NAME OF AUTHOR/*NOM DE L'AUTEUR* SAID EASA

TITLE OF THESIS/*TITRE DE LA THÈSE* A MODEL FOR INVESTIGATING TRAFFIC ASSIGNMENT AND CONTROL IN A FREEWAY CORRIDOR

UNIVERSITY/*UNIVERSITÉ* McMASTER UNIVERSITY

DEGREE FOR WHICH THESIS WAS PRESENTED/ *GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE* MASTER OF ENGINEERING

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE DEGRÉ* MAY 1976

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE* DR. B.L. ALLEN

DATED/*DATÉ* April 22, 1976 SIGNED/*SIGNÉ* Said M. Easa

PERMANENT ADDRESS/*RÉSIDENCE FIXE* 56 Bowman St., Hamilton, Ontario

A MODEL FOR INVESTIGATING TRAFFIC ASSIGNMENT AND CONTROL IN A FREEWAY CORRIDOR

A MODEL FOR INVESTIGATING TRAFFIC ASSIGNMENT AND CONTROL IN A FREEWAY CORRIDOR

by

SAID M. EASA

B.Sc. (Hons.), Cairo, 1972

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF ENGINEERING

McMaster University
Hamilton, Ontario
March, 1976

MASTER OF ENGINEERING (1976)

McMASTER UNIVERSITY

HAMILTON, ONTARIO

TITLE: A MODEL FOR INVESTIGATING TRAFFIC ASSIGNMENT AND CONTROL IN A FREEWAY CORRIDOR

AUTHOR: Said Mohamed Easa, B.Sc. (Hons.) Cairo, 1972.

SUPERVISOR: Dr. B. L. Allen

NUMBER OF PAGES: xii, 169

ABSTRACT

A procedure for predicting traffic behaviour in a freeway corridor is developed. Traffic demand can vary over time and is assigned to the freeway and surrounding network subject to the hypothesis that individual travellers will minimize their travel time. The impact of queueing time on minimum paths is included by utilizing a traffic diversion model. The model is capable of diverting all, some, or none of the traffic from a particular queueing path, and can therefore be used to investigate the effects of freeway entrance ramp control upon the adjacent road system.

A computer program was developed and incorporates several other new features for network traffic flow assignment. These include turning volume calculations without the requirement of separate turning links, the ability to impose turning movement prohibitions at critical locations, and a procedure for identifying illogical paths.

A new minimum path algorithm was developed to ensure that illogical paths were not used, and preliminary tests indicate that it is more computationally efficient than previous versions. Due to such improvements in the assignment procedure, link-node representation was simplified and use of the model requires considerably less coding effort as well.

The new model was fully tested using a relatively large freeway corridor network, and results show that it can be useful for evaluating ramp control strategies and predicting the resulting flows and queues corresponding to any desired case of diversion.

ACKNOWLEDGEMENTS

The author would like to express his sincere appreciation to Dr. B. L. Allen for his consistent encouragement and guidance throughout the entire period of this study. The helpful suggestions and big effort he made in reviewing this work are also acknowledged.

The author is grateful to the Ontario Ministry of Transportation and Communications for making available some materials and data used in this work.

The research was sponsored in part by a McMaster Graduate Fellowship in Urban Studies and the Ministry of Transportation and Communications, Ontario.

The author would like also to thank Miss D. Pitkin for typing the final copy of the thesis.

TO MY PARENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Urban transportation networks have long been subjected to the problem of recurrent traffic congestion during peak periods. In an attempt to prevent or reduce the effect of such congestion, especially on freeways, investigators and researchers have developed different types of control strategies [11]. The main emphasis of their strategies is to limit the access to the freeway so that congestion can be eliminated, thus allowing the facility to operate at the most efficient service level during the peak periods.

On the other hand, freeway control strategies including ramp metering or closure generally result in the controlled ramps having upstream queues which might interfere with surface street operations. Furthermore, drivers often divert from those queues to surface streets even though they experience larger travel times, and thus may create poor operations on the streets. Researchers have recognized this fact and emphasis has been concentrated on the operations of both the freeway and the associated surface streets.

> The effectiveness of an improvement plan must be viewed with respect to the total network being studied; it is not sufficient to provide congestion-free operations on the freeway at the expense of heavily congested interconnecting surface streets [1, p.85].

Obviously, if the interaction between freeway and surface street operations is to be investigated, and thus the total impact of improvement plans evaluated, an evaluation model is required.

One such model has been developed in the Bay Area Freeway Operations Study [1]. It is an analytic technique for evaluating freeway improvement plans, including the associated network of surface streets. Although this technique can provide system-wide measures of traffic performance, it requires given demands or volumes on network sections and known or estimated additional volumes on surface streets due to implementation of a certain improvement plan. This procedure could be used effectively only after completion of the traffic assignment phase in the transportation planning process.

Another model by Yagar [20] has been developed. It can be used in predicting flows and queues in a freeway corridor experiencing time-varying demand by assigning traffic according to the principle of minimum individual travel cost. It can also be used in assessing the system-wide effects of any proposed changes to the network. However, the model does not include such features as traffic diversion or turn prohibitions which are necessary if actual traffic behaviour is to be well represented, or if illogical paths within the network are to be avoided. In addition, the link-node representation adopted in the model tends to be complicated.

The Brown and Scott model [2] is another traffic assignment model which seemingly could be used in studying a problem of our concern. However, the model does not account for time-varying demands, and although it uses a link-node representation which avoids illogical paths, such representation requires a large coding effort. Furthermore, no traffic diversion is included in the traffic assignment procedure of the model.

Due to these deficiencies, it is felt that a satisfactory corridor model for evaluating control strategies has not yet been developed. The requirements for such a model should include the following:

1. Simple Link-Node Representation

    The model should be able to use a simple link-node representation of the corridor so that the coding effort required by the user of the model may be reduced or the size of the corridor increased.

2. Turn Prohibitions

    In connection with the requirement of simplicity in the link-node representation, the model should be able to avoid illogical paths such as those between exit and entrance ramps. It should also allow for the use of network controls such as prohibition of a left turn.

