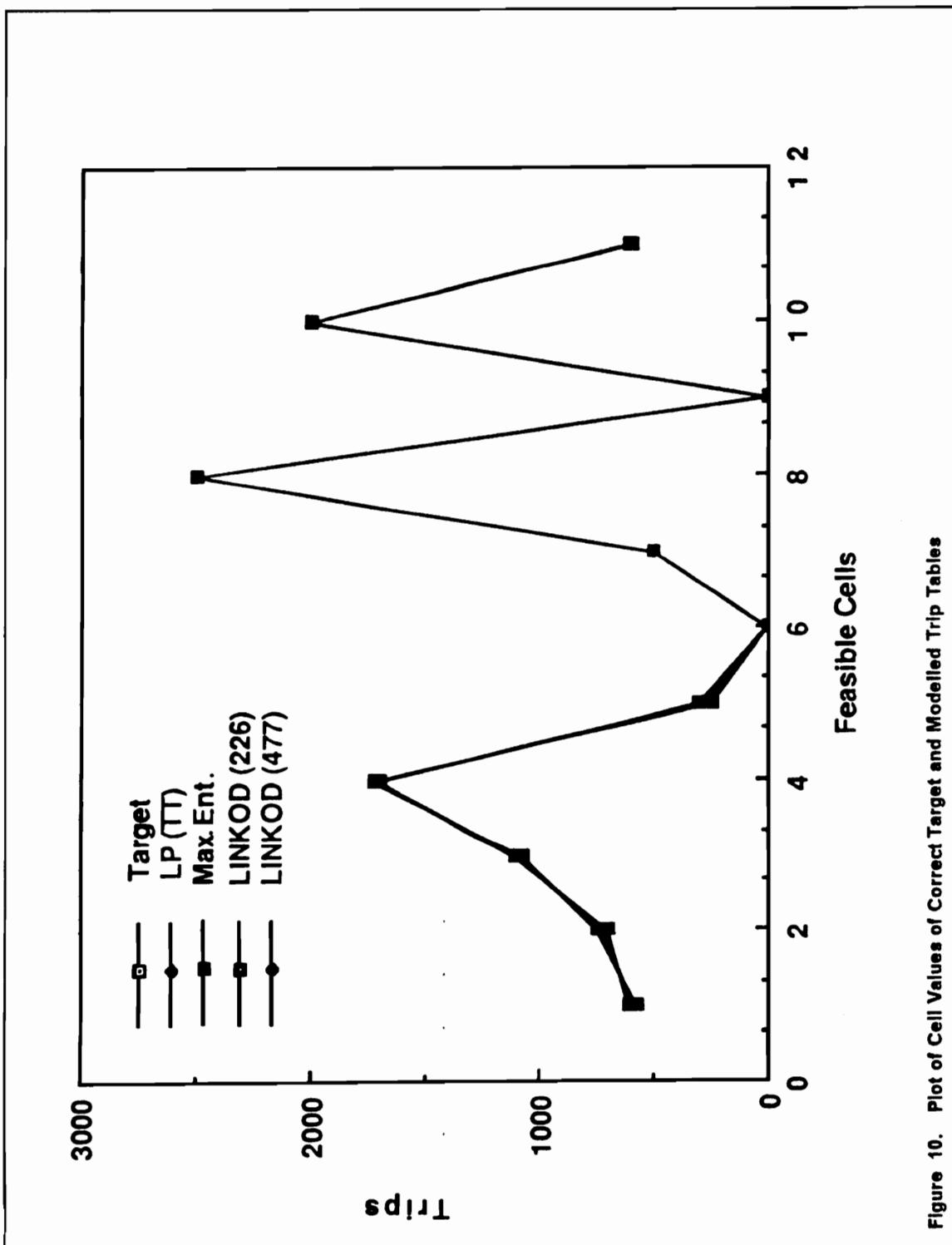


Figure 10. Plot of Cell Values of Correct Target and Modelled Trip Tables

The issue of application of model in real-time applications relies heavily on the speed of execution of the models on a computer. In this respect, the linear programming model exhibits clear superiority when compared to the other two models, in the light of tests on the above sample network. Figure 11 depicts graphically the run times for the three models under comparison, as applied to the test network. This bar graph illustrates case 2, which employed a "relatively small errors" target trip table. This case is considered for comparison, since an available old trip table may be taken as representative of "relatively small errors" target. Moreover, a "true" trip table may never be available for use.

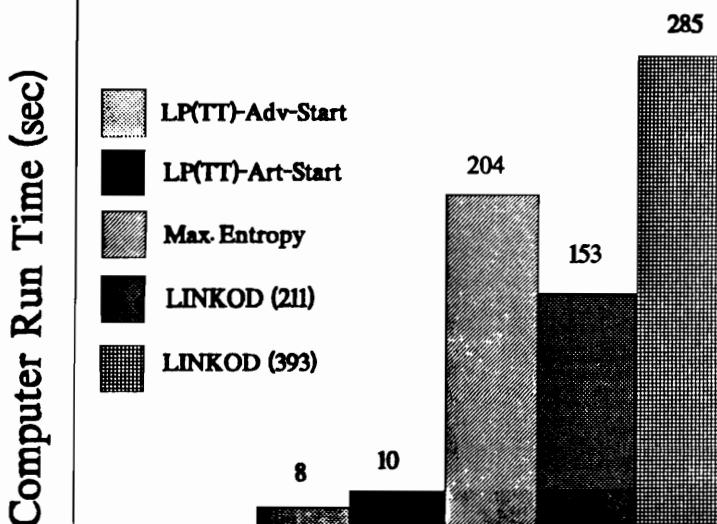


Figure 11. Computer Run Times for Test Network (Case 2)

Clearly, the speed of execution of the LP(TT) model on a PC is relatively impressive, being much faster than the LINKOD and the maximum entropy models, for the sample test network. While the run time for the LINKOD model using 211 iterations is reasonable, although still far exceeding the LP(TT) run time, the solution obtained falls short of necessary accuracy tolerances. For the sample network, since

the observed link volumes are known to be consistent, and the existence of an equilibrium solution table is confirmed, volume error must be significantly small. This is achieved by increasing the number of iterations to 393, which leads to more accurate results, which are comparable to that of LP(TT). However, the run time is very high, as shown in Table 25. The run time for the maximum entropy model is slightly lower, but much higher than for the LP(TT) model. This time would increase further if a greater number of iterations are used for better results. Thus, for the maximum entropy and the LINKOD models, a trade-off has to be sought between the quality of results and run time. This aspect is not such a major concern for the linear programming model, although one can envisage premature termination for large sized problems based on such a trade-off. The linear programming approach produces results of comparable or superior quality with far lesser execution effort, as shown by test runs.

## 6.9 Tests on Additional Networks

Further tests were conducted on four more hypothetical networks to confirm the above conclusions. One of these networks was obtained from the literature, and the other three were arbitrarily generated. The run times and the degree to which observed link volumes are replicated on assignment are tested. Table 35 summarizes these results, including those for the detailed example ("Corridor Network") presented earlier.

These results further endorse the claim that the LP(TT) model is superior with respect to both run time and replication of observed volumes. The tests show that

**Table 35. Run Times and Replication of Link Volumes for Additional Test Networks**

Test Network (Appx. D)	Source	Model	Run Time (sec) +	Model Iterations	Modelled vs. Obs. Link Vols.		Eqm. Assigned vs. Obs. Link Vols.	RMSE	MAE	Remarks
					RMSE	MAE				
1	Gur et al. (1980)	LP( $\Pi$ )@ LP( $\Pi$ )@@ Max. Ent. LNKOD	8 10 204 285	41 50 $10^*$ 393	0 0 U 2.76	0 0 U 2.27	2.53 2.53 11.11 2.54	1.31 1.31 8.54 2.12	1.31 1.31 8.54 2.12 (Case 2)	Detailed Results in Sec. 6.7 & 6.8 (Case 2)
2	Nguyen (1977)	LP( $\Pi$ )@ LP( $\Pi$ )@@ Max. Ent. LNKOD	3 5 164 139	13 53 $10^*$ 217**	0 0 U 4.10	0 0 U 3.30	0 0 U 5.80	4.81 4.81 6.80 5.80	3.91 3.91 4.59 4.39	
3	Arbitrarily Generated	LP( $\Pi$ )@ LP( $\Pi$ )@@ Max. Ent. LNKOD	1 2 127 55	5 30 $10^*$ 122**	0 0 U 29.47	0 0 U 27.75	0 0 U 17.24	1.83 1.83 16.63 30.02	1.29 1.29 13.27 18.05	
4	Arbitrarily Generated	LP( $\Pi$ )@ LP( $\Pi$ )@@ Max. Ent. LNKOD	3 3 202 24	29 30 $10^*$ 79**	0 0 U 4.23	0 0 U 4.23	0 0 U 1.97	0.46 0.46 28.03 4.13	0.27 0.27 21.53 2.11	
5	Arbitrarily Generated	LP( $\Pi$ )@ LP( $\Pi$ )@@ Max. Ent. LNKOD	1 2 102 43	5 22 $10^*$ 119**	0 0 U 3.07	0 0 U 2.78	0 0 U 3.07	2.17 2.17 7.25 3.44	1.51 1.51 5.61 2.80	

+ : On IBM/PS2 Model 507Z, 10 MHz

@ : Advanced-Start Procedure

@@ : Artificial Initial Basis Procedure

U : Unavailable

\* : 10 iterations were used as suggested by the model developer

\*\* : Very slow convergence rate obtained in RMSE and MAE values for further iterations

in certain examples, the maximum entropy and the LINKOD models are slow to converge to the same degree of accuracy, as obtained from the LP(TT) model. In such cases, the algorithms were terminated well short of optimality. The LP(TT) model, on the other hand, quickly produced exact solutions without facing such problems.

The closeness of the LP(TT) model results with those of other models, and its ability to replicate observed flows on links when assigned using the user-equilibrium principle, are clear indicators of the validity of the model. Further investigation, testing and evaluation on bigger, real networks will shed more light on the reliability of the model and its adaptability. There is, however, a need for testing the models on a network for which a true O-D table is available. This will enable a correct evaluation of the models. In the absence of this, the superiority of a model cannot be ascertained.

# **7.0 APPLICATION OF THE LINEAR PROGRAMMING MODEL TO A NORTHERN VIRGINIA NETWORK**

## **7.1 *Introduction***

The overall goal of this research project under the US DOT/VDOT University Transportation Centers Program is to evolve diversion strategies for real-time applications, with the specific aim of addressing the congestion problem in Northern Virginia. Thus, the logical step consistent with the research approach, after the development of an approach for synthesizing the O-D trip table, is to apply it on the network of interest. Such an exercise was conducted to gain useful insights into the behavior of the model on a real network, and to assess its scope for achieving the overall goal. Although an actual real-time application of the model would warrant addressing many related issues, it is worthwhile to apply the linear programming

model in its present state, as a non-real-time exercise. With this motivation, a real network was chosen for the case study, and the model applied. The choice of the study area, data collection, application of the model, implementation difficulties and their circumvention, preliminary results, and further scope of work, are all discussed in the following sections.

## **7.2 *The Case Study***

The Northern Virginia urban area, which includes Alexandria City, Fallschurch, Arlington County, and portions of Fairfax County, is facing severe congestion problems, while the resources have not kept pace with the demand. The problem is more pronounced in the interstate systems of I-395, I-66, I-95, and I-495 in the region. The Virginia Department of Transportation (VDOT), with an intent to reduce congestion levels and improve the operational efficiency of the interstates I-395 and I-66, has installed the Traffic Management System (TMS) at Arlington. It has also implemented demand restriction policies by reserving exclusive lanes on I-66 and I-395 for high occupancy vehicles (HOVs) during peak periods.

### **7.2.1 *The Traffic Management System (TMS)***

The TMS is a computerized freeway surveillance and control system for monitoring and controlling traffic flow mainly along the two interstates, I-66 and I-395 (VDOT). A portion of I-95 is also currently being brought under the jurisdiction of

TMS. The TMS is designed to carry out surveillance of traffic through closed circuit TV and loop detectors, detect incidents, meter ramps and control traffic to reduce congestion. The system uses complicated detection algorithms in conjunction with loop-detector information and checks if the speed and vehicle occupancy cross the threshold values defined to indicate congestion. In addition to 550 traffic counters imbedded in the pavement of I-395 and I-66, 48 closed circuit television cameras monitor the traffic on these interstates, and help the traffic controllers with a visual picture of the situation. On detecting an incident, and confirming it through CCTV's, the controllers alert the police and the media. As an information service to unaffected motorists, who are likely to pass through the incident site, the center displays the location of the incidents and the status of traffic through variable message signs as in Figure 12, which number 96 along I-66, I-395 and I-95.

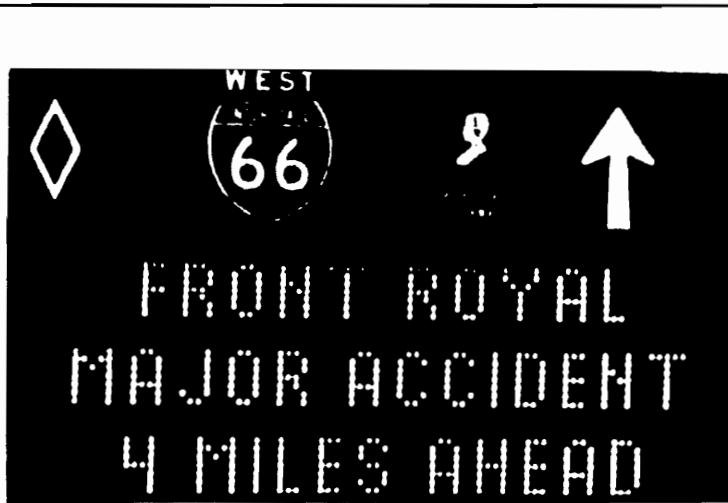


Figure 12. A Variable Message Sign Informing Motorists About a Major Accident (Source: VDOT)

Conspicuous in the above message is however, the absence of any advisory guidance on the best alternate routes to be taken by the motorists. Currently, the TMS does not have the capability to determine best alternate routes for the motorists.

This research work is intended to provide the ability to the TMS to offer such a guidance, through the use of algorithms for determining optimal rerouting plans. To accomplish this in real time, the sensor data received from the field by the control center could be advantageously used. Yet another ambitious task would be to perform area-wide diversion analysis that would include not only the interstates, but also the major arterials in the region. Currently, the TMS does not have the capability to obtain traffic flow information from arterial streets in real time. But this facility is feasible by linking the other two control centers in Arlington county and in Alexandria City, with the TMS. The link between the Arlington county center and the TMS already exists, but data transfer procedures are not yet operational. When the links between the centers are operational, the three systems together provide scope for mutual exchange of data, which will enable the TMS center to develop area-wide diversion plans.

### **7.2.2 Study Network**

Responding to the present concern of VDOT over the growing congestion problem in the Northern Virginia area, it was deemed appropriate to apply the current research work to this area. Recognizing the importance of performing area-wide analysis, and with the intent of utilizing the TMS as the focal point of control for diversion planning, an appropriate study network had to be delineated. In choosing the network, one of the considerations was to include the portions of the interstates under the control of TMS. This is to facilitate displaying diversion messages through variable message signs during incident management, and for the appropriate control of traffic. Another consideration was the computational effort that would be demanded

to run the software models to achieve the objectives. The size of the network had to be restricted to a size that is manageable, and which would not demand prohibitive computational effort. Since the whole Northern Virginia network would be too big for our model application, a smaller area had to be chosen. If a smaller area is chosen, external-external trips passing through the area, external-internal trips, and internal-external trips would have to be considered in the analysis. To account for external trips, both entering the network and leaving it, the volume of traffic on the links crossing the cordon line are considered. Thus, external zones for the study area are to be located on the links crossing the cordon, and just outside of it. The corresponding link flows are considered as the zonal flows. For the cordon line to be sensible and meaningful, it must clearly demarcate the interior of a region from the exterior. Conventionally, this demarcation is provided by the town boundary, or a river, for example, which provide distinct crossing points from the exterior to the interior and vice-versa. This requisite for a cordon was also an influencing factor in the delineation of the study network.

With the above considerations, the chosen area comprised the road network between the capital beltway (I-495/I-95) and the District of Columbia, on the Virginia side (Figure 13). The beltway formed a suitable cordon. Only the primary roads (route numbers 600 and above), and the Interstate highways were included in the network structure, in order to contain its size for model application. Field visits to the study area were conducted. Extensive sensor data from the three traffic control centers in the region were collected for model application. Based on the availability of data, and the structure of the road system, a network aggregation was performed for model application.

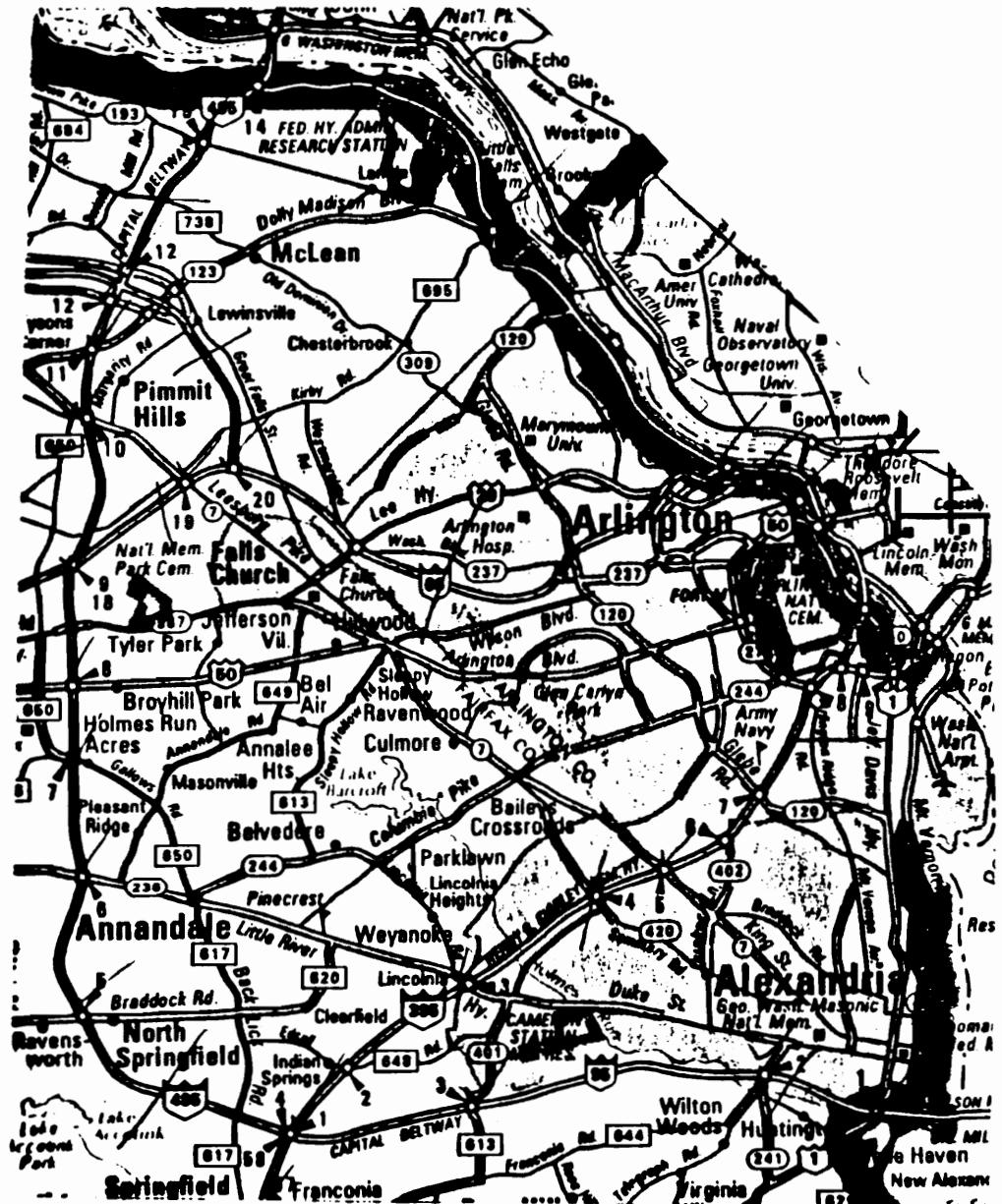


Figure 13. Northern Virginia Study Area

### **7.2.3 Network Aggregation, Coding and Data Reduction**

#### **7.2.3.1 *Network Aggregation***

The network was aggregated to the size shown in Figure 14 for several reasons. For the regions covered by the three control centers, even though extensive sensor data were collected, many of the links could not be considered, either due to the absence or due to non-operation of sensors. Also, portions of the network not covered by the control centers do not have sensors. The aggregated network consists of 47 zones, 60 intersection nodes, and 316 directed links.

#### **7.2.3.2 *Link Impedances***

The linear programming model needs as input, the data to compute the impedance offered by the network links. As stated earlier, the BPR link performance function is built into the model to compute the link travel time, given the free flow travel time, capacity, and the flow volume. For this real network, aggregation has been carried out to combine, in many cases, a group of links into one link. It is incorrect to assume a single impedance function for a group of links with varying flows and physical characteristics. Also, the above function determines only the travel time for the link, and not the intersection delays. Since, intersection delays on arterials have a considerable influence on the overall travel time, it would be unwise to ignore these delays in a real network. On the other hand, a consideration of individual intersections will lead to a microscopic analysis, increasing the computational burden tremendously. A compromise was sought by using the average

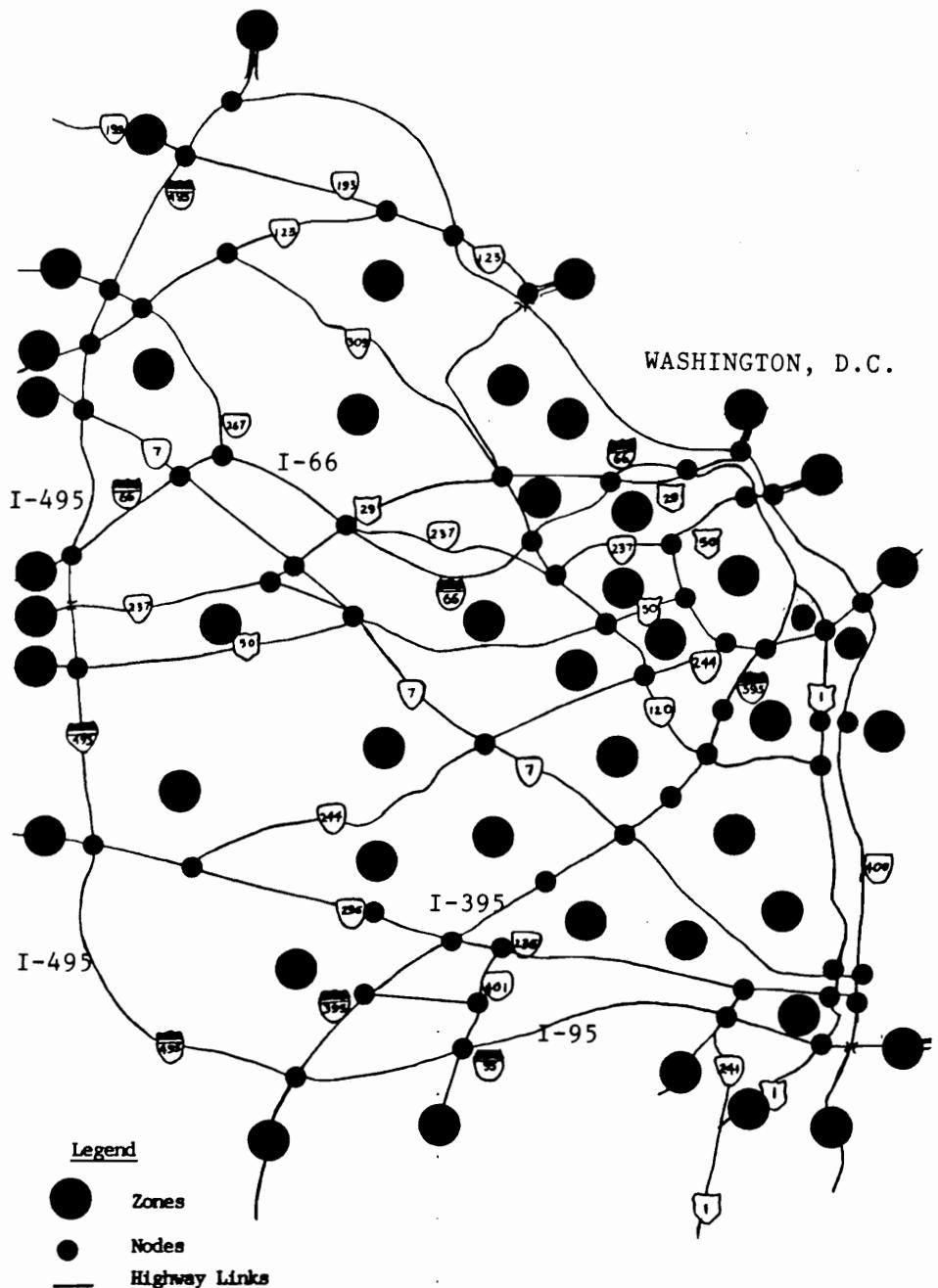


Figure 14. The Study Network

speeds for different regions of the network, as provided by the Metropolitan Washington Council of Governments. The look-up table provided by the agency for routes in different regions of the area was used. The link travel times are then easily computed by dividing the link length by the average speed of progression.

#### ***7.2.3.3 Network Coding***

Network coding involves numbering the zones, nodes, and links in a systematic manner. For efficient computer storage, the following scheme of numbering was adopted. Destination zones were numbered first, starting from one. After all the destination zones were assigned numbers, origin zones were numbered starting with the next higher number to the last destination zone number. (For the study network, since all zones serve as both origin and destination, the task of numbering origin zones separately did not arise). Next, intermediate nodes (intersection nodes) were assigned numbers, starting with the number next to the last number assigned to an origin zone. Starting from zone 1, all the nodes that are connected to the zone in question were numbered sequentially. After all the zones were exhausted, the next higher number, which was allotted to a node was considered, and any unnumbered nodes connected to it through links were given increasing numbers. This process was continued until all the nodes were numbered. The links were not numbered separately. They are internally numbered by the model, based on node numbers. A link is represented by its beginning ("From") and ending ("To") node numbers. Link details were input in the data file such that links with the smallest "From" and "To" nodes were at the top of the file. The links then take on numbers according to this order.

#### **7.2.3.4 Data Reduction**

For the two interstates, I-66 and I-395, under the control of the TMS, sensor data for all the links were available. Morning peak hour data for 7-8 A.M. was considered. However, for other links of the network for which the sensor data was unavailable, either the 24-hour count conducted by the Virginia Department of Transportation (VDOT), or the Annual Average Daily Traffic (AADT) were used. AADT was reduced to peak hour volume through appropriate conversion factors. Thus, the data set used was non-homogeneous. However, further refinement of the model and enhancement of the data sets are proposed for future research.

#### **7.2.4 Location of Zones**

The Metropolitan Washington Council of Governments (1987) has subdivided the metropolitan area, which includes the study area, into impact assessment districts, as shown in Figure 15. These districts are further divided into zones, which could be adopted for this study. However, there are too many zones in the study area. To limit the size of the network, the impact assessment districts were considered as the corresponding zones for the purpose of the study. Since in this study, only a sub-area has been considered, the zones outside the cordon were established at major entry/exit points. The traffic volume entering or leaving the study cordon were hence assumed to be the trip productions or attractions, respectively, at these zones. For zones within the cordon, centroid connectors were established by judgement.

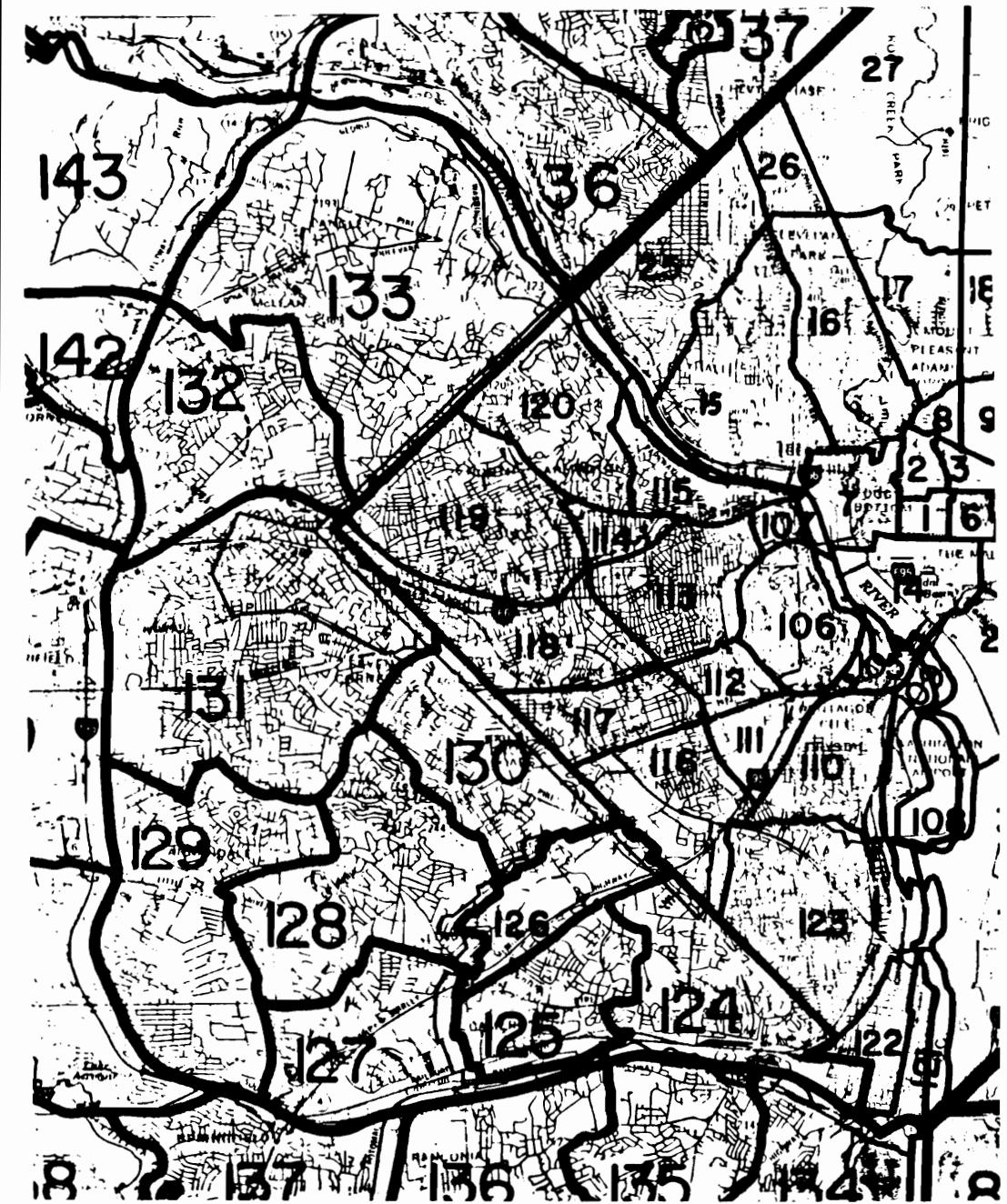


Figure 15. Impact Assessment Districts, as Defined by the Metropolitan Washington Council of Governments (1987), for the Study Area

## **7.3 Overall Approach for Diversion Planning**

The main intent in this application is to evolve real-time diversion strategies in the event of accidents or congestion-causing events. Incident management usually were, until now, limited to informing the motorists about the incident through television, radio channels, and variable message signs, and clearing the incident spot as quickly as possible. Area-wide diversion analysis and planning, which could provide a more appropriate and beneficial strategy, was not considered. This formed the basis for defining the research approach. The approach envisages utilization of real-time sensor data from the three traffic control centers in the area, thus enabling area-wide analysis and control. The approach is better illustrated by considering the following scenario.