3. Traffic Diversion

    The model should account for traffic diversion from queueing minimum paths to more accurately reflect driver behaviour. Through this procedure a better prediction of the impact of control strategies on surface streets can be obtained and quantified.

4. Accuracy

The model should represent accurately the time variation in traffic demands especially for the peak periods where small oversaturation can be easily detected. Also, a microanalysis is necessary to emulate the traffic operation of the critical sections in the corridor. In addition, the accuracy requirement necessitates the model to be able to avoid illogical paths in the assignment procedure.

It is, therefore, the purpose of this paper to develop a corridor model which meets all of the preceding requirements. A portion of the basic assumptions and framework of Yagar's procedure was used in the development. However, the model described herein represents a significant improvement over previous work by virtue of several new features.

To more realisticly predict driver behaviour, the model incorporates a new procedure for traffic assignment. Homburger's method [8] assigns all demand to the nonsaturated minimum paths, while Yagar's method accounts for congestion cost on saturated links and traffic could be assigned to saturated paths if they result in a lesser travel cost. Due to the inconvenient situation of stop-and-go operations on a saturated minimum path, some drivers will prefer to use a nonsaturated minimum path even if they experience a larger travel cost. A traffic diversion procedure was developed and used for this purpose. Options are provided for total diversion, no diversion, and partial diversion.

Additionally, the model incorporates the following features:

1. A method for simplifying link-node representation was developed. This method reduces the amount of coding effort required by the users.
2. A totally new minimum path algorithm with turn prohibitions was developed to automatically identify and avoid network illogical paths.
3. For purposes of assigning queued demand on a certain link to minimum paths in the following time slice, (the peak period is divided into equal lengths of homogeneous demands called time slices), the model considers the origin as the upstream node of that link and a new procedure for overlapping the minimum paths of queued demands was developed to prevent illogical paths which might occur in assigning that queued demand in different time slices.
4. There is allowance in the model for the user to specify the intersection nodes at which turning volumes are desired. The calculated turning volumes are printed on a drawing of the corresponding intersection.

For any model of this type, it is apparent that the form and structure of the minimum path algorithm is extremely important. Not only does it indicate the accuracy with which traffic is actually assigned to a network, but it dictates the basic form and structure of the entire procedure. For this reason, it seems logical to first investigate the minimum path algorithm and to then proceed with discussions of the basic model structure.

Therefore, in Chapter 2, an appraisal of existing minimum path algorithms and a detailed description of the new one are given. An indication of the efficiency of the new algorithm is also given.

The basic model structure is described in Chapter 3. This includes a proposed method for representing a corridor network, model assumptions, and treatment of queues, costs, and capacities.

In Chapter 4, a traffic diversion procedure is suggested to account for queue existence in the minimum paths. Included also is a detailed description of the different illogical paths that might occur in a network, along with a method for identifying these paths automatically in the computer program. This chapter includes as well, a method for calculating turning volumes without the use of turning links. Description of the logical sequences of the model and fields to which the model might be applied are included at the end of the chapter.

Chapter 5 presents an application of the model to a real corridor using hypothetical data. A discussion of the results is included.

Finally, a summary of the work presented, conclusions drawn, and suggestions for further research are given in Chapter 6.

Included in the appendices are descriptions of the methods adopted in automatically determining network illogical paths, and a description of the method used for calculating turning volumes without using turning links. Instructions for using the computer program, and a complete listing of the computer program along with definitions of important variables used in the program are also included.

CHAPTER 2

A NEW MINIMUM PATH ALGORITHM WITH TURN PROHIBITIONS

2.1 Introduction

There are two main groups of minimum path algorithms--matrix methods and tree methods. In the former, minimum paths between every node and all other nodes in the network are determined simultaneously. In the latter, minimum paths from a given node to all other network nodes are calculated separately. Since only some of the minimum paths are required in most transportation networks and most of these networks are sparsely connected, the tree methods are usually more efficient.

The tree-building algorithms may be used for transportation networks with or without turn penalties and prohibitions. However, some of the former requires additional effort in network coding and others are invalid. In addition, none of the existing algorithms allows more than one link in the same direction to have the same upstream and downstream nodes. Such a feature would be useful in simplifying representation of some network sections, such as merging, weaving, interchanges and intersections.

The basic purpose of this chapter is to develop a minimum path algorithm which accounts for network turn prohibitions without requiring additional coding effort. In fact, the proposed method reduces the coding effort normally required of conventional link-node representation. Testing of the algorithm procedure and computational efficiency is given.

2.2 Appraisal of some Existing Methods

Kirby and Potts [9] evaluated different minimum path algorithms with turn penalties and prohibitions and it is worthwhile to repeat some of their findings here. One obvious method for correctly allowing for turn penalties and prohibitions is to use links in the network for these turns [17] as shown in Fig. 2-1. Turn prohibitions could be accounted for by omitting the corresponding links from the network. Although this method accounts correctly for turn penalties and prohibitions it requires a large amount of network coding and computer storage.

Caldwell's method [4] provides allowance of turn penalties and prohibitions by constructing a pseudo-network in which nodes represent the original links and links are represented by "hooks". Turn prohibitions can be taken into account by using their hooks with infinite penalties. As stated by Caldwell it is not necessary for a minimum path from an origin node to a destination node i, passing through a node j, to coincide with the minimum path from that origin node to node j. This is a fundamental concept for networks with turn penalties and prohibitions. This concept is illustrated using the simple example shown in Fig. 2-2(a). The costs are indicated on the links and all