### **7.3.1 The Scenario**

A major chemical spill has occurred on I-395 (N) (marked with black circle in Figure 16) during a morning peak, blocking that direction completely to traffic. Though emergency response has begun immediately, the incident being major, the restoration of the interstate to normal traffic is expected to take a few hours. Traffic has begun to queue up at the blocked location. The TMS at Arlington, which monitors this section of the interstate has acted immediately on detecting the incident, and has alerted the police. It has also sent advisories on the incidents and has displayed "Major Accident Ahead" sign through variable message electronic boards located on

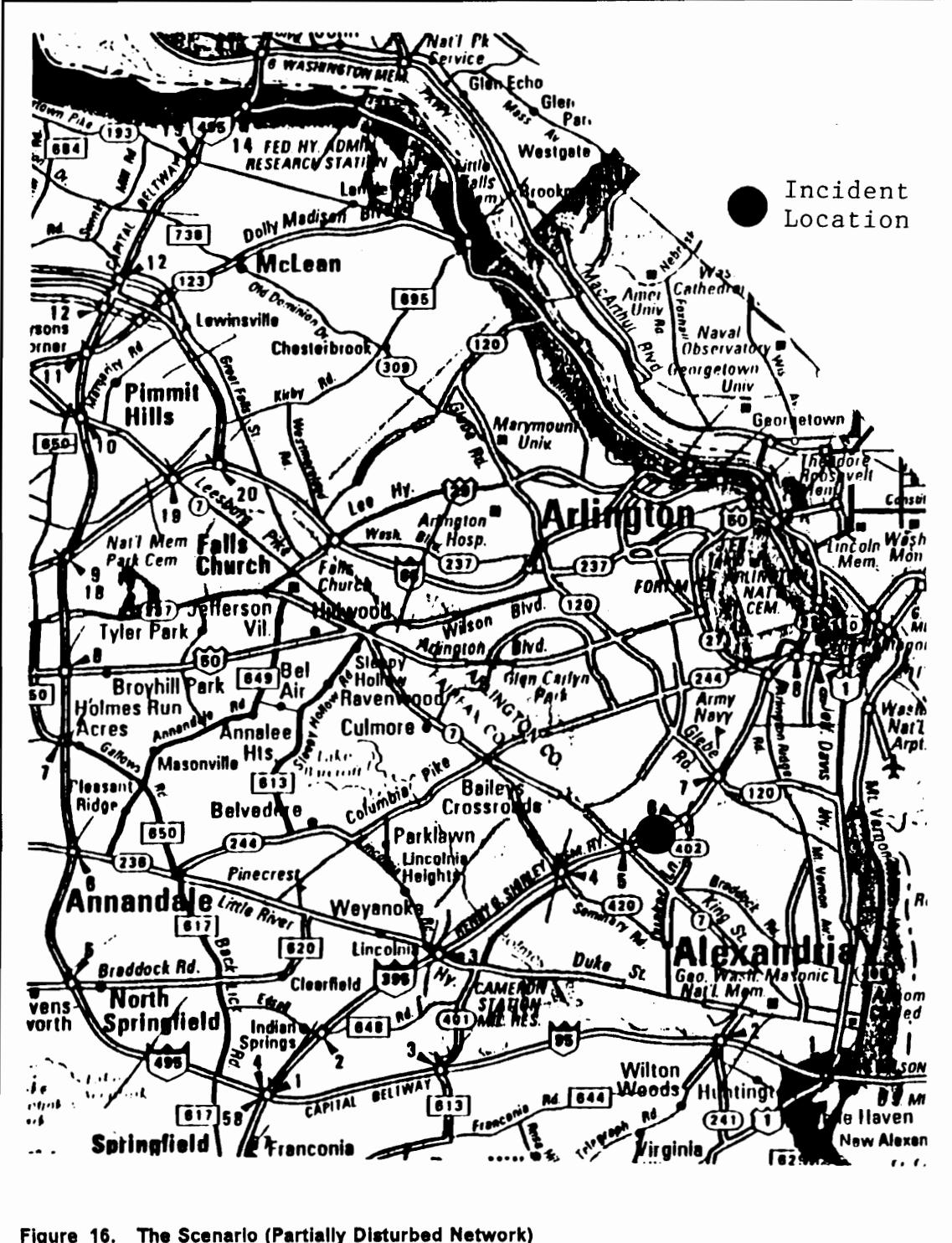


Figure 16. The Scenario (Partially Disturbed Network)

the interstate. Advisories and incident location details have been broadcast through motorist communication channels.

There are three categories of motorists that may be directly or indirectly affected in this scenario. These are as follows.

1. **Motorists near the incident location:** This is the most affected group, since these motorists will have to face most uncertainties and delays. Diversion of these motorists assumes a first priority, and the efficacy of such an operation rests in determining exit locations quickly, so as to minimize delays. In the scenario case, travellers on I-395N who have crossed the intersection with Rt. 7 constitute this group.
2. **Motorists farther from the incident location, but expected to pass through the closed portion:** Another group of motorists is constituted by those who have already commenced their trip and are proceeding toward the incident location, unaware of the incident, but are currently away from it. This is the interest group for this research. It is the intention to provide useful diversion information to these motorists so that they do not face blockages, queues, and diversions near the incident location. In this case, those who are currently on I-395N south of the incident, and those from arterials proceeding to join the above section of the interstate would be affected, if not advised of alternate routes.
3. **Motorists yet to commence their trips:** The third category of motorists who will be affected due to the incidents are the ones who are yet to make travel commitments, but would generally pass through the incident location, if they were to make their trips. The strategy is to delay their departures and/or alter their route plans.

The above three groups have a combined influence on the traffic flow, and induce dynamic behavior to the flow. The interaction between the changing conditions of flow and motorists' choice of routes must be monitored for devising effective diversion strategies.

## **7.4 The Approach**

The approach calls for an integrated action, both from the users' and operator's stand points. Appropriate diversion strategies need to be adopted for specific groups of motorists. Motorists who have reached the incident location (group 1) do not have much choice, and must find a way to bypass the incident. Group 2 motorists have the option of choosing among alternate routes. However, they need to be guided to take the best alternate route to their destinations, depending on the status of traffic in the network at the time of the incident. For the user-optimal approach, this would involve determining their destinations. An attractive scheme is to utilize the concept of determining O-D trip patterns utilizing link volumes at the time of the incident, as stated earlier. This can be achieved by employing the linear programming model developed in this research. The travellers who are yet to begin their trips are another potential source for reducing congestion. Work places and homes are potential targets for alerting the motorists about the incidents and rerouting plans. The messages can be relayed through television, or even telephone cables can be utilized to transmit visual data either on television or on computer screens.

The strategy of this research is to employ the linear programming model to synthesize the O-D trip table for the study area, using normal link volumes for the

period of the day when the incident occurs. The resulting table is taken to be representative of the trip patterns at the time of the incident. The model output would also give complete information of the routes motorists would take normally, under undisturbed conditions. This key information, along with the number of trips for various O-D interchanges will enable determining the traffic that is going to be affected by the incident. The next step is to evolve diversion plans by assessing the various diversion options with the objective of minimizing delays and detours. This can be accomplished by assigning the O-D trips to the network. Since, a consideration of the dynamics of traffic flow with time enables a more realistic assessment, a dynamic traffic assignment model must be used. The TMS can act as the focal point, where area-wide traffic data can be collected through sensors in its own jurisdiction, and from the other two control centers. Various algorithms to achieve the above objectives can also be run on a computer housed at the TMS.

The diversion plans/strategies chosen can then be implemented by posting signs to direct traffic, radio and TV advisories, deploying personnel to monitor traffic, and by employing Transportation Systems Management (TSM) actions, if warranted. The changeable message signs and other means of communication can be then employed to advise traffic of alternate routes. As diversion of traffic gets into progress, the network conditions must be constantly assessed to appropriately evolve further diversion plans.

The overall methodology for implementation, as a joint effort by the three control centers in the region is depicted in Figure 17.

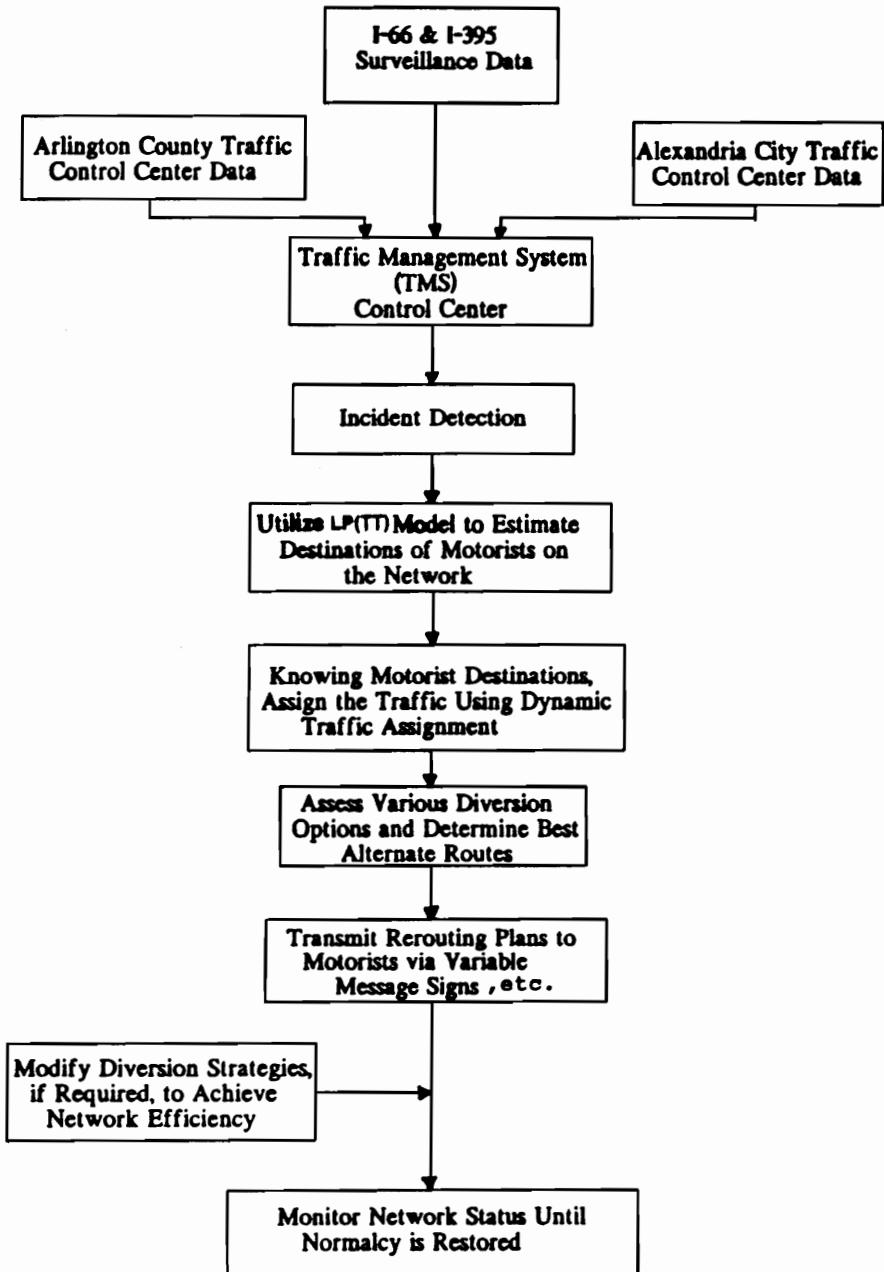


Figure 17. Proposed Approach for User-Optimal Diversion Planning

## **7.5 Implementation Strategies and Model Refinements**

Several implementation strategies and model refinements have been incorporated as a result of experiences in running the model on the real network. These were necessitated due to reasons ranging from network details to computational difficulties encountered during model execution. The issues affecting the implementation are discussed below.

### **7.5.1 Accountability of Centroid Connector Links with Unknown Volumes**

As stated earlier, the centroids for the internal zones had to be connected to the nearest links through centroid connectors. Since the larger areas representing the impact assessment districts were taken as zones for the study, suitable highways that could play the role of centroid connectors could not be found. Hence, centroid connectors to these zones had to be arbitrarily fixed. The traffic volumes on these links are obviously unknown. Hence, no limits can be set on the productions/attractions of the zones linked to these connectors. Consequently, these links are permitted to carry unrestricted amount of traffic volume, thus providing free access to highway traffic. This requires that the cost of travel on these links must be set at a negligible value. This is easily accomplished by setting the length of these links to a small value (0.01 mile) and the speed at a higher value (60 mph). This case is easily accommodated by the linear programming model through a simple strategy

in the algorithmic implementation, by not penalizing the artificial variables corresponding to these connector links, in the objective function, as follows.

Initially, while preparing the observed volume data file, the volumes corresponding to the centroid connectors (dummy links) are set as zero. While reading this data file, a flag is established to indicate whether a link is dummy or not. In the initialization process, the right hand sides corresponding to the constraints for these links are set at zeros, and while pricing the artificial variables, the reduced costs for  $y^+$  and  $y^-$  corresponding to centroid connector link  $a$  are taken as  $(-\bar{\pi}_a)$  and  $(\bar{\pi}_a)$  respectively, instead of  $(M - \bar{\pi}_a)$  and  $(M + \bar{\pi}_a)$ . The foregoing modifications ensure the desired effect.

### 7.5.2 Prevention of Degeneracy and Cycling

Most real-life problems are known to be degenerate i.e., at least one of the basic variables in the solution has zero value. This gives rise to conceptual and computational difficulties in the application of linear programming concepts. On a degenerate pivot, the current basis is just switched to another basis representing the same extreme point solution. It is possible that another degenerate pivot may be performed, still remaining at the same extreme point, and so on, in a manner that the procedure gets trapped in a cycle without being able to reach optimality. This difficulty can be overcome by implementing cycling-prevention rules like the lexicographic rule for selecting an exiting variable or by using Bland's rule for selecting entering and leaving variables. However, these rules are known to be either computationally expensive to implement, as in the case of lexicographic method, or computationally inefficient, as in the case of Bland's rule. The

lexicographic rule can be interpreted as a perturbation technique, where the original right hand sides are slightly perturbed to prevent exact zero values. Recent advances in commercial codes on linear programming rely on ad-hoc perturbations of the bounds on the degenerate basic variables, thus eliminating degeneracy at the current solution (Bazaraa et al., 1990). On similar lines, a "practical" anti-cycling rule is exploited here by actually perturbing the right hand side values of constraints corresponding to the centroid connectors. Since these links are permitted to carry unrestricted flow, small but varying values are given to the corresponding right hand sides in the observed volume vector. With this incorporation, the model runs performed so far did not show any cycling, and converged satisfactorily.

### 7.5.3 Circumvention of Numerical Difficulties

A few numerical difficulties were encountered during the model application and were resolved appropriately. Principal among them was the limitation set by the integer variable for the arc costs in the NETFLO routine. Since the NETFLO routine is programmed for all-integer variables, the limiting range of values for a 4-bit integer is from -2,147,483,647 to 2,147,483,647. When a shortest path subproblem is solved using the NETFLO routine, the arc cost vectors transferred from the MAIN program may exceed the integer limit range, depending on the value of  $M$ , which again depends on the number of arcs in the network. If the element(s) of the transferred arc cost vector exceed(s) the limit, the program freezes during execution. This problem is resolved by ordering the real-valued link-cost vector first, and then assigning integer values within the permissible range, corresponding to the order of magnitude of link costs. The alternate approach of scaling the arc costs to bring them within the

range was also tried, but not adopted, since it introduces errors when the costs are in close ranges.

## ***7.6 Model Application***

The linear programming model could not be directly implemented on a personal computer for the above real network (Figure 14), because of the necessity for bigger array sizes, and due to compiler limitations. The mainframe computer (IBM 3090) was then resorted to for model execution. Minor program modifications were made to make it compatible with the mainframe.

When the development of the linear programming approach reached a stage when the model LP (without incorporating prior trip table information) was operational, it was believed to be a good opportunity to test the model on the real network, and a test was conducted at that time. On analyzing the results, the need for including prior trip information to guide the solution became evident. The model LP(TT) was then employed. The preliminary results are presented below, as further work on the testing of the model is under progress. The unavailability of a peak hour trip table for the area prevented a meaningful comparison on a common basis. However, the LP and LP(TT) model results were compared to an existing, but old, 24-hour total vehicle trip table developed by the Metropolitan Washington Council of Governments (1985), and provided by VDOT's Northern Virginia District Office, and the general trend was observed. This led to the following general conclusions.

## **7.7 General Conclusions on Preliminary Results of the Models**

Some general conclusions were arrived at after analyzing the preliminary results of the LP model. An attempt was made to compare the model results with the available 24-hour total vehicle trip table, dating back to 1985, which was projected for the Washington, D.C. and surrounding areas by the Washington Council of Governments, using socio-economic variables. The 24-hour total vehicle trips were reduced to peak hour volumes by a suitable conversion factor. On comparison, differences were observed in the travel patterns between the model results and the available trip table. This comparison has, however, highlighted some aspects of the O-D trip table estimation problem, which was useful in further refinement and enhancement of the LP model.

The LP model attempts to extract O-D flows that are in user equilibrium, if it exists, with the objective of minimizing total system cost. This leads to the following trend in estimated travel patterns. Since efficient, low-cost paths conforming to user equilibrium can be more frequently found for O-D zone pairs that are closer to each other, than between those that are farther apart, more trips are allocated to nearby zones. This led to the inference that even though the LP model is an efficient method to derive a trip table that replicates observed link volumes exactly, the incorporation of a trip distribution assumption (for example, in the form of prior information) would produce more realistic travel patterns. This resulted in the enhancement of model LP to model LP(TT).

Model LP(TT) was then applied to a smaller portion of the above network (Figure 18), which could be accommodated to run on the PC. This network includes the I-395 and I-95 corridors. It has 31 zones, 31 intersection nodes and 164 directed links. The 1985 trip rate values were used in the trip table constraints for the model. Since the current PC version of LP(TT) has the limitation of 265 constraints (both link and trip table constraints included), all the links were included in the link constraints, and the remaining 81 constraints were allotted for prior trip information. With 31 zones in the network, a total of  $31(31-1)=930$  inter-zonal flows are feasible. Only a fraction of these interchanges could hence be used as trip table constraints. Since there is a one-to-one correspondence between the internal zones of the network and the corresponding districts of the available prior trip table, 80 key O-D pairs for internal zones (representing internal-internal movements) were used as constraints. A source of difficulty, however, is how to compute the external-external, internal-external, and external-internal zonal trip rates. This is caused due to the consideration of a sub-area of a region, and the production or attraction at the cordon of the sub-area is equated to the link flows entering or leaving the cordon. The difficulty is then due to the inability to identify external zones that contribute to the entering flow on the link at the cordon. The last remaining constraint was allotted to an external-external interchange along the I-395(N) corridor, by aggregating the inter-zonal flows assumed to contribute to the flows entering and leaving the cordon at the location of external zones under consideration.

The LP(TT) model went through 16952 iterations, of which 11492 iterations were associated with Step 1, 5231 iterations with Step 2, and 229 iterations with Step 3, before declaring optimality. Obviously, the solution is not an equilibrium solution, since Step 3 has been resorted to. This is to be expected in real networks, where good data sets are difficult to obtain, and strict equilibrium conditions may not obtain

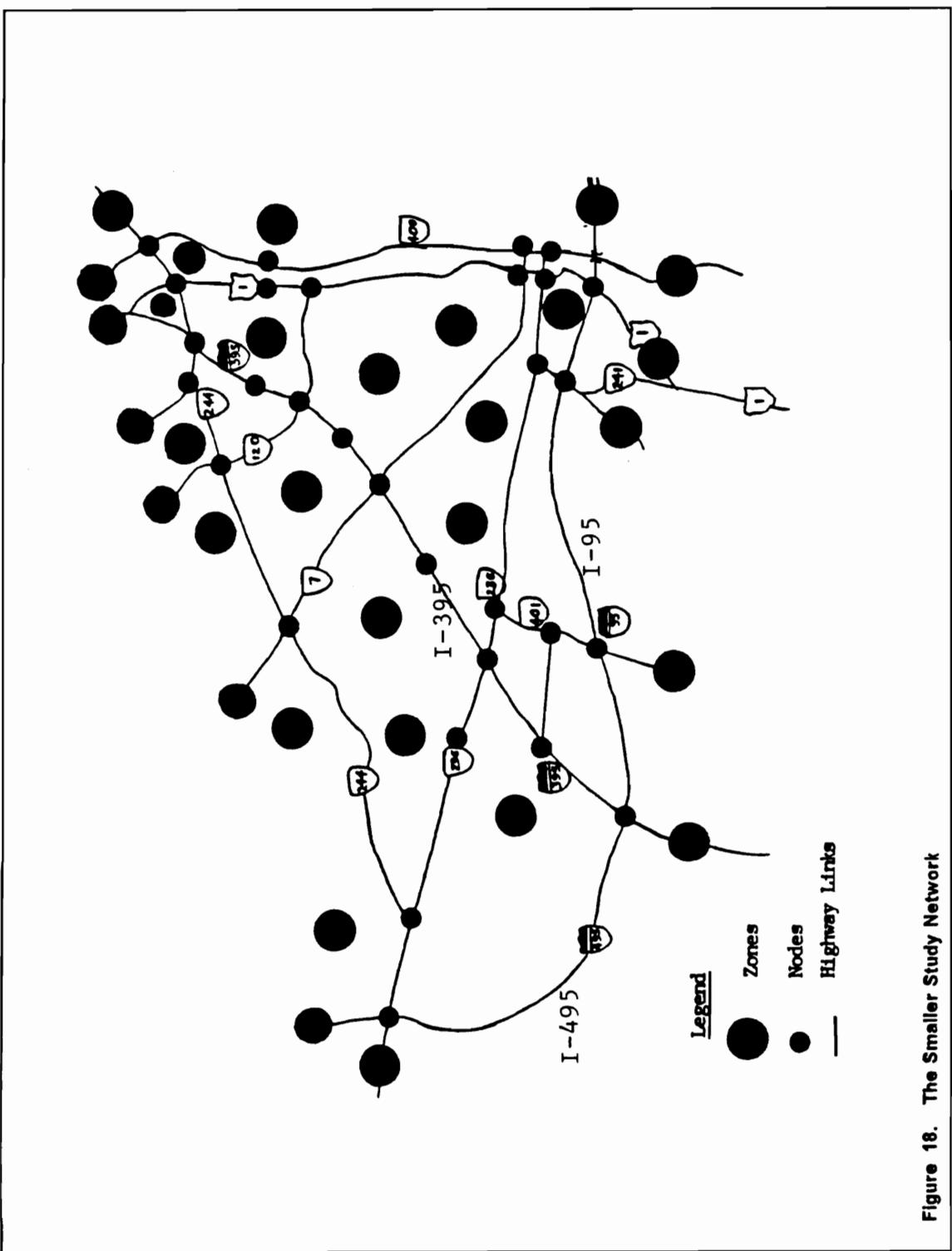


Figure 18. The Smaller Study Network

at all. Measures of closeness of observed link volumes to modelled volumes produced a RMSE of 136.57 and a MAE of 34.95 (mean link volume for the network = 2382.03). Comparing the closeness of the resulting trip table with the target table with respect to the given cell values (mean = 168.06), a RMSE of 370.48 and a MAE of 136.37 were obtained. The general trend observed from the results indicate that the additional trip table constraints of model LP(TT) aid in producing a more realistic O-D flow pattern, compared to those of model LP without these constraints. These additional constraints also enable the solution to extract a greater number of nonzero trip table cell values, by increasing the size of the basis. These initial tests were useful in exposing the issues pertinent to application of the model on real networks, and in appropriate implementation strategies. The computational effort, however, was excessively high for real-time applications. One of the possibilities to cut short the execution time is to implement an advanced-start basis procedure. Other means of speed-up may include examining the option of premature termination of the algorithm. Further tests with enhanced data sets are expected to shed more light on such real-world applications of the model.

## **8.0 CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH**

A linear programming model for estimating O-D trip tables from observed link traffic volume data has been proposed in this research work. Its development and solution algorithm have also been detailed. The procedure utilizes an efficient column generation technique to derive a solution. A potential application of the model for traffic diversion planning during congestion-causing events has also been explored. Conclusions on the model and recommendations for enhancing it are presented below.

### **8.1 *Conclusions***

The developed model procedure utilizes shortest path network flow programming subproblems in order to determine a path decomposition of flow which will reproduce

the observed flows as closely as possible, and which is driven by user-equilibrium principles. Indeed, if there exists a user-equilibrium solution which reproduces the observed flows, then this approach will finitely produce such a solution exactly, solving only shortest path subproblems with nonnegative costs coefficients. The method is a finitely convergent, linear programming approach, in contrast with the competing iterative, infinitely convergent, nonlinear programming approaches. Moreover, it is very efficient, provided that the suggested heuristic procedure is adopted to approximately solve the shortest simple path subproblems, whenever the underlying network has negative cost circuits. This method readily extends to accommodate the case in which it is required to produce a solution that has a tendency toward matching a specified prior target trip table, perhaps from amongst alternative optimal solutions to the previous model.

Preliminary computations and comparisons with other models performing a similar function on test problems as presented in Chapter 6, exhibit that the proposed method appears to be significantly superior with respect to both the quality of the solution, as well as the computational effort required. The application of the model on a real network of Northern Virginia presented an opportunity to address certain issues relevant to real networks. The results of this application underscore the importance of specifying a prior trip table for guiding the solution. The model can potentially be linked to a traffic diversion planning process, where the intent is to benefit the motorists by providing them with best alternate routes, in the context of a perturbed user equilibrium.

## **8.2 Recommendations for Further Research**

Further tests on larger, real application networks for which a reasonably accurate trip table is available, needs to be conducted in order to fully test the proposed model and algorithm. This trip table will enable a reliable comparison of the results. Also, the sensitivity of the model solution to variations in the penalty parameters needs to be tested. Additionally, tests can be conducted on the virtue of further enhancing the algorithm in case it terminates at Step 3, where the algorithm cannot find a nonbasic variable  $x_{ij}^k, k \in \bar{K}_{ij}, (i,j) \in OD$  eligible to be entered into basis. This may be as a result of encountering a shortest simple path problem on a network having negative cost circuits, and then having solved it only approximately by using the procedure of Remark 8. In this case, certain efficient paths, such as those generated by Dial's (1971) method, can be examined in coordination with the cost structure of the shortest path subproblems solved at the last visit to Step 3, in order to possibly generate variables that might be enterable in the current basis. The ambitious attempt to run the model for real-time applications for large networks may, however, require improvement in model execution time, through further algorithmic enhancements, or the design of heuristic approaches extracted from the proposed exact algorithm. This last recommendation appears to be the most useful avenue for further research, with the challenge being to attain a reliable, robust, real-time implementation strategy.

## REFERENCES

- 2020 Transportation Program, (1988), "Beyond Gridlock: The Future of Mobility as the Public Sees It," Advisory Committee on Highway Policy, Washington, D. C.
- Barbour, R. and J. D. Fricker, (1989), "Balancing Link Counts at Nodes Using a Variety of Criteria: An Application in Local Area Traffic Assignment," Transportation Research Record 1220 :33-39, Washington, D. C.
- Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali, (1990), Linear Programming and Network Flows, John Wiley & Sons, Inc., New York, New York.
- Beagan, D. F., (1990), "Maximum Entropy Matrix Estimation (ME2)," Memorandum, Central Transportation Planning Staff, Boston, Massachusetts.
- Bell, M. G. H., (1983), "The Estimation of an Origin-Destination Matrix from Traffic Counts," Transportation Science, 17(2):198-217.
- Bendtsen, P. H., (1974), "Simplification of Traffic Models," Traffic Engineering and Control, 15:817-819.
- Brenninger-Gothe, M., K. O. Jornsten, and J. T. Lundgren, (1989), "Estimation of Origin-Destination Matrices from Traffic Counts Using Multiobjective Programming Formulations," Transportation research, 23(B)(4):257-269.
- Bromage, E. J. (1988), "The Highway Emulator," Release 3.0, Central Transportation Planning Staff, Boston, Massachusetts.
- Bureau of Public Roads, (1964), "Traffic Assignment Manual," U. S. Department of Commerce, Urban Planning Division, Washington, D. C.
- Cary, M., C. Hendrickson, and K. Siddharthan, (1981), "A Method for Direct Estimation of Origin-Destination Trip Matrices," Transportation Science, 15(1):32-49.

- Cascetta, E., (1984), "Estimation of Trip Matrices from Traffic Counts and Survey Data: A Generalized Least Squares Estimator," Transportation Research-B, 18B(4/5):289-299.
- Dafermos, S. C., and T. Sparrow, (1969), "The Traffic Assignment Problem for a General Network," Journal of Research of the National Bureau of Standards, Mathematical Sciences, Section B, 73B:91-118.
- Dial, R., (1971), "A Probabilistic Multipath Traffic Assignment Model Which Obviates Path Enumeration," Transportation Research, 5: 83-111.
- Dowling, R. G. and A. D. May, (1984), "Comparison of Small Area O-D Estimation Techniques," Unpublished.
- Erlander, S., S. Nguyen, and N. F. Stewart, (1979), "On the Calibration of the Combined Distribution-Assignment Model," Transportation Research, 13(B):259-267.
- FHWA, (1985), "Highway Statistics: Summary to 1985," Washington, D. C.
- FHWA, (1988), "Highway Statistics: 1988," Washington, D. C.
- FHWA, (1989), "The Status of the Nation's Highways and Bridges: Conditions and Performances", Biennial Report to Congress, Washington,D.C.
- Fisk, C. S., (1988), "On Combining Maximum Entropy Trip Estimation with User Optimal Assignment," Transportation Research, 22(B)(1):69-79.
- Fisk, C. S., (1989), "Trip Matrix Estimation From Link Traffic Counts: The Congested Network Case," Transportation Research, 23(B)(5):331-336.
- Fisk, C. S. and D. E. Boyce, (1983), "A Note on Trip Matrix Estimation From Link Traffic Count Data," Transportation Research, 17(B)(3):245-250.
- Gaudry, M. and L. Lamarre, (1979), "Estimating Origin-Destination Matrices from Traffic Counts: A Simple Linear Intercity Model for Quebec," The Logistics and Transportation Review, 15(5):631-642.
- Gur, Y., M. Turnquist, M. Schneider, L. LeBlanc, and D. Kurth, (1980), "Estimation of an Origin-Destination Trip Table Based on Observed Link Volumes and Turning Movements", Volume 1: Technical Report, FHWA, Rept. FHWA/RD-80/034, Washington, D. C.
- Hall, M. D., D. Van Vliet, and L. G. Willumsen, (1980), "SATURN - A Simulation-Assignment Model for the Evaluation of Traffic Management Schemes," Traffic Engineering & Control, Vol. 21: 168-176.
- Hamerslag, R. and B. H. Immers, (1988), "Estimation of Trip Matrices: Shortcomings and Possibilities for Improvement," Transportation Research Record 1203, Washington, D. C.

- Han, A. F., R. G. Dowling, E. C. Sullivan, and A. D. May, (1981), "Deriving Origin-Destination Information from Routinely Collected Traffic Counts", Volume II: Trip Table Synthesis for Multipath Networks, RR81-9, Institute for Transportation Studies, University of California, Berkeley.
- Han, A. F. and E. C. Sullivan, (1983), "Trip Table Synthesis for CBD Networks: Evaluation of the LINKOD Model," Transportation Research Record 944 :106-112.
- Hanks, J. W. and T. J. Lomax, (1991), "Roadway Congestion in Major Urban Areas", Presented at the 70th Annual Meeting, Transportation Research Board, Washington, D.C.
- Hobeika, A. G., H. D. Sherali, W. A. O'Neill, R. Sivanandan, and B. Akundi, (1989), "Optimal Diversion Strategies for a Modified Urban Network," Phase 1 Project Report, US DOT/VDOT University Transportation Centers Program, Center for Transportation Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Submitted to Mid-Atlantic Universities Transportation Center (MAUTC), The Pennsylvania State University, Pennsylvania.
- Hobeika, A. G., R. Sivanandan, Y. Zhang, and K. Ozbay, (1990), "Optimal Diversion Strategies for a Congested Urban Network," Phase 2 Project Report, US DOT/VDOT University Transportation Centers Program, Center for Transportation Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Submitted to Mid-Atlantic Universities Transportation Center (MAUTC), The Pennsylvania State University, Pennsylvania.
- Hobeika, A. G., R. Sivanandan, K. Ozbay, R. Subramaniam, and Y. Zhang, (1991), "Real-Time Diversion Strategies for Congested Urban Networks," Phase 3 Project Report, US DOT/VDOT University Transportation Centers Program, Center for Transportation Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Submitted to Mid-Atlantic Universities Transportation Center (MAUTC), The Pennsylvania State University, Pennsylvania.
- Hogberg, P., (1976), "Estimation of Parameters in Models for Traffic Prediction: A Non-Linear Regression Approach," Transportation Research , 10:263-265.
- Holm, J., T. Jensen, S. K. Neilson, A. Christensen, B. Johnson, and G. Ronby, (1976), "Calibrating Traffic Models on Traffic Census Results Only," Traffic Engineering and Control , April:137-140.
- Kennington, J. L., and R. V. Helgason, (1980) Algorithms for Network Programming , John Wiley & Sons, New York.
- Kurth, D., M. Schneider, and Y. Gur, (1979), "Small-Area Trip Distribution Model," Transportation Research Record 728 :35-40, Washington, D. C.
- Lasdon, L. S., (1970), Optimization Theory for Large Systems , MacMillan.
- Lindley, J. A., (1989), "Urban Freeway Congestion Problems and Solutions: An Update," ITE Journal , December:21-23.