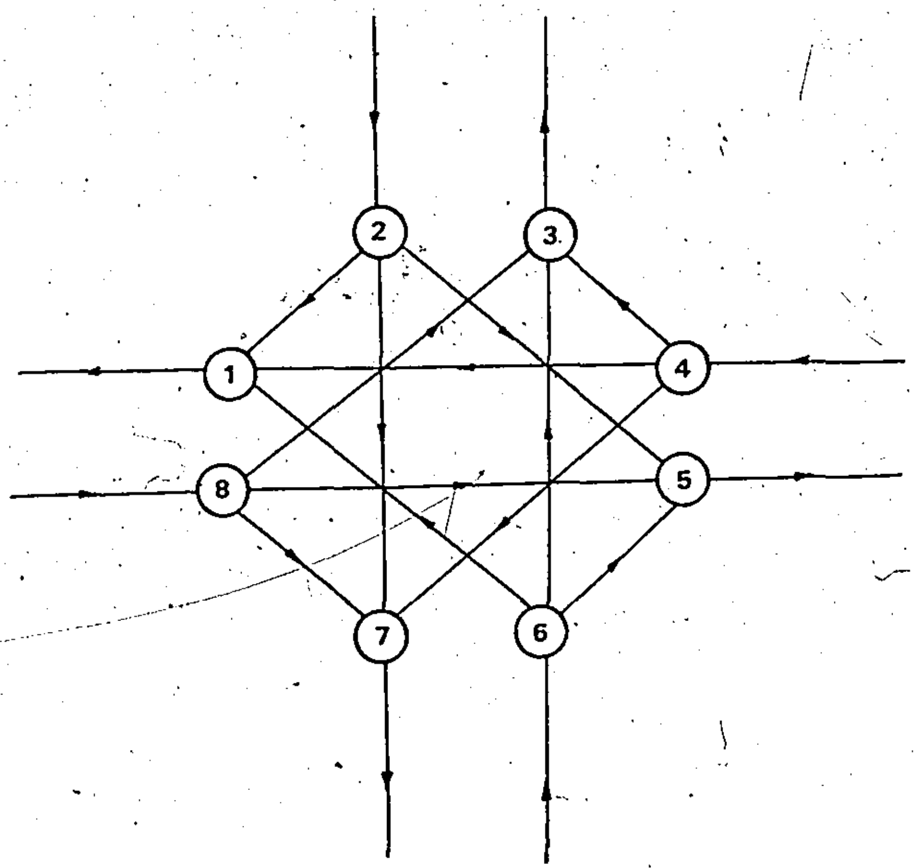


FIG. 2.1 CONVENTIONAL LINK-NODE REPRESENTATION OF AN INTERSECTION

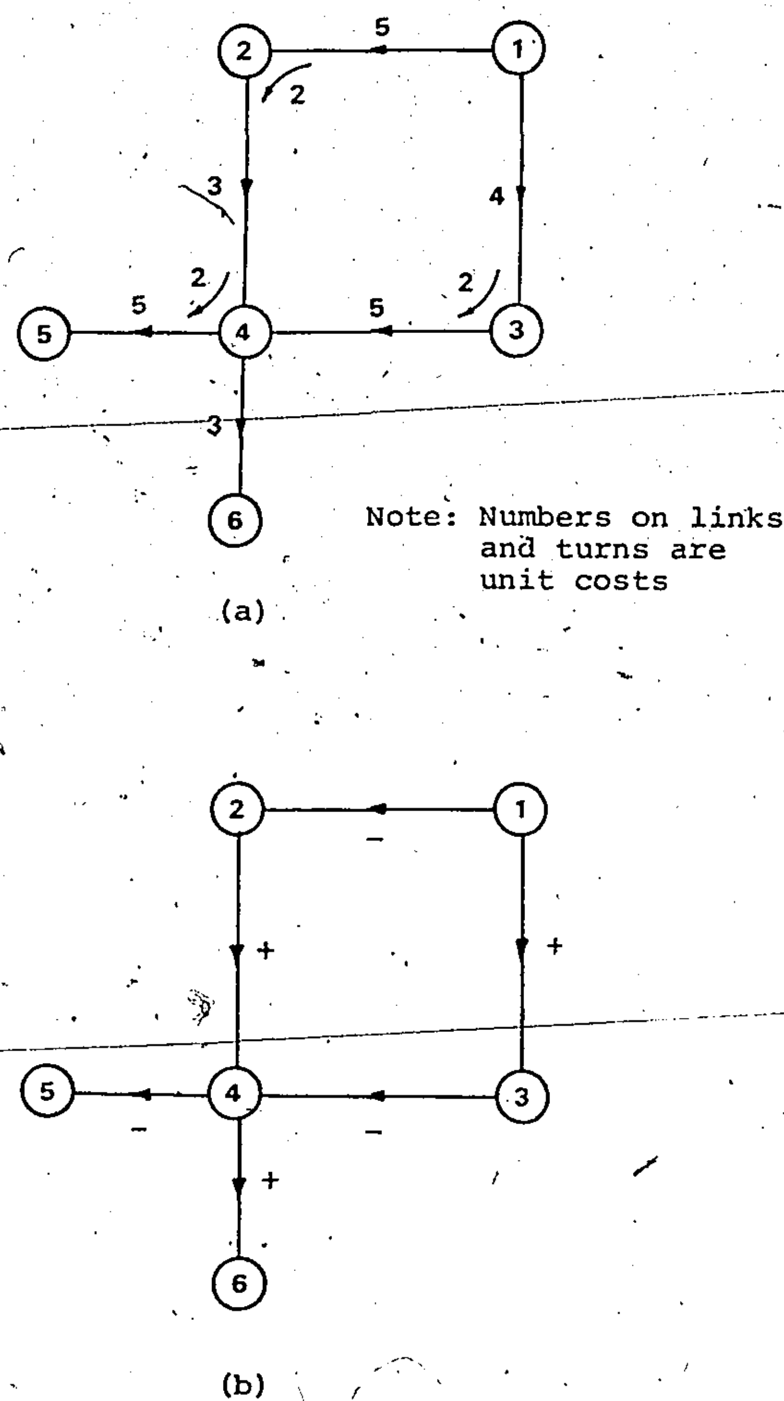
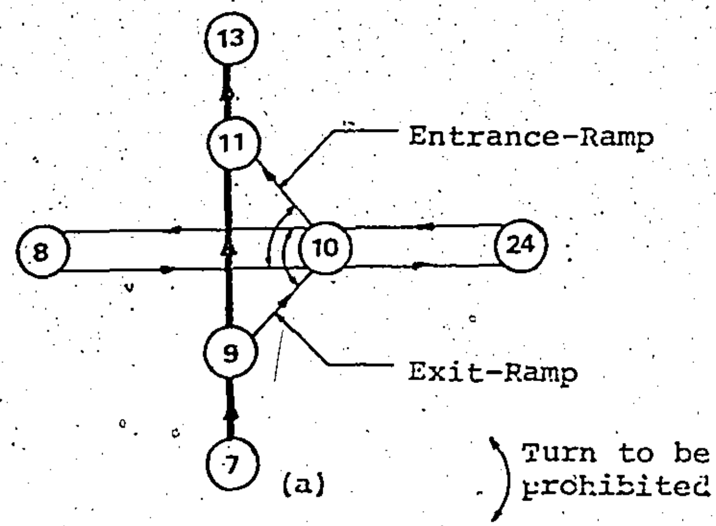