- Low, D. E., (1972), "A New Approach to Transportation Systems Modelling," Traffic Quarterly , 26:391-404.
- Maher, M. J., (1983), "Inferences on Trip Matrices from Observations on Link Volumes: A Bayesian Statistical Approach," Transportation Research 17B(6):435-447.
- McNeil, S. and C. Hendrickson, (1985), "A Regression Formulation on Matrix Estimation Problem," Transportation Science, 19(3):278-292.
- Metropolitan Washington Council of Governments, (1987), "Impact Assessment Districts," Washington, D. C.
- Meyer, M. D., T. F. Humphrey, C. M. Walton, K. Hooper, R. G. Stanley, C. K. Orski, and P. A. Peyser, Jr., (1989), "A Toolbox for Alleviating Traffic Congestion," Institute of Transportation Engineers, Washington, D. C.
- Microsoft, (1987), "FORTRAN 4.1 Optimizing Compiler," Microsoft Corporation.
- Murchland, J., (1977), "The Multi-proportional Problem," University College, London, Research Note JDM 263 (Unpublished).
- Neumann, E. S., J. Halkias and M. Elrazaz, (1983), "Estimating Trip Rates from Traffic Counts," Journal of Transportation Engineering , 109:565-578.
- Nguyen, S., (1977), "Estimating an OD Matrix from Network Data: A Network Equilibrium Approach," University of Montreal Publication No. 60.
- Nguyen, S., (1984), "Estimating Origin-Distribution Matrices from Observed Flows" in Transportation Planning Models, Editor: M. Florian, Elsevier Science Publishers B.V., The Netherlands, pp.363-380.
- O'Neill, W. A., (1985), Trip Table Estimation Program in Advanced BASIC, for Network Equilibrium Approach, Personal Communication.
- O'Neill, W. A., (1986), FORTRAN Program for Equilibrium Trip Table Assignment, Personal Communication.
- O'Neill, W. A., (1987), "Origin-Destination Trip Table Estimation Using Traffic Counts", Ph.D dissertation submitted to University of New York at Buffalo, New York.
- O'Neill, W. A. and R. Sivanandan, (1989), "Synthetic O-D Microcomputer Trip Table Packages," Proceedings, Third International Conference on Microcomputers in Transportation, San Francisco, California.
- Robillard, P., (1975), "Estimating the O-D Matrix from Observed Link Volumes," Transportation Research , 9: 123-128.
- Smith, D. P. and B. G. Hutchinson, (1981), "Goodness of Fit Statistics for Trip Distribution Models," Transportation Research - A, 15A:295-303.

- Smock, R., (1962), "An Iterative Assignment Approach to Capacity Restraint on Arterial Networks," Highway Research Board Bulletin 347:60-66.
- Stowers, J. R., (1988), "Organizing and Funding for 2020," in "A Look Ahead: Year 2020", Transportation Research Board, Washington, D. C.
- Turnquist, M., and Y. Gur, (1979), "Estimation of Trip Tables from Observed Link Volumes," Transportation Research Record 730: 1-6, Washington, D. C.
- U. S. Congress, (1988), Conference Report to Accompany H. R. 4794, House of Representatives, 101st Congress, 2nd Session, Report 100-957.
- US DOT, (1989), Discussion Paper on Intelligent Vehicle-Highway Systems, Washington, D. C.
- US DOT, (1990), "Moving America: New Directions, New Opportunities. A Statement of National Transportation Policy: Strategies for Action," Washington, D. C.
- Van Vliet, D. and L. G. Willumsen, (1981), "Validation of the ME2 Model for Estimating Trip Matrices from Traffic Counts", Proceedings of the Eighth International Symposium on Transportation and Traffic Theory, Editors: V. F. Hurdle, E. Hauer and G. M Stuart, University of Toronto Press, pp. 640-655.
- Van Zuylen, H. J., (1978), "A method to Estimate a Trip Matrix from Traffic Volume Counts," in "Transportation Models," PTRC Proceedings of the Summer Annual Meeting, University of Warwick, England, pp. 305-311.
- Van Zuylen, H. J., (1981), "Some Improvements in the Estimation of an OD Matrix from Traffic Counts," Proceedings of the Eighth International Symposium on Transportation and Traffic Theory, Editors: V. F. Hurdle, E. Hauer, and G. N. Stuart, University of Toronto Press, pp. 656-671.
- Van Zuylen, H. J., and D. M. Branston., (1982), "Consistent Link Flow Estimation from Counts", Transportation Research-B , 16B(6): 473-476.
- Van Zuylen, H. J., and L. G. Willumsen, (1980), "The Most Likely Trip Matrix Estimated from Traffic Counts", Transportation Research-B , 14B: 291-293.
- VDOT, "I-395/I-66's Traffic Management System", Traffic Control Center, Arlington, Virginia, and Personal Communication with Staff (1991).
- Wardrop, J. G., (1952), "Some Theoretical Aspects of Road Traffic Research," Proceedings, Institute of Civil Engineers , Part 2, 1: 325-378.
- Wills, M., (1977), "Non Linear Estimation of Origin-Destination Flows and Gravity Model Parameters from Traffic Counts on Links," Department of Geography, University of British Columbia, Mimeographed Notes.
- Willumsen, L. G., (1978), "Estimation of an O-D Matrix from Traffic Counts - A Review," Working Paper 99, Institute of Transport Studies, University of Leeds, England.

Willumsen, L. G., (1982), "Estimation of Trip Matrices from Volume Counts : Validation of a Model Under Congested Conditions," in "Transportation Analysis and Models," PTRC Proceedings of the 10th Summer Annual Meeting, University of Warwick, England, pp. 311-323.

Wilson, A. G., (1970), Entropy in Urban and Regional Modelling, Pion Limited, London.

## **APPENDICES**

## **Appendix A. Backtracing Algorithm**

The purpose of this algorithm is to trace the minimum paths between each O-D pair from the minimal cost arc flows obtained from the NETFLO routine. Knowing the destination demands, origin supplies and arc flows, the backtracing algorithm traces the associated paths as follows.

Let a network flow programming problem defined on a digraph with a set of supply nodes  $L_s$  and a set of demand nodes  $L_d$  be considered, and let it be assumed that a set of consistent link flows are given. Hence, there is a net creation of flow at nodes in  $L_s$ , a net absorption of flow at nodes in  $L_d$ , and the flow is conserved at all other nodes. It is required to determine a set of path flows from nodes in  $L_s$  to nodes in  $L_d$ , consistent with the net supplies and demands at these nodes, along with any superimposed flows in circuits, which will produce the given link flows. The following algorithm accomplishes this task.

Step 1:

If  $L_d = \phi$ , then stop. The required decomposition of the link flows into path flows is available. Otherwise, pick a node  $J \in L_d$ . Find any arc coming into this node which has a positive flow on it, and let the other node associated with this arc be node  $I_k$ , and put  $k = 1$ . (If a node in  $L_s$  qualifies as such as  $I_k$ , pick such a node).

Step 2:

If  $I_k \in L_s$ , go to step 3. Otherwise, go to step 4.

Step 3:

A path from  $I_k$  to  $J$  with  $k$  arcs on it has been found. Find the quantity  $\Delta = \min\{\text{supply(current) at } I_k, \text{demand(current) at } J, \text{flow on arcs in the detected path}\}$ . Note that  $\Delta > 0$ . Designate  $\Delta$  as the (additional) flow between O-D pair  $(I_k, J)$ . Subtract  $\Delta$  from each of the quantities in the above minimand. If  $\Delta = \text{supply at } I_k$ , remove  $I_k$  from  $L_s$ . If  $\Delta = \text{demand at } J$ , remove  $J$  from  $L_d$ . Return to step 1.

Step 4:

Find an arc with positive flow coming into  $I_k$ , and let  $I_{k+1}$  be the other node associated with this arc (If a node  $L_s$  qualifies as such as  $I_{k+1}$ , pick such a node).

If the current path constructed thus far already contains  $I_{k+1}$ , go to step 5.  
Otherwise, increment  $k$  by 1, and return to step 2.

Step 5:

A directed cycle (circuit) with a positive flow has been traced. Find  $\delta = \text{smallest flow on any arc in this circuit, and note that } \delta > 0$ . Subtract  $\delta$  from the flow on each arc in this circuit and return to step 1.

Hence, in the above routine, at each main pass, either an O-D path with a positive flow is traced and accounted for, or else an inefficient circuit with a positive flow is detected and eliminated.

## Appendix B. INITSOLN Algorithm

This algorithm determines the initial basic feasible solution (IBFS), before the actual column generation technique procedure is commenced. Via the backtracing algorithm, a feasible solution is obtained as,  $(x, x_a) = (\bar{x}, \bar{x}_a)$  where  $\bar{x}$  is the value of the original O-D flow vector,  $x$ , and  $\bar{x}_a$  is the value of an artificial flow vector,  $x_a$ . If  $\bar{x}_a \neq 0$ , the problem is infeasible (i.e., the link flows cannot be precisely duplicated by O-D flows), except in the case of the presence of a cycle, whence it is possible that  $\bar{x}_a > 0$  from the backtracing procedure, though the solution may be feasible. Otherwise, the solution  $(x, x_a) = (\bar{x}, \bar{x}_a)$  is to be revised to a basic feasible solution to the linear program. The various steps in this task are listed below.

### Initialization:

Construct the basic variable list  $KBAS = \{y_a^+, a \in A, Y_{ij}^+, (i,j) \in \overline{OD}\}$ , put  $B^{-1} = [/], c_B B^{-1} = [M, \dots, M, M_\sigma, \dots, M_\sigma]$ .

Let  $RHS = \{RHS(S_a), RHS(S_t)\}$ , where  $RHS(S_a)$  and  $RHS(S_t)$  are the right hand side vectors corresponding respectively, to the set of link constraints  $S_a$ ,  $a \in A$ , and the set of trip table constraints  $S_t, t \in \overline{OD}$ . More specifically,

$RHS(a) = (\bar{f}_a) - (\text{total flow on link } a \text{ due to the solution } \bar{x})$ ,

$RHS(t) = Q_{ij} - (\text{total O-D flow for interchange } (i,j) \text{ due to the solution } \bar{x})$ .

If for any  $k \in t$ ,  $RHS(k) < 0$ , make  $RHS(k) = |RHS(k)|$ ,  $KBAS(k) = Y_{ij}$ ,  $B^{-1}(k,k) = -1$ , and  $c_B B^{-1}(k) = -c_B B^{-1}(k)$ . Define  $PNB = \{x_j \text{ nb} : \bar{x}_j > 0\}$  to be the list of currently positive nonbasic variables, and assume that  $PNB \neq \emptyset$  (Else, go to step 6).

### Step 1:

Pick  $x_j \in PNB$ . Determine  $y_j = B^{-1} Y_j$  as its updated column. Find maximum  $\{|y_{jk}| : KBAS(k) \text{ is an artificial variable at value 0}\} = |y_{jk'}|$ , say.

If  $|y_{jk'}| = 0$  or if  $k' \notin PNB$ , go to step 2.

Otherwise,

(a) Pivot on  $y_{jk'}$  to update  $B^{-1}$  and  $c_B B^{-1}$  as usual.

(b) Make  $KBAS(k') = x_j$  and  $RHS(k') = \bar{x}_j$ .

(Note: The artificial variable which was basic in position  $k'$  is now eliminated from the problem).

(c) Remove  $x_j$  from  $PNB$ .

(d) Go to step 5.

Step 2:

Find  $(z_j - c_j) = c_B B^{-1} (Y_j) - c_j$ . If  $(z_j - c_j) \geq 0$ , go to step 3.

Otherwise, go to step 4.

Step 3:

If  $(y_j) \leq 0$ , go to step 4. (Note: In this case we must have  $(z_j - c_j) = 0$ , or else the problem is unbounded).

Otherwise, find

$$\Delta = \underset{k : y_{jk} > 0}{\text{minimum}} \left\{ \frac{RHS(k)}{y_{jk}} \right\} = \left\{ \frac{RHS(k')}{y_{jk'}} \right\}, \text{ say.}$$

Hence, the variable  $KBAS(k') = x_{k'}$ , say, leaves the basis at value zero. Do the following:

- (a) Pivot as usual on  $y_{jk'}$  to update  $B^{-1}$  and  $c_B B^{-1}$ .
- (b) Put  $KBAS(k') = x_j$  and put  $RHS(k') = \bar{x}_j + \Delta$ .
- (c) For each  $k \neq k'$ , put  $RHS(k)$  equal to  $RHS(k) - \Delta y_{jk}$ .
- (d) Remove  $x_j$  from PNB.
- (e) Go to step 5.

Step 4:

We now decrease  $x_j$  from its current value of  $\bar{x}_j$ . Determine

$$\Delta_1 = \underset{k : y_{jk} < 0}{\text{minimum}} \left\{ \frac{RHS(k)}{-y_{jk}} \right\} = \left\{ \frac{RHS(k')}{-y_{jk'}} \right\}, \text{ say,}$$

where,  $\Delta_1 = \infty$  if  $(y_j) \geq 0$ .

Put  $\Delta = \min\{\Delta_1, \bar{x}_j\}$ .

If  $\Delta = \bar{x}_j$ , do the following:

- (a) Put  $RHS(k)$  equal to  $RHS(k) + \Delta y_{jk} \forall k$
- (b) Remove  $x_j$  from  $PNB$  (its value is now zero).
- (c) Go to step 5.

Otherwise, if  $\Delta = \Delta_1 < \bar{x}_j$ , do the following:

- (a) Pivot as usual on  $y_{jk}$  to update  $B^{-1}$  and  $c_B B^{-1}$ .
- (b) Put  $KBAS(k') = x_j$  and put  $RHS(k') = \bar{x}_j - \Delta$ .
- (c) For each  $k \neq k'$ , put  $RHS(k)$  equal to  $RHS(k) + \Delta y_{jk}$ .
- (d) Remove  $x_j$  from  $PNB$ .
- (e) Go to step 5.

#### Step 5:

If  $PNB = \phi$ , go to step 6.

Otherwise, return to step 1.

#### Step 6:

We now have  $KBAS$  equal to the set of basic variables,  $B^{-1}$ ,  $c_B B^{-1}$ ,  $RHS$  equal to values of basic variables, and all nonbasic variables equal to 0. Note that since  $PNB$  reduces by one through each pass of steps 1-5, we reach this step finitely. Given this BFS, we can perform the column generation revised simplex algorithm with the following modification.

Suppose we generate an entering updated column  $y_j$ . We can check as in step 1 above if some  $|y_{jk'}| \neq 0$  exists, where  $KBAS(k')$  is an artificial variable at value zero. If so, then, since this artificial variable is at value zero, we can simply pivot (degenerately) on  $y_{jk'}$  to update  $B^{-1}$ ,  $c_B B^{-1}$ , (RHS), and  $KBAS$ , and thereby eliminating this artificial variable from the problem.

# **Appendix C. COMPUTER PROGRAM DOCUMENTATION AND IMPLEMENTATION STRATEGIES**

## **C.1 *Introduction***

The column generation technique developed in this research for solving the linear programming problem to estimate a trip table from link volumes and a prior trip table has been translated into a FORTRAN program for computer implementation. The Microsoft FORTRAN Optimizing Compiler, Version 4.1 (1987), is used for compilation and execution on a personal computer with the MS-DOS operating system. The program can also be run on a mainframe computer with minor modifications to render it compatible with the mainframe FORTRAN version. The structure of the program is reflective of the algorithm itself. The program is designed for an interactive input of user data and

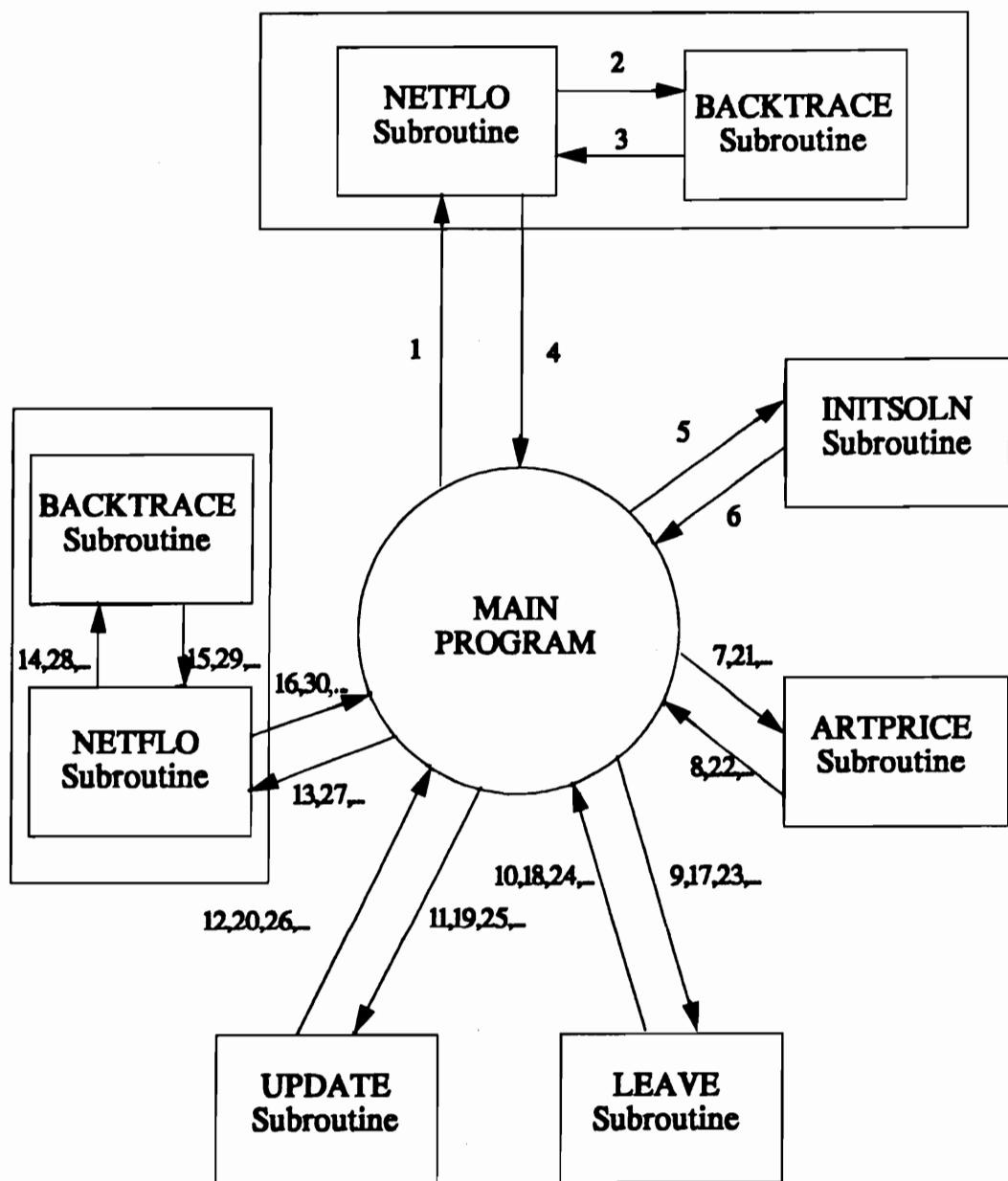
parameters. Several implementation strategies for speeding up the execution of the program are incorporated, and are discussed in this chapter. The program is divided into a main program and six subroutines to accomplish the distinct tasks of the various steps of the algorithm. These are described below.

## C.2 *Overall Program Structure*

The whole program is divided into the following:

- MAIN Program
- Subroutine INITSOLN
- Subroutine ARTPRICE
- Subroutine NETFLO
- Subroutine BACKTRACE
- Subroutine LEAVE
- Subroutine UPDATE

The MAIN program acts as the driver and calls the appropriate subroutines at various stages of the algorithmic implementation. The flow chart for the overall program structure is shown in Figure 19.



1,2,3... : Flow Sequence

Figure 19. Overall Program Structure for the Linear Programming Model LP(TT)

After reading the input data and user-defined parameters and options, the MAIN program transfers the control to subroutine NETFLO, which is a minimum cost network flow programming algorithm. This routine passes its output to BACKTRACE subroutine to determine shortest path costs for all feasible O-D pairs, based on observed link impedances. This information is used later in pricing the nonbasic variables. Initializing the linear programming solution process is the next step. The control is then passed to subroutine INITSOLN, which records the initial basic feasible solution. Subroutine ARTPRICE is then called for pricing the artificial variables. After pricing, the eligible variables are entered into the basis by determining, each time, the leaving basic variable through the use of subroutine LEAVE, and then updating the primal and dual solutions through subroutine UPDATE. When no artificial variable is found eligible for entering, the next task is to price the legitimate nonbasic variables ( $x_j^k$ 's) representing the O-D pairs, for entering into the basis. This is accomplished through the use of two subroutines, NETFLO and BACKTRACE again, which solve the appropriate subproblems. The output of NETFLO is used by BACKTRACE to construct shortest paths between the O-D pairs, and thus determine the reduced cost coefficients corresponding to the legitimate nonbasic variables. If a nonbasic variable is found eligible to enter into the basis, it is entered by finding the leaving basic variable, and updating the solutions. A check for equilibrium is conducted, and the algorithm is terminated if an equilibrium is attained. If not, the control is passed to ARTPRICE for pricing artificial variables again, and the sequence of steps is repeated. On the other hand, if neither an artificial variable nor a legitimate variable is found eligible to enter, optimality is declared, and the algorithm terminated.

### C.3 *MAIN Program*

The basic function of the MAIN program is to control the sequence of steps of the column generation algorithm. In addition to initializing the revised simplex scheme for the column generation technique, it also manages input/output files. Certain checks are also made in the MAIN program for determining optimality, and conditions for appropriate transfer of control. The flow chart for the MAIN program is shown in Figure 20.

The program starts by requiring the user to input the names of the network data and observed volume data files, that are to be read. This enables the user to build these data files beforehand. The network data file is comprised of the number of links, nodes and zones of the network; the 'from' and 'to' nodes, free-flow travel time, and the capacity of each of the network links. The observed volume file consists of the link numbers and the observed volumes on the network links. After reading these two files, the user is asked to define the number of prior trip table cell values that he/she is interested in specifying for guiding the final solution. This gives an option to the user to specify only key cell values that are considered important, and not necessarily the whole table. The actual cell values need to be input by the user next, followed by the value of  $\sigma$ . The name of the output file to which the output data are to be written is given next. At this stage, the internal clock is activated to keep a record of the start time of program execution. The link costs, which in this case are the impedances to travel offered by the links, are computed based on observed link volumes. The Bureau of Public Roads (BPR) (1964) link performance function is used to calculate this impedance. If such link costs are previously known, data on

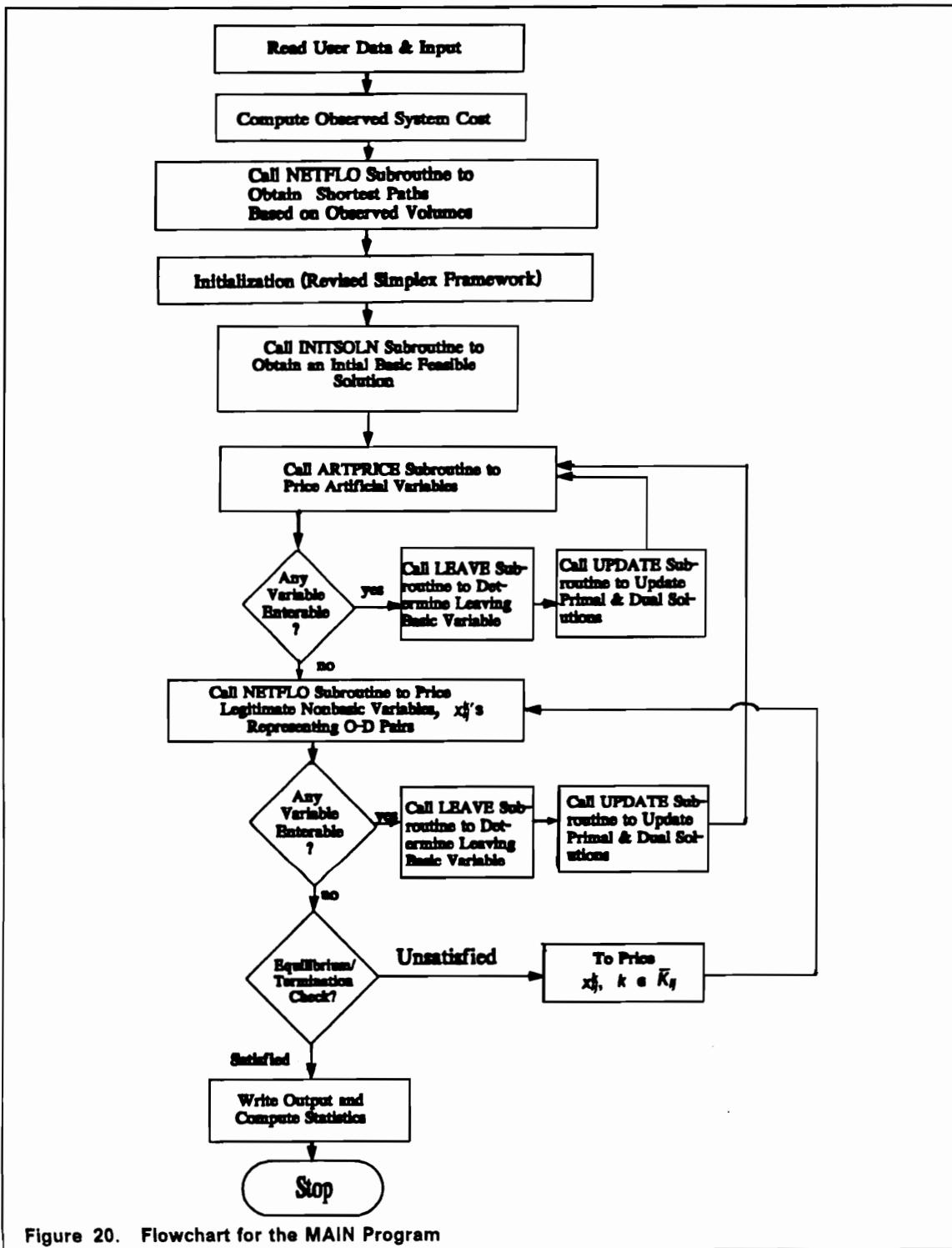


Figure 20. Flowchart for the MAIN Program

free-flow travel time and capacity of links are not necessary. The value of  $\bar{C}_{\text{total}} = \sum_a \bar{c}_a \bar{f}_a$  is determined for later user-equilibrium checks.

The first major task is the computation of shortest path costs, based on observed link volumes, for all feasible O-D pairs. This information is stored to be used later in determining whether the path corresponding to the legitimate nonbasic variable to be entered into the basis, is the shortest or not. This enables pricing the nonbasic variables as in [5.8] and [5.9]. Only the path costs need be stored, and not the paths themselves, since a comparison of path costs is sufficient to indicate whether the path in question is the shortest or not. To get the shortest paths, subroutines NETFLO and BACKTRACE are employed. Since NETFLO and BACKTRACE are again used to solve the shortest path subproblems  $SP_{ij}^I$  and  $SP_{ij}^{II}$  for subsequent pricing of nonbasic variables, the updated vector  $c_B B^{-1}$  is transferred to NETFLO at each call to solve the subproblem.

Initializing the revised simplex tableau is a major task of the MAIN program. The basic variables list vector,  $KBAS(.)$ , the simplex multipliers vector  $c_B B^{-1}(.)$ , the right hand side vector  $RHS(.)$ , and the  $B^{-1}$  matrix are to be initialized.  $KBAS(a)$  is set to  $y_a^+$   $\forall a \in A$ , representing the deviations in the link flows. This takes care of constraints [4.12b].  $RHS(a)$  is set to  $\bar{f}_a$   $\forall a \in A$ . For constraints [4.12c], representing prior trip information,  $Y_{ij}^t$  are taken as initial basic variables, and  $RHS(t)$ ,  $t \in \overline{OD}$ , are set as  $Q_{ij}$   $\forall (i, j) \in \overline{OD}$ .  $B^{-1}$  is set as  $[I]$ , the identity matrix, and  $c_B B^{-1}$  as  $[M, \dots, M, M_\sigma, \dots, M_\sigma]$ . The parameters that keep track of the number of iterations for various steps are initialized at zero. The initial objective value and  $M_2$  are also computed.

Call is now made to subroutine INITSOLN for obtaining an initial solution. As new legitimate variables are entered into the basis, the solution is also updated. The details of the INITSOLN algorithm are explained later. The artificial

variables  $y_a^+$ ,  $y_a^-$ ,  $Y_{ij}^+$ ,  $Y_{ij}^-$  are then priced by calling the subroutine ARTPRICE. The eligible variable, if any, is entered into the basis by determining the leaving basic variable through the subroutine LEAVE. The solutions are updated by calling the UPDATE subroutine. If no artificial variable is found eligible to enter, call is then made to subroutines NETFLO and BACKTRACE to solve the appropriate subproblem,  $SP_{ij}^I$  or  $SP_{ij}^{II}$ , for pricing the legitimate nonbasic variables representing  $x_j^k$ 's. If an enterable variable is found, it is entered into the basis by calling the LEAVE and the UPDATE subroutines, and the control is then passed back to the subroutine ARTPRICE for pricing the artificial variables again, and the procedure is repeated. Equilibrium/optimality checks are also performed to determine termination. After terminating the algorithm, statistics are computed, and the output is written.

## C.4 Subroutine INITSON

This routine determines an advanced start initial basic feasible solution to begin the optimization process. Though this routine is not a necessity for the optimization process, it ensures a speedier execution of the algorithm. Alternatively, the column generation procedure can be directly begun with an all-artificial basis, in which case this subroutine can be bypassed. Essentially, the INITSON algorithm consists of extracting feasible O-D path flows from the given link volumes, and entering them into basis. This is accomplished through a backtracing algorithm (Appendix A), and a subsequent purification process to

derive a basic feasible solution (Appendix B). The flow chart for the INITSLN algorithm is shown in Figure 21.

Initially, the origin and destination nodes are identified based on whether there is net supply or demand at a node. This is done by examining the total incoming and outflowing observed flows at each of the nodes. If there is net outflow, then the node is an origin node; if not, it is a destination node. These origin and destination zone numbers are stored as lists for examination. If the net flow is zero, the node is a transshipment node. Now, through the backtracing procedure detailed in Appendix A, several O-D pair flows that satisfy the observed link volumes are extracted. Their paths, path costs and the flow values are stored in memory as a list. These are the positive nonbasic variables (PNBs) eligible to enter into basis. The right hand side vector  $RHS(.)$  is then set as:

$$RHS(a) = (\bar{f}_a) - (\text{total flow on link } a \text{ accounted for by the extracted PNBs}) \quad \forall a \in A$$

$$RHS(t) = Q_{ij} - (\text{total flow for O-D pair } (i,j) \text{ accounted for by the extracted PNBs})$$

$$\forall (i,j) \in \overline{OD}$$

For trip table constraints, it is possible that the elements of the above vector may turn out to be negative, representing a negative deviation. In this case, the initialization settings are altered as follows. Suppose row  $k$  represents the above case, then the following settings are made:

$$RHS(k) = |RHS(k)|$$

$$KBAS(k) = Y_{\bar{i}\bar{j}}$$

$$B^{-1}(k,k) = -1$$

$$c_B B^{-1}(k) = -c_B B^{-1}(k)$$

The next task is to enter these nonbasic variables into the basis. Each variable in the list is picked one-by-one, and is entered into the basis, by finding the updated column, increasing or decreasing its current value to improve the basic

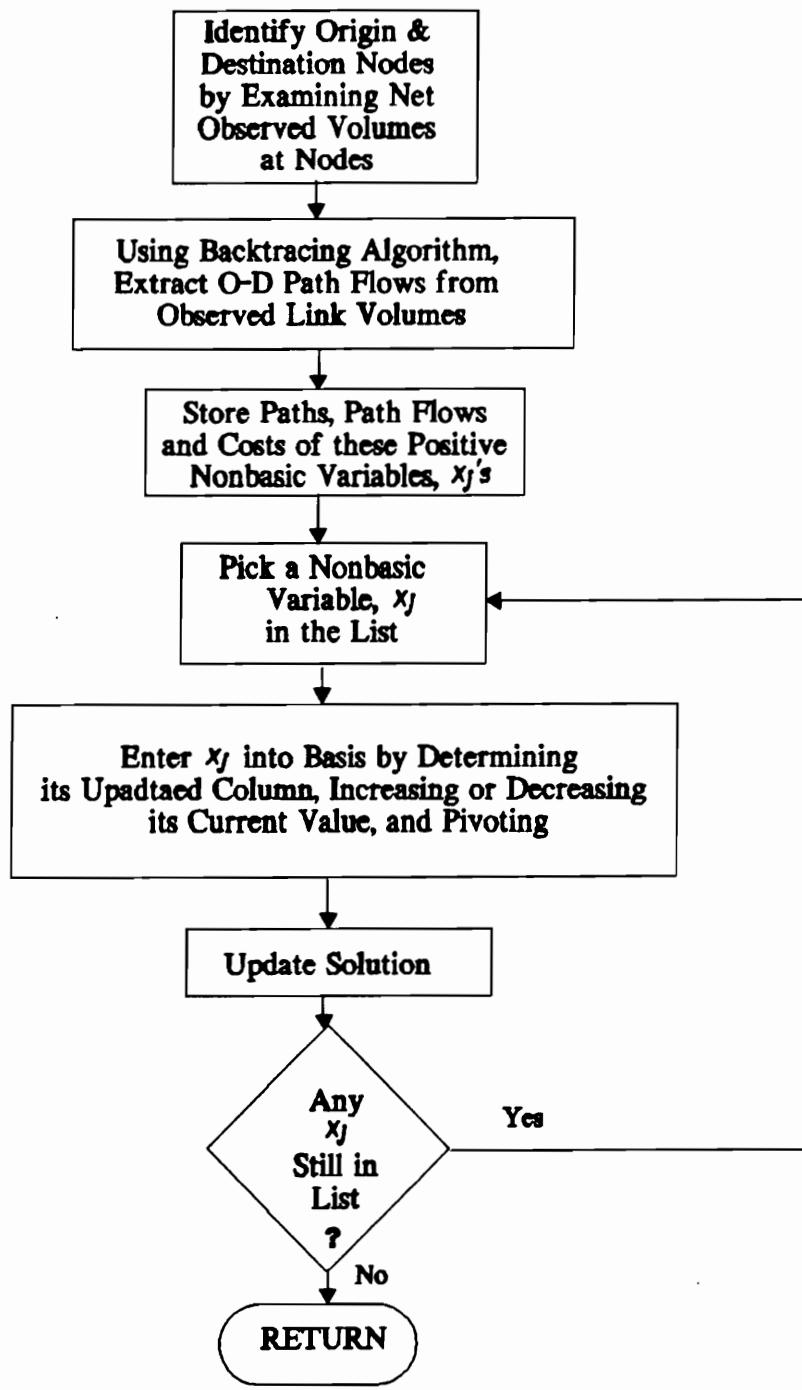


Figure 21. Flowchart for the INITSONL Subroutine

feasible solution, and pivoting using the minimum ratio test, as detailed in the INITSLN algorithm (Appendix B). The reduced cost is computed as per [5.8] or [5.9] according to whether the path represented by the nonbasic variable is the shortest or not, for the corresponding O-D pair. The  $c_B B^{-1}$  and  $RHS$  vectors,  $B^{-1}$ , and the objective value are updated in the process. A purified initial basic feasible solution (possibly with artificials), in a revised simplex framework, is now at hand. The control is then transferred to the MAIN program.