FIG. 2.2 EXAMPLE OF NETWORKS INCLUDING TURN PENALTIES

turns have a constant penalty of 2 units. The minimum path from node 1 to node 4 is 1-2-4, whereas the minimum path from node 1 to node 5 is 1-3-4-5. Although Caldwell's method accounts correctly for turn penalties and prohibition, it is not widely used due to the extra effort required by the user to construct the pseudo-network.

There are other methods which include turn penalties by adding a constant cost for any change of direction at an intersection [3,5]. This could be achieved by labelling each north-south link with a 'plus' sign and each east-west link with a 'minus' sign as shown in Fig. 2-2(b). However, it is possible that such a procedure could produce invalid results. For the example in Fig. 2-2(a), this method would obtain the minimum path between nodes 1 and 4 as 1-2-4 and subsequently discard the alternate route 1-3-4. Therefore, the resulting minimum path between nodes 1 and 5, would be incorrectly chosen as 1-2-4-5.

Additionally, it should be noted that one must be careful when accounting for turn prohibitions by adding links with infinite costs [18]. In the example illustrated in Fig. 2-3(a), it is necessary to avoid the illogical paths between links 9-10 and 10-8, and between links 8-10 and 10-11. To accomplish this, dummy nodes 36 and 37 are added and connected to node 10 with dummy links of very high costs as shown in Fig. 2-3(b). Although the illogical paths seem to be prohibited, this method allows the U-turn at node 24 which is illogical. Therefore, the illogical paths under consideration can still be performed through that U-turn.



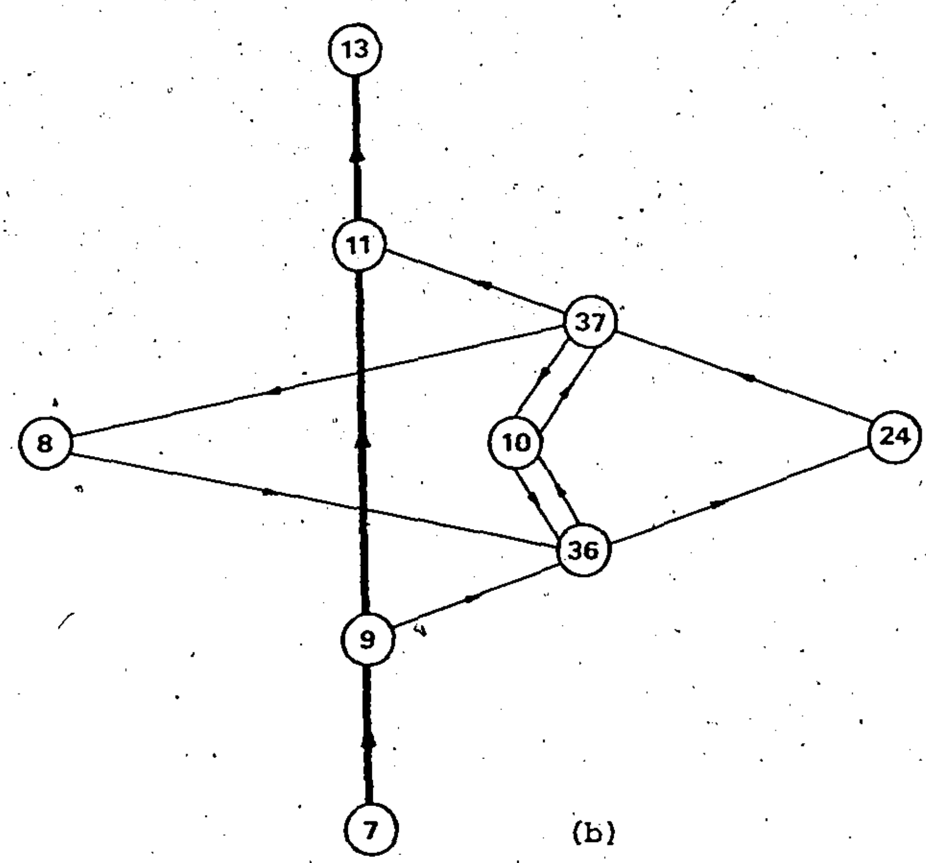


FIG. 2-3 EXAMPLE OF USING DUMMY NODES AND LINKS TO ELIMINATE ILLOGICAL PATHS

Clearly, if these problems are to be avoided, an improved algorithm is required.

2.3 Description of the Proposed Algorithm

Not only does the algorithm described in this section avoid all deficiencies of other methods, it simplifies representation of certain network sections.

The algorithm is based on the following important definitions:

a) Node-Absolute Minimum Cost

Each of the paths from an origin to a given node may have different costs, each cost corresponding to one upstream link of that node. The minimum of these costs is defined as the node-absolute minimum cost. This is the cost considered in the minimum path algorithms without turn penalties and prohibitions.

b) Link-Related Minimum Cost

At any node having turn prohibitions, each downstream link has its link-related minimum cost for later consideration in the minimum path procedure. It is the minimum of the costs corresponding to the upstream links of that turn prohibition node, except those links which have turn prohibitions with that downstream link.