## C.5 Subroutine ARTPRICE

After obtaining an initial basic feasible solution, the iterative process of the column generation algorithm (CGA) is commenced. Pricing the artificial variables is the first step in the four-step process, as detailed in Chapter 5. This is carried out in ARTPRICE subroutine. The following are computed:

$$\text{maximum } \{\bar{\pi}_a : a \in A\},$$

$$\text{minimum } \{\bar{\pi}_a : a \in A\},$$

$$\text{maximum } \{\bar{\mu}_{ij} : (i, j) \in \overline{OD}\},$$

$$\text{minimum } \{\bar{\mu}_{ij} : (i, j) \in \overline{OD}\},$$

$$v^0+ = M - \text{maximum } \{\bar{\pi}_a : a \in A\},$$

$$v^0- = M + \text{minimum } \{\bar{\pi}_a : a \in A\},$$

$$N^0+ = M_\sigma - \text{maximum } \{\bar{\mu}_{ij} : (i, j) \in \overline{OD}\}, \text{ and}$$

$$N^0- = M_\sigma + \text{minimum } \{\bar{\mu}_{ij} : (i, j) \in \overline{OD}\}$$

$v^0 = \min\{v^{0+}, v^{0-}, N^{0+}, N^{0-}\}$  is also determined, and the control is passed back to the MAIN program. If  $v^0 < 0$ , then the artificial variable corresponding to  $v^0$  is enterable into the basis. If so, the corresponding settings are made. The MAIN program then calls the subroutines LEAVE and UPDATE to determine the leaving basic variable, pivot, and update the solutions. The flowchart for this subroutine is depicted in Figure 22.

## C.6 Subroutine NETFLO

Once all the currently eligible artificial variables have been entered into the basis, the next step is the pricing of nonbasic variables representing legitimate O-D pairs. This implies that the columns of the  $x$ -variables need to be determined, and a check made if any  $x_{ij}^k$  variable has a negative reduced cost. This is accomplished through solving the shortest path subproblems  $SP_i^k$  and  $SP_j^k$  as in [5.6] and [5.7]. Toward this end, the current dual solution vector  $\bar{\pi}$  (" $c_B B^{-1}$ ") is transferred from the MAIN program to NETFLO, and the link cost vectors  $(M_2 \bar{c} - \bar{\pi})$  and  $(M_1 \bar{c} - \bar{\pi})$  are used in the subproblems  $SP_i^k$  and  $SP_j^k$ , respectively. NETFLO is a FORTRAN program to solve minimal cost network flow problems with bounded arc flows. The code for the algorithm was borrowed from *Algorithms for Network Flow Programming* (Kennington and Helgason, 1970) and modified to suit present needs. Essentially, given some supplies and demands at various nodes of a network, this algorithm will determine the minimum cost allocation through the network to transport the supplies at supply nodes and to satisfy the demands at demand nodes, knowing link costs, and subject to arc

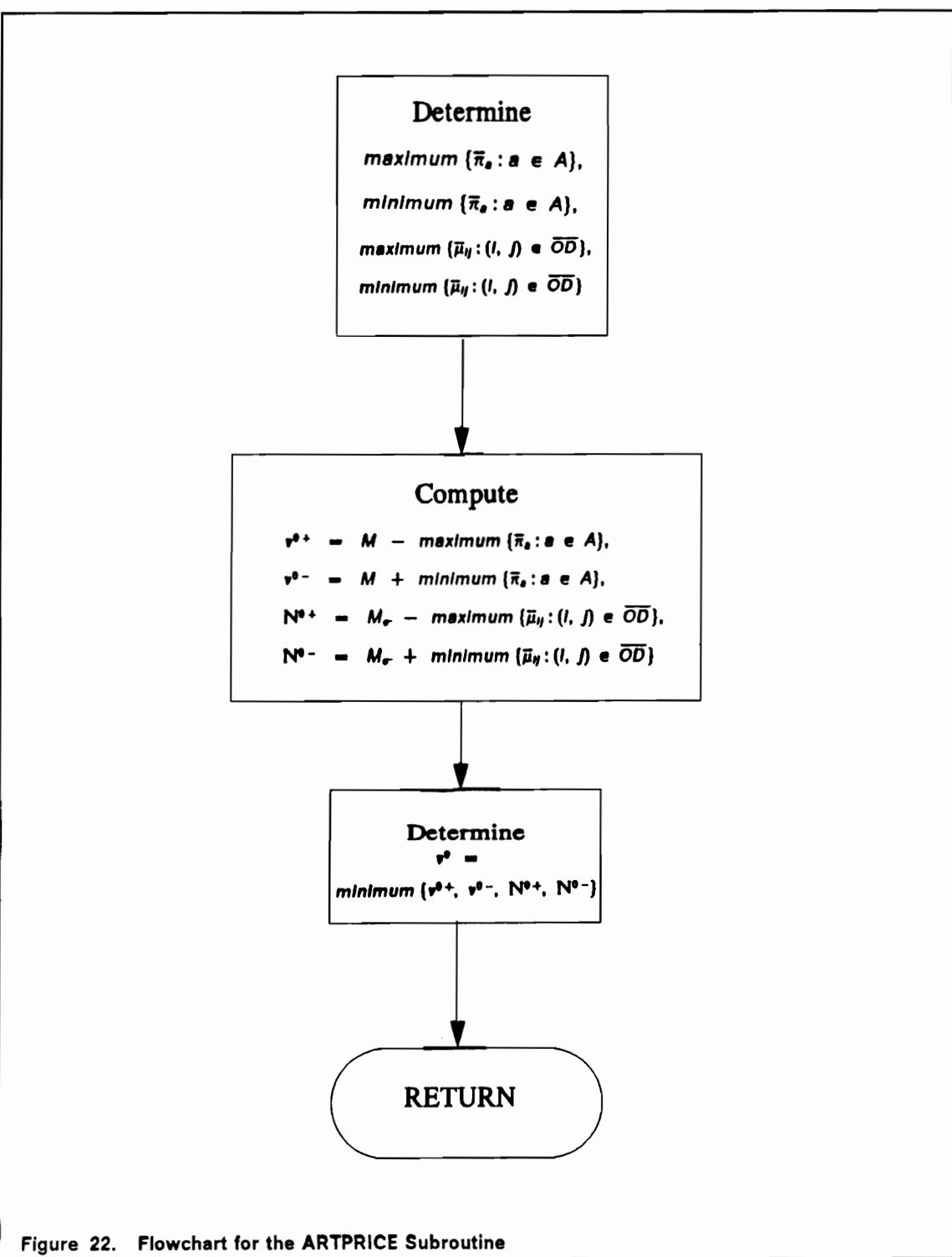


Figure 22. Flowchart for the ARTPRICE Subroutine

bounds. This algorithm is chosen for use in this research with a specific purpose. Since, some of the elements of the link cost vector for subproblem  $SP_{II}$  may be negative, negative cost circuits may be present in the network, in which case the usual shortest path algorithms would enter an endless cycle. An attempt is made to overcome this problem by resorting to NETFLO, using suitable arc bounds. Here, the subproblem is solved by allotting unit demand at each destination node and a supply at the origin node, equal to the number of destinations. The upper bound on each arc flow is made equal to the supply, and its lower bound is fixed at zero. The imposition of the upper bounds prevent unboundedness in the presence of negative circuits. The NETFLO solution then yields arc flows that satisfy the destination demands using the minimum cost paths between the supply-demand points.

The simple shortest path between each O-D pair is then extracted from the NETFLO output using the BACKTRACE algorithm, as described in Appendix A. For each origin, NETFLO is run once, and BACKTRACE is run as many times as there are number of destinations, to obtain the shortest paths between that origin and other destination nodes. It is to be noted that the solution is not guaranteed to be optimal to the shortest simple path problem in the presence of negative cost circuits, as stated earlier. However, this is a NP complete problem. The reduced costs for the  $x$ -variables are determined from the costs of these paths. Thus, in order to determine the most eligible nonbasic variable to enter into basis, the reduced cost corresponding to a  $x$ -variable that is most negative needs to be determined. This is carried out easily by comparing the current reduced cost for the O-D pair in question with the one stored earlier. Thus, at the end of the NETFLO algorithm, the nonbasic  $x$ -variable that has the least reduced cost, is at hand. The control is now returned to the MAIN program.

## C.7 Subroutine BACKTRACE

The output from the subroutine NETFLO give the set of arc flows as a solution to the minimal cost network flow problem of transporting a unit quantity of supply from the origin node under consideration to each of the demand nodes. These link flows are then used to determine the shortest paths for different O-D pairs, via the backtracing algorithm. This algorithm is programmed in subroutine BACKTRACE. The algorithm is detailed in Appendix A, and the flow chart for this routine is shown in Figure 23.

While determining these paths is straightforward in the absence of negative link cost circuits, the backtracing algorithm employs some heuristics in the presence of negative circuits. In either case, whether there is a cycle present or not, the interest is in extracting the shortest paths between different O-D pairs. The costs of these paths determine the reduced costs of the nonbasic variables  $x_{ij}^k$ 's representing these O-D pairs. The intent is thus to price the nonbasic variables for entering the most eligible nonbasic variable into the basis.

As the name implies, the backtracing algorithm traces the flows backward from the destinations to origins. The algorithm is initiated by storing the destination nodes in a list, for examination one after another. When the demand at a destination node is completely exhausted, it is eliminated from the list, and the next destination node is considered. For each of the destination nodes (or zones), the algorithm begins by finding an arc coming into the node with positive flow on it. If the incident node is an origin node, a path from the destination node to an origin node has been found. The minimum of supply at the origin, demand at the destination, and the flow on the arcs in the detected path is determined,

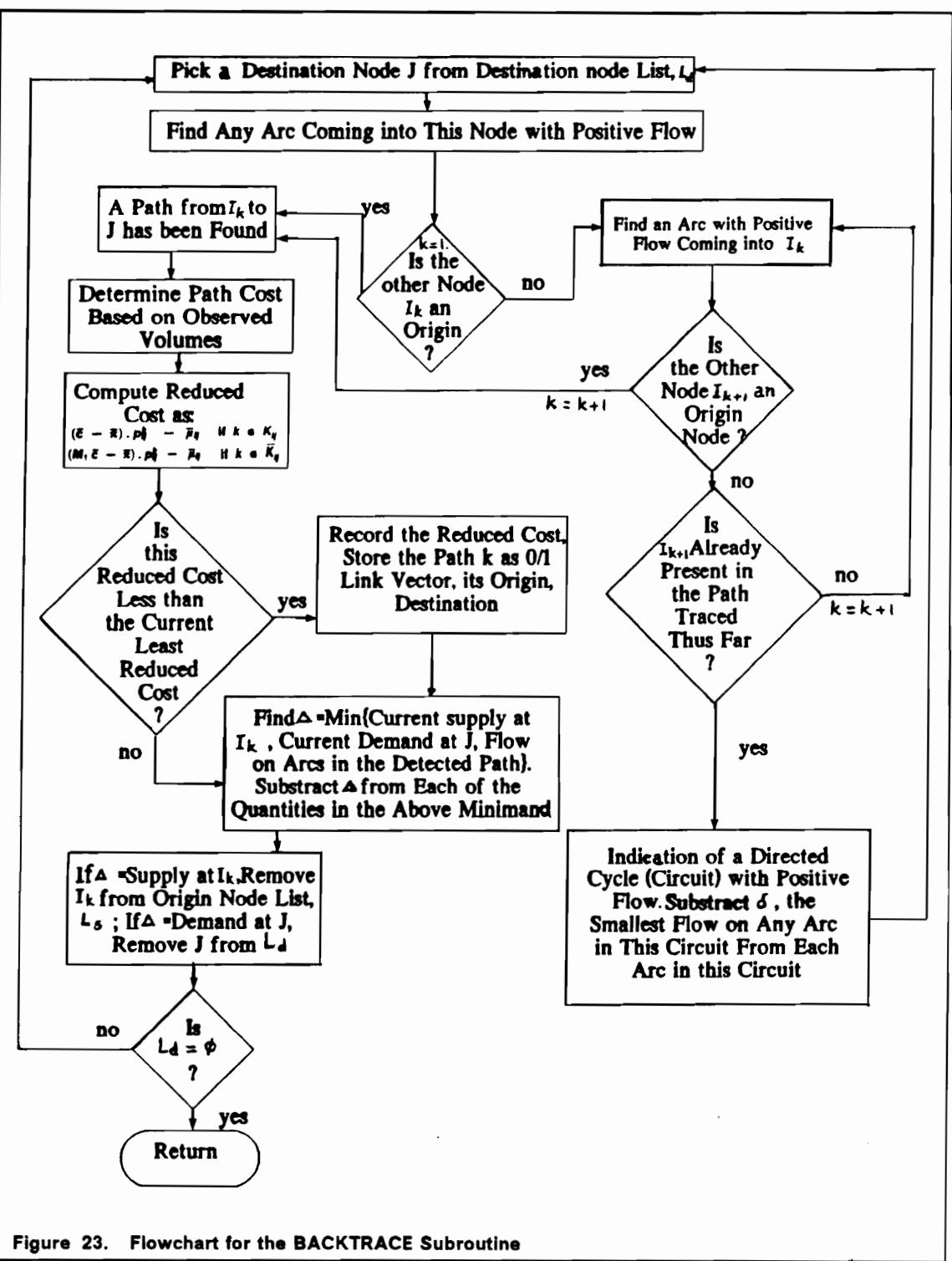


Figure 23. Flowchart for the BACKTRACE Subroutine

and is subtracted from the above three quantities. If the supply/demand at the origin/destination nodes becomes zero, these nodes are removed from the list, and the next destination node in the list is considered. On the other hand, if the incident node associated with the arc is a transshipment node, any arc with positive flow coming into this node is found. If the incident node associated with this arc is an origin node, the minimum of three quantities is found as above and the same procedure repeated again. In case the incident node has already been encountered in the current path construction, it implies that a cycle (circuit) with positive flow has been traced. The smallest flow on this cycle is detected and subtracted from all the arcs of this circuit, thus eliminating it. The procedure is then repeated by again considering the destination node list.

Thus, at each iteration of the backtracing algorithm, either a shortest path between an O-D pair is detected and accounted for, or else a circuit with positive flow is detected and eliminated. Since, the motivation here is to extract the O-D pair with the least reduced cost, the enumerated path  $k$  at each iteration is stored in the form of a 0,1 vector corresponding to the absence/presence of a link in the link list vector. The corresponding reduced cost for the nonbasic variable  $x_{ij}^k$  is also determined and stored, if its value is less than zero, implying a savings in cost (reduction in objective function value). In computing the reduced cost, the parameter  $M_1$  associated with deriving a user equilibrium needs to be considered, where appropriate. It is proved in Lemma 3 that the subproblem  $SP_I$  is a shortest path problem with nonnegative costs, and its solution yields a path that is the shortest for that O-D pair. Hence, when this subproblem is solved, the reduced cost is simply computed by  $v_{ij} = (\bar{c} - \bar{\pi}) p_{ij}^k - \bar{\mu}_{ij}$ , as given in [5.8]. However, when subproblem  $SP_{II}$  is solved, and if the shortest path obtained for an O-D pair is not the shortest based on observed volumes, then the reduced

cost is computed as  $v_{ij}^2 = (M_1 \bar{c} - \bar{\pi}) p_{ij} - \bar{\mu}_{ij}$ , as in [5.9]. The so computed reduced cost at each iteration is compared with the least value stored for an earlier iteration. If the current value is lesser, the current path, the corresponding O-D pair, and the reduced cost replace the already stored values. If not, the earlier values are retained. Thus at the end of one complete algorithmic iteration through the NETFLO and BACKTRACE routines, the most eligible nonbasic variable, its path, and reduced cost are at hand. The control is then passed to the MAIN program.

## C.8 Subroutine LEAVE

Having determined either an artificial nonbasic variable through subroutine ARTPRICE, or a legitimate nonbasic variable representing an O-D pair through subroutines NETFLO and BACKTRACE, to enter the basis, the next step is to determine the leaving basic variable. This is accomplished by subroutine LEAVE, the flowchart for which is shown in Figure 24.

The usual minimum ratio test is employed for this purpose. The first task in this routine is the determination of the updated column,  $B^{-1} Y_k$ , using the  $Y_k$  vector representing the column of the nonbasic variable,  $z_k$ , passed on from the ARTPRICE or the NETFLO/BACKTRACE subroutines, and the current  $B^{-1}$  matrix. Additionally, an initial check is made to see if there are any artificial variables in the basis at value zero, with the absolute value of the corresponding element of the updated column vector,  $y_k$ , not equal to zero. If so, a degenerate pivot is carried out to eliminate the artificial from the basis. If not, the usual minimum

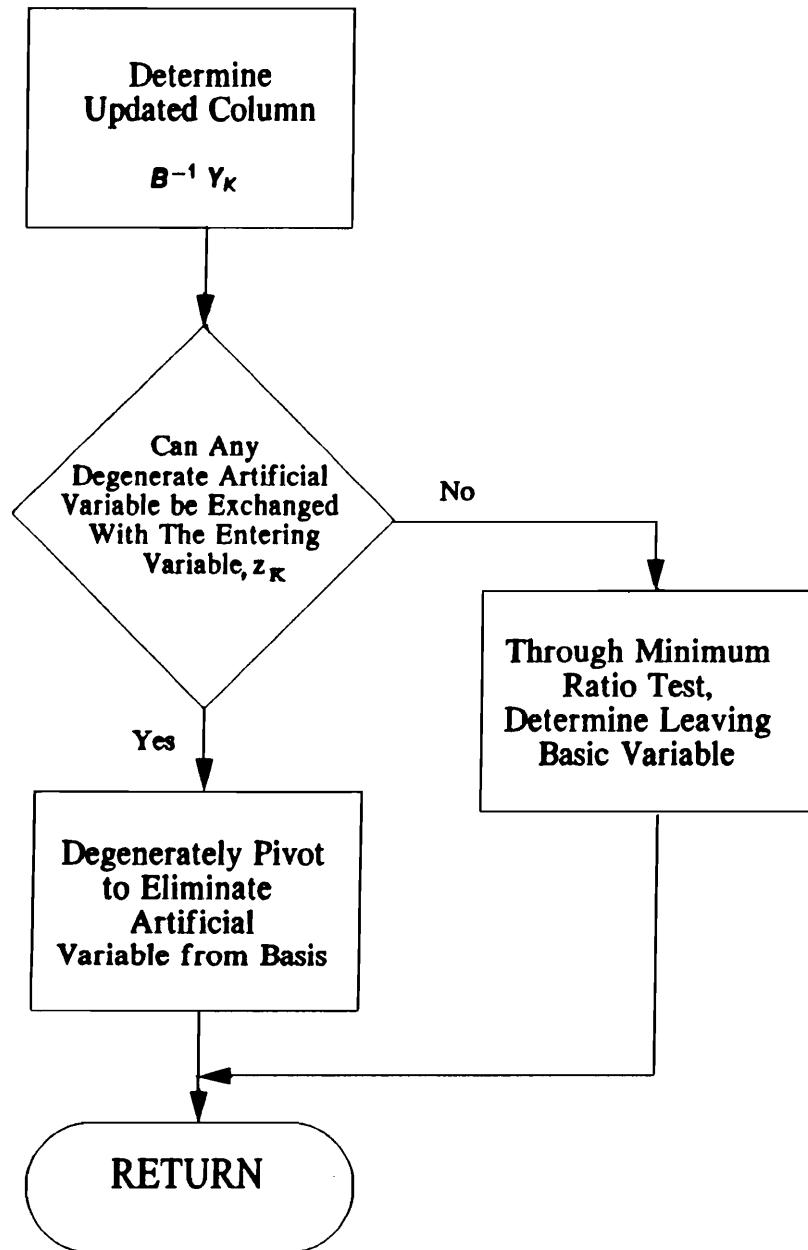


Figure 24. Flowchart for the LEAVE Subroutine

ratio test is carried out using the  $RHS$  and  $y_K$  vectors, to determine the leaving basic variable. Control is then passed to the MAIN program for updating the solution.

## C.9 Subroutine UPDATE

Once the leaving basic variable is determined in subroutine LEAVE, updating the solution is the next step. This is done by the UPDATE subroutine. The basic variable list is updated by substituting the leaving basic variable with the entering basic variable. The pivoting operation is carried out next, and  $B^{-1}$  is updated by row-by-row Gaussian elimination. The  $c_B B^{-1}$ , and  $RHS$  vectors are also updated in the process. The objective value is updated next, and the path comprising the basic variable is recorded, and control is transferred to subroutine ARTPRICE for pricing the artificial variables, thus commencing another iteration. The flowchart for the routine is shown in Figure 25.

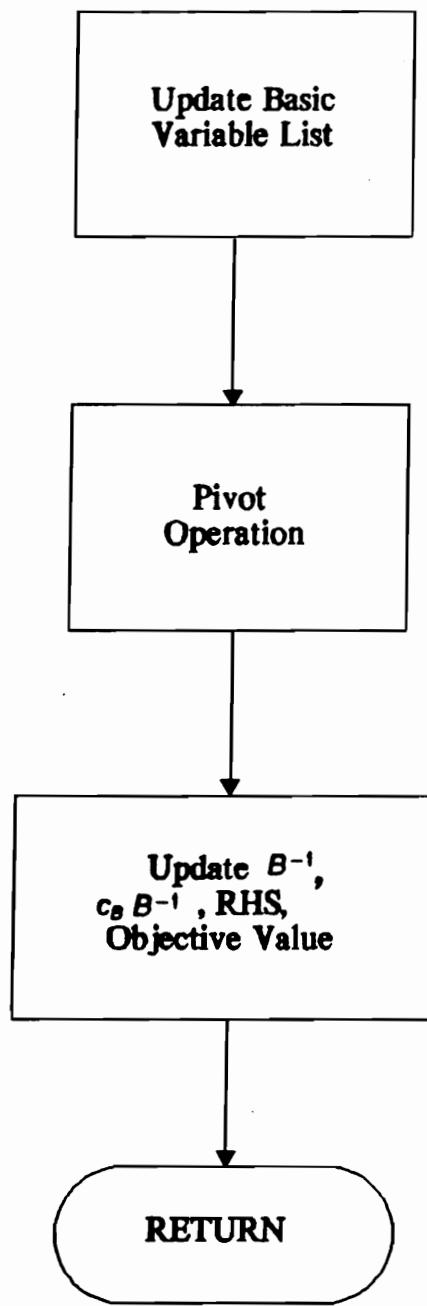
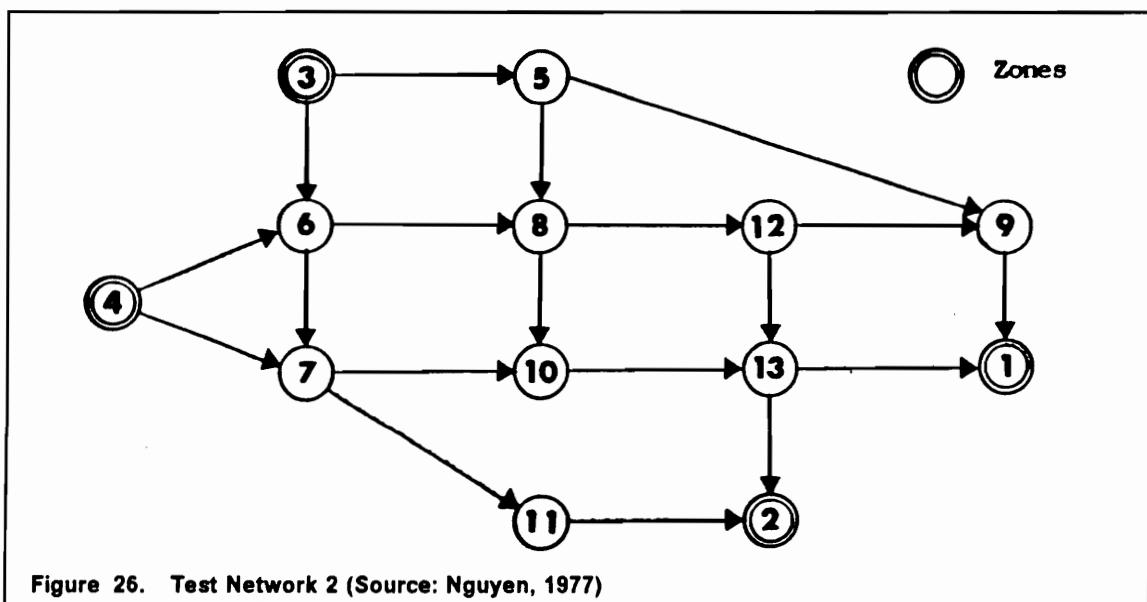


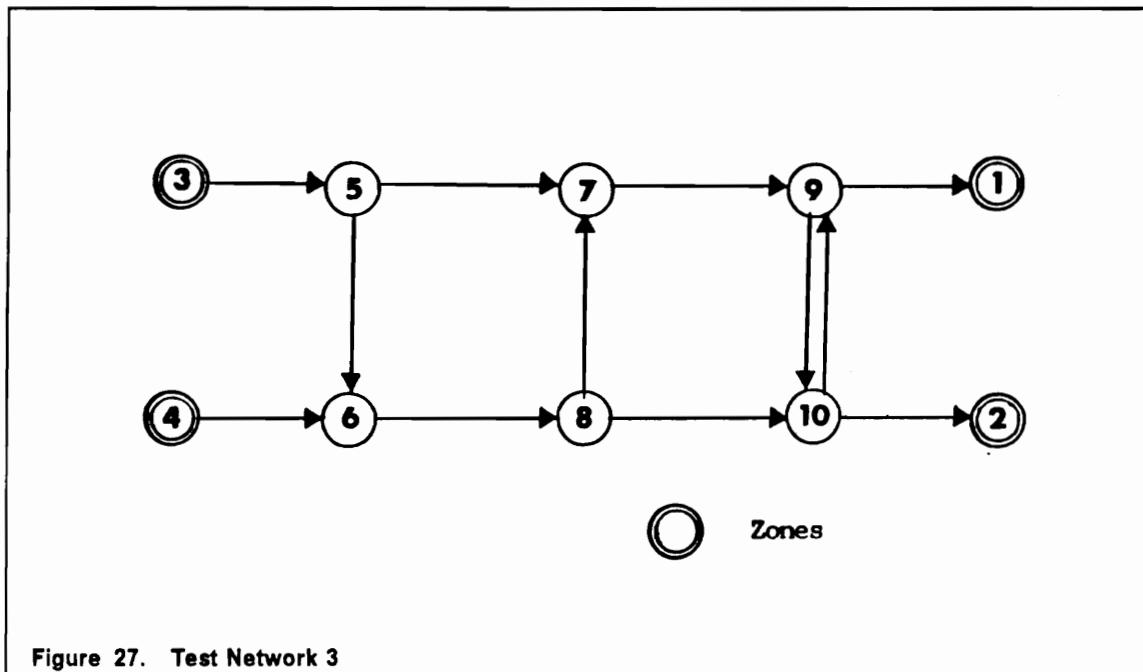
Figure 25. Flowchart for the UPDATE Subroutine

## Appendix D. Details of Additional Test Networks



**Table 36. Test Network 2 Characteristics (Source: Nguyen, 1977) (Data Slightly Modified)**

Link Number	Beginning Node	Ending Node	Observed Impedance	Free-Flow Impedance	Observed Volume
1	3	5	38	9	733
2	3	6	35	7	567
3	4	6	10	9	29
4	4	7	24	12	571
5	5	8	9	7	205
6	5	9	35	14	528
7	6	7	18	9	294
8	6	8	12	3	302
9	7	10	17	10	371
10	7	11	19	9	494
11	8	10	19	13	284
12	8	12	16	5	223
13	9	1	22	9	628
14	10	13	13	6	655
15	11	2	31	11	494
16	12	9	10	5	100
17	12	13	16	9	123
18	13	1	16	9	372
19	13	2	24	8	406



**Figure 27. Test Network 3**

**Table 37. Test Network 3 Characteristics**

Link Number	Beginning Node	Ending Node	Observed Impedance	Free-Flow Impedance	Observed Volume
1	3	5	7.54	5	950
2	4	6	5.19	4	950
3	5	6	6.06	6	202
4	5	7	17.16	8	748
5	6	8	17.36	7	1152
6	7	9	23.71	8	856
7	8	7	6.01	6	108
8	8	10	23.58	8	1044
9	9	1	6.78	4	1100
10	9	10	6.12	6	148
11	10	2	6.28	5	800
12	10	9	6.16	6	392

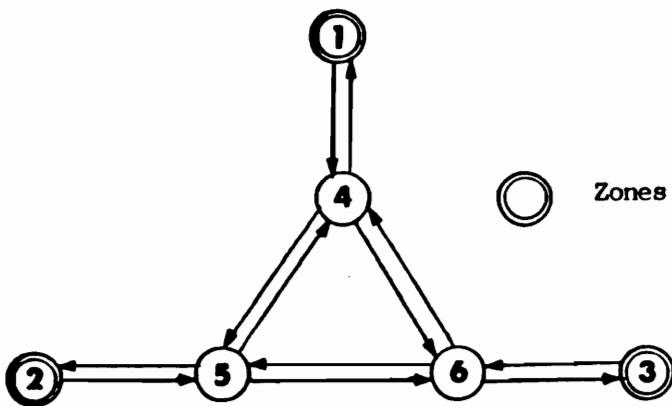
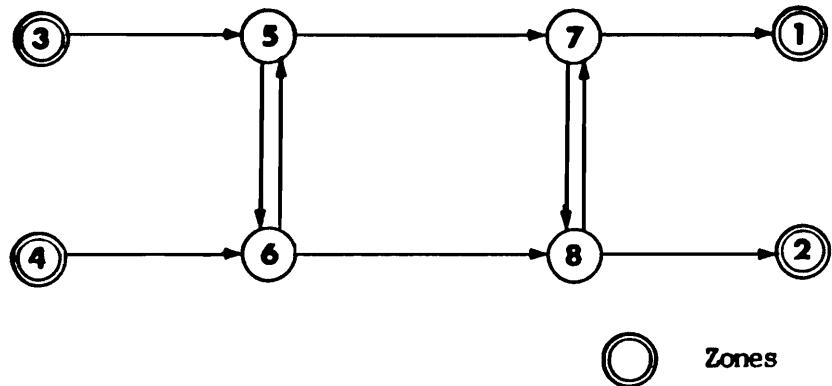


Figure 28. Test Network 4

Table 38. Test Network 4 Characteristics

Link Number	Beginning Node	Ending Node	Observed Impedance	Free-Flow Impedance	Observed Volume
1	1	4	4	3	1000
2	2	5	4	3	800
3	3	6	5	4	1000
4	4	1	5	4	750
5	4	5	9	6	600
6	4	6	16	9	400
7	5	2	6	4	950
8	5	4	3	2	350
9	5	6	7	5	700
10	6	3	5	4	1100
11	6	4	13	9	400
12	6	5	10	5	600



**Figure 29. Test Network 5**

**Table 39. Test Network 5 Characteristics**

Link Number	Beginning Node	Ending Node	Observed Impedance	Free-Flow Impedance	Observed Volume
1	3	5	6	4	1500
2	4	6	5	4	2000
3	5	6	5	3	100
4	5	7	10	6	1700
5	6	5	7	4	300
6	6	8	11	7	1800
7	7	1	8	5	1750
8	7	8	6	4	550
9	8	2	7	5	1750
10	8	7	6	4	600

## VITA

R. Sivanandan was born on January 29, 1959 in Tamil Nadu state in India. He obtained his Bachelor of Science degree in Civil Engineering in 1981 from Kerala University, India. He then joined the Master's program at the Indian Institute of Technology (I.I.T), Madras, India, and was awarded the M. Tech degree in Civil Engineering (specializing in transportation) in 1983. He gained over two and one half years of engineering and teaching experience in Madras, before coming to U. S. for pursuing the doctoral degree. He joined the Virginia Polytechnic Institute and State University (Virginia Tech) in Fall 1986, to specialize in transportation engineering. As a senior graduate student, he was employed as an Instructor, and was assigned various important tasks, including full responsibility of teaching undergraduate classes, and leadership role in a research project. During the course of this research work, he worked at the University Center for Transportation Research. He was married to Kausalya in August 1990. He completed all the requirements for the award of the Ph. D. degree in Fall 1991. He plans to devote his career to the field of transportation engineering.

A handwritten signature in black ink, appearing to read "R. Sivanandan", is written in a cursive style and is underlined.