Suppose one wishes to determine the minimum paths between node 1 and nodes 11 and 12 in the network with known link travel costs shown in Fig. 2-4. Assume turns are prohibited between the following pairs of links:

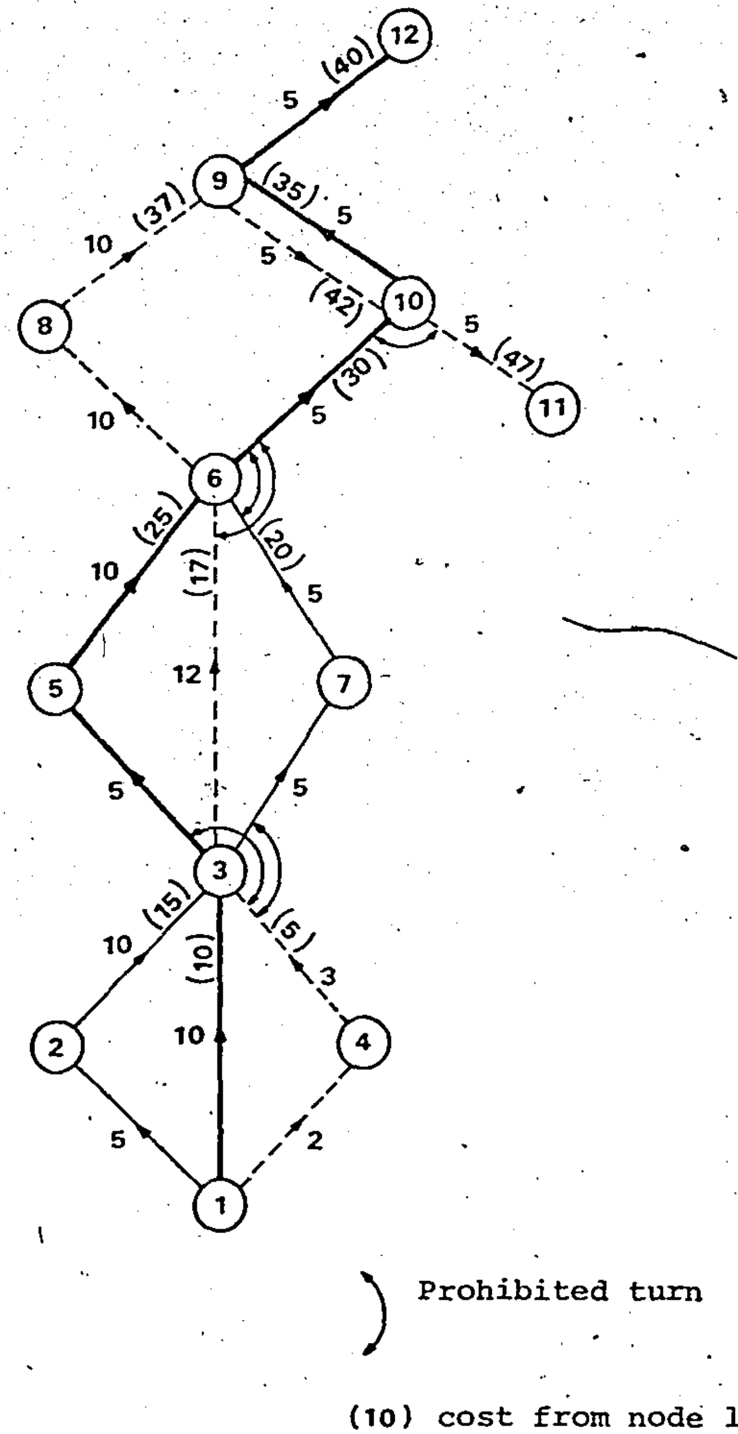


FIG. 2-4 A GIVEN EXAMPLE OF MINIMUM PATHS WITH TURN PROHIBITIONS

From	To
4- 3	3- 5
4- 3	3- 7
3- 6	6-10
7- 6	6-10
6-10	10-11

There are three paths from node 1 to node 3, with the associated costs:

Path	Cost
1-4-3	5
1-3	10
1-2-3	15

Three downstream links of node 3 are found. Link 3-5, however, may not be associated with link 4-3 due to the turn prohibition. Therefore, that link must use its link-related minimum cost, which in this case is 10. One possible path between nodes 1 and 6 is, therefore, 1-3-5-6, with a cost of 25.

Link 3-6 has no turn prohibition with any of the upstream links of node 3 and thus it will use the node-absolute minimum cost of 5. Therefore, another possible path is 1-4-3-6 with a cost of 17.

Similarly, the link-related minimum cost of link 3-7 is 10, giving a cost of 20 on the path 1-3-7-6.

Now node 6 has three costs corresponding to the entering links:

Link	Cost
3-6	17
7-6	20
5-6	25

If one now considers link 6-10, the link-related minimum cost is 25 via link 5-6. Therefore, the cost of node 10 through link 6-10 is 30. Link 6-8 uses the node-absolute minimum cost of 17 via link 3-6. The cost of node 9 through 6-8 and 8-9 is, therefore, 37.

Link 10-9 uses the link related minimum cost of 30 (U-turn is considered a turn prohibition), giving node 9 another cost of 35. Link 9-10 uses the link related minimum cost of 37, giving node 10 another cost of 42.

Now nodes 9 and 10 have the following costs:

node 9		node 10	
link	cost	link	cost
10-9	35	6-10	30
8-9	37	9-10	42

Similarly, link 10-11 uses the link-related minimum cost of 42 giving node 11 a cost of 47 which is the minimum cost from node 1 to 11 through nodes: 1-4-3-6-8-9-10-11.

Link 9-12 uses the node-absolute minimum cost of 35, giving node 12 a cost of 40 which is the minimum cost from node 1 to 12 through nodes: 1-3-5-6-10-9-12.

The above example shows the following important points:

a) The minimum path between nodes 1 and 12, as shown by the solid lines, does not necessarily coincide with the minimum path between the intermediate nodes 1 and 3 or between nodes 3 and 6. In addition, it is not necessary that this minimum path coincide with the second minimum path as exemplified by the path between nodes 3 and 6.

b) The path shown by dashed lines from node 1 to node 11 does not coincide with the path between nodes 1 and 12, although they have several common nodes.

c) More than one cumulative cost to a node must be considered for both the turn prohibition nodes, and at those nodes having a U-turn (node 9).

Based on these observations, the minimum path algorithm should be structured to contain forward and backward procedures.

The forward procedure is performed from the origin under consideration to all other nodes in the network as follows:

1. Each node at which there are turn prohibitions (including nodes having U-turns) is identified and given four costs, each having an infinite value (practically it is assumed to be 9999 units of cost). The origin is given a cost of zero.
2. For each turn prohibition node four upstream link numbers (each having an initial value of zero) are stored.
3. When a link is considered in the minimum path its upstream node cost is calculated as follows:
   a) If that link has turn prohibition with upstream links of its upstream node, it uses its link-related minimum cost.
   b) If no turn prohibitions exist, it uses the node-absolute minimum cost.

4. Using the cost of the upstream node determined in the previous step, the resulting cost at the downstream node is obtained. Then two cases are found:
   a) If that downstream node is not a turn prohibition node and the cost of that node is lowered, the link number and its associated cost are stored, otherwise that link is cancelled.
   b) If that node is a turn prohibition node the link number and its associated cost are stored unless that link had been previously considered and had a lower cost. The costs of that turn prohibition node are then arranged in ascending order.
5. When the link is not cancelled in the previous step the downstream links of its downstream node are stored for later consideration.
6. For each stored link in step 5 repeat steps from 3 through 6. The procedure continues until the links stored for latter consideration are all considered. After that each destination has a unique cost and an immediate upstream link corresponding to the minimum path from the origin to that destination.

The backward procedure is performed from each destination to the origin under consideration to finally identify the minimum path. (During this procedure the first congested point in the path is determined for later consideration in the assignment procedure). The procedure is as follows:

1. The upstream link and node numbers of the destination corresponding to the minimum path are obtained.

2. If the upstream node is not a turn prohibition node its upstream link is obtained. (Each of these nodes has only one cost which is the minimum to it from the origin and one upstream link corresponding to that cost).
3. If the upstream node is a turn prohibition node, two cases are considered:
   a) If there is a turn prohibition between the previously determined link and any of that node upstream links, the link corresponding to its link-related minimum cost is considered with its upstream node.
   b) If there is no turn prohibition, the link corresponding to the node absolute minimum cost is considered with its upstream node.
4. Repeat steps 2 and 3 until the origin is reached.

As is usual practice the number of links entering a turn prohibition node is restricted to 4 while the number of links leaving that node can be any number. However, more than four entering links can be accommodated, if necessary, through slight changes in the method.

The method could account for turning costs without using links for these turns through a straight forward combination of this algorithm and the method adopted for turning volume calculation without using turning links (to be discussed in Chapter 4). The resulting algorithm would not necessitate links indications to identify turns. It could also allow for turn penalties which are constant or proportional to the turning volumes. However, this procedure has not been developed in this thesis and is left for further research.

2.4 Testing the Algorithm

The proposed minimum path algorithm is applied to a network with and without turn prohibitions. The purpose of this application is to present minimum paths for a study network to clarify the method. In addition, it is intended to give an indication of the computer time used when the method is applied to a network with and without turn prohibitions.

The network studied is shown in Fig. 2-5. At first, the network was considered free from turn prohibitions. The minimum paths between origin 1 and destinations 4,5, 6,7, 8, and 9 were obtained. These paths are shown by solid lines in Fig. 2-5. It can be seen that each node has a unique minimum cost corresponding to one upstream link which gives that minimum cost from the origin node 1. The minimum cost from node 1 to each node is shown in brackets.

In Fig. 2-6 the same network was considered with the required turn prohibitions. These turn prohibitions are determined automatically in the program, and are indicated in Fig. 2-6. It can be noted that the network contains 37 turn prohibitions including U-turns. Turn prohibitions at nodes 4, 5, and 6 are not included because they are origin-destination nodes. Illogical paths through them are avoided using another procedure to be discussed later in Chapter 4.

The minimum paths considering turn prohibitions are shown in Fig. 2-6. In this case, each turn prohibition node has a maximum of four costs which are shown in brackets. Each downstream link at any of these nodes will use its link-

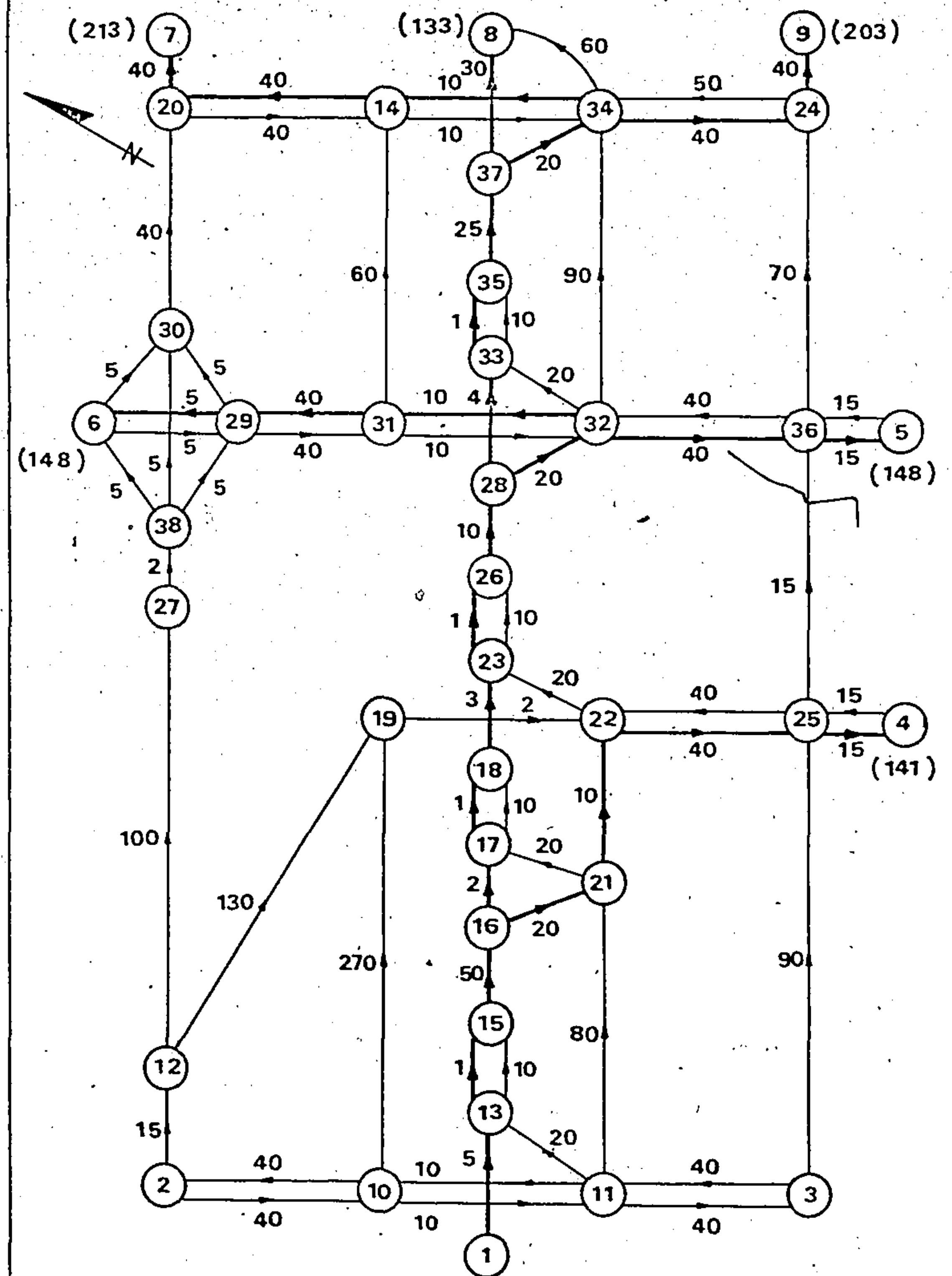


FIG. 2.5 MINIMUM PATHS IN STUDY NETWORK WITHOUT TURN PROHIBITIONS

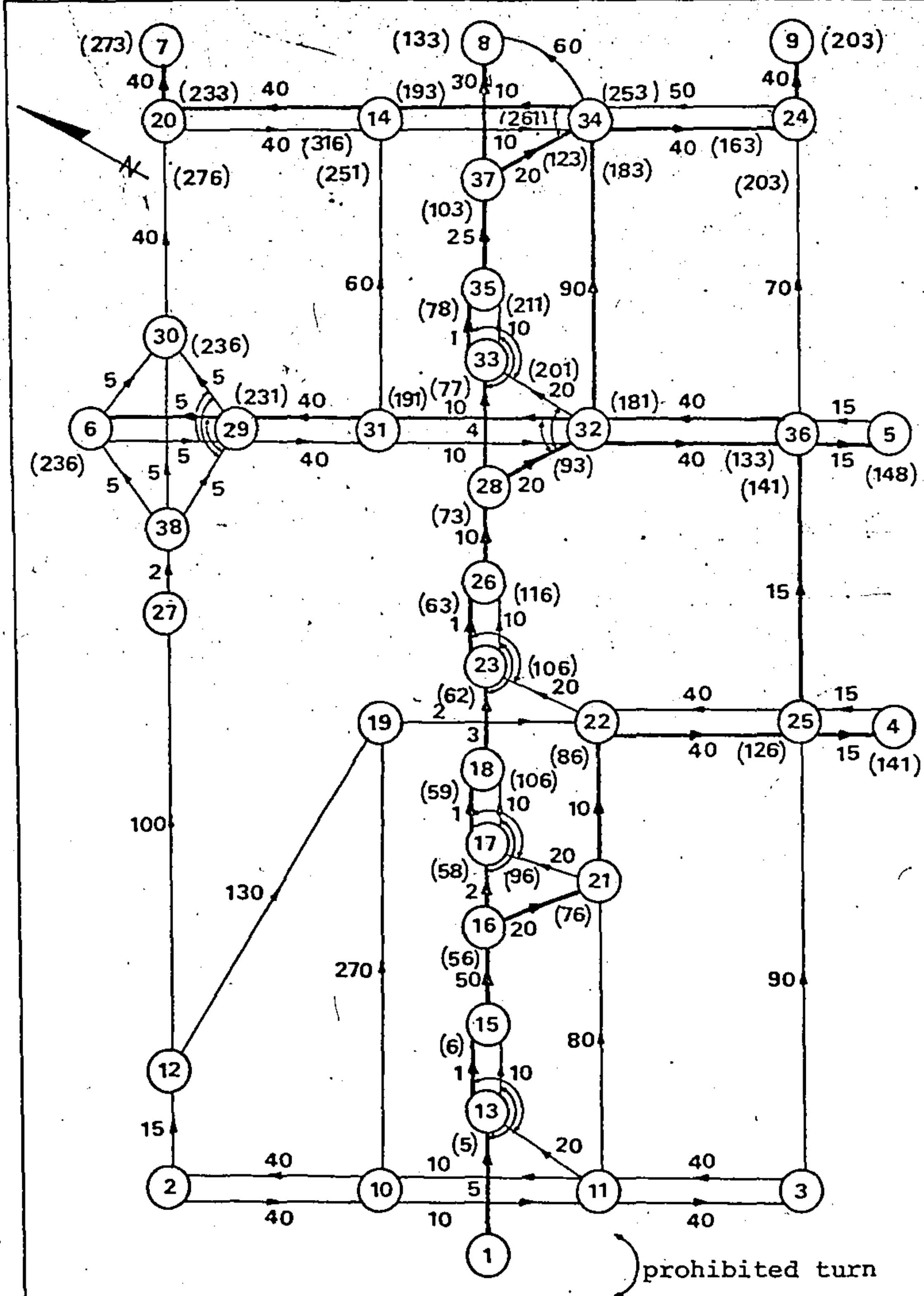


FIG. 2.6 MINIMUM PATHS IN STUDY NETWORK WITH TURN PROHIBITIONS

related minimum cost if it has turn prohibition with any of the upstream links of the node, otherwise it will use the node-absolute minimum cost. For example, at node 34, there are four costs corresponding to the following links:

Link 37-34	Cost = 123
Link 32-34	Cost = 183
Link 24-34	Cost = 253
Link 14-34	Cost = 261

When link 34-14 is considered in the minimum path its upstream node cost is 183 (its link-related minimum cost) and not 123, since there is turn prohibitions between that link and links 37-34 and 14-34. Therefore, link 34-14 gives a cost of 183 + 10 = 193 at node 14. When considering link 34-24 its upstream node cost is 123 (its link-related minimum cost), since there is a turn prohibition between that link and the upstream link 24 -34. Therefore, the cost at node 24 due to link 34-24 is 123 + 40 = 163.

Similarly, for node 32, link 32-34 uses the node-absolute minimum cost 93, while each of links 32-31 and 32-33 use its link-related minimum cost 181. Therefore, in the backward procedure, the minimum path from node 1 to nodes 7 and 9 are respectively:

1-13-15-16-17-18-23-26-28-32-34-14-20-7

1-13-15-16-17-18-23-26-28-33-35-37-34-24-9

It can be seen that the two paths are different upstream of the common node 34. This obviously could not occur in a network without turn prohibitions. The minimum paths to all destinations are given in Table 2-1.

It can be noted that the network contains 38 nodes and 71 links. The computer execution times for the network with and without turn prohibitions were respectively 45 and 35 seconds on a CDC6400. In the two cases, this time was used in obtaining 576 minimum paths, as well as performing the assignment and other activities of the model. Therefore, the increase in the computer time due to consideration of turn prohibitions is less than 30% for 37 turn prohibitions.

From the algorithm testing, the following conclusions have been drawn:

a) The increase in computer time is expected to be proportional to the number of turn prohibitions in the network. This is advantageous over other methods where the computer time was approximately doubled regardless of the number of turn prohibitions used [15]. Therefore, it is recommended that the algorithm can be used even with a few prohibited turns.

b) The algorithm was found to function properly under any number of turn prohibitions at a node. Therefore, there is no limit to this number.

c) A method for automatically identifying the network turn prohibitions considering information provided for queue spillback on links was developed and will be described later. Therefore, the algorithm does not require direct provision of turn prohibitions in the input data.

d) The algorithm allows the use of a simple link-node

TABLE 2-1: MINIMUM PATHS FROM NODE 1 FOR THE NETWORK WITH TURN PROHIBITIONS

Node number in the path from origin to destination	Destination Node Number					
	4	5	6	7	8	9
	1	1	1	1	1	1
	13	13	13	13	13	13
	15	15	15	15	15	15
	16	16	16	16	16	16
	21	17	21	17	17	17
	22	18	22	18	18	18
	25	23	25	23	23	23
	4	26	36	26	26	26
		28	32	28	28	28
		32	31	32	33	33
		36	29	34	35	35
		5	6	14	37	37
				20	8	34
				7		24
						9

representation at different network sections. It permits two or more links in the same direction to have the same upstream and downstream nodes. This feature is extremely useful in effecting simplicity of network coding.

e) The new algorithm is at least as efficient as previous ones since the simplified network representation results in a decrease of computer time, which is likely to exceed the increase resulting from the turn prohibition procedure.

CHAPTER 3

BASIC MODEL STRUCTURE

3.1 Introduction

One essential task in a traffic assignment procedure is to represent the transportation network in terms of a set of links and nodes. It is desirable to use the most efficient representation possible, since either the network coding effort would be reduced or a larger size of network may be handled. One purpose of this chapter is to describe a method of network representation which accomplishes that efficiency.

The model allows the peak period to be represented by equal-length time slices, each having a constant rate of exogenous demands. These demands are assigned to a network corridor using the principle of individual travel cost minimization, where the minimum cost path may contain some time in queue (cost is treated throughout as travel time). Another purpose of this chapter is to explain the assumption and approximations contained in the model and to describe in detail how the model treats flows, queues, costs and capacities in the assignment procedure.

3.2 Representation of a Physical Corridor Network

A corridor network, as defined herein, consists of main roadways and connecting roadways approximately normal to the main roadways. The main roadways are uni-directional since the only flows considered are in the direction of major flow in the peak period. The connecting roadways may be bi-directional. Both the main roadways and connecting roadways may be freeways or arterials. At the intersection of main roadways and connecting roadways there may be a grade-separated interchange with ramps, or a signalized intersection if the intersection is at grade.

Fig. 3-1 shows a hypothetical northbound corridor network. This corridor consists of a freeway A-B-C and two arterials D-E-F and G-H-I. The freeway has interchanges at A, B and C. Each of the two arterials has three intersections, D, E, F and G, H, I respectively. These are connected to freeway interchanges with bi-directional streets IC, CF, HB, BE, GA, and AD.

In the link-node representation, each of the corridor roadways are represented by links which start and end at nodes. These nodes occur at points where demand, flow, and/or roadway characteristics change. However, if the roadway has uniform flow, it is considered as a single link even though it might have varying capacities along its length. The capacity in such a case is represented by the minimum section capacity. If the roadway meets other links or has exogenous flows at certain points, nodes are provided at these points.

The link-node representation of the hypothetical corridor in Fig 3-1 is shown in Fig. 3-2. A detailed description of the different components of the corridor is given in Table 3-1.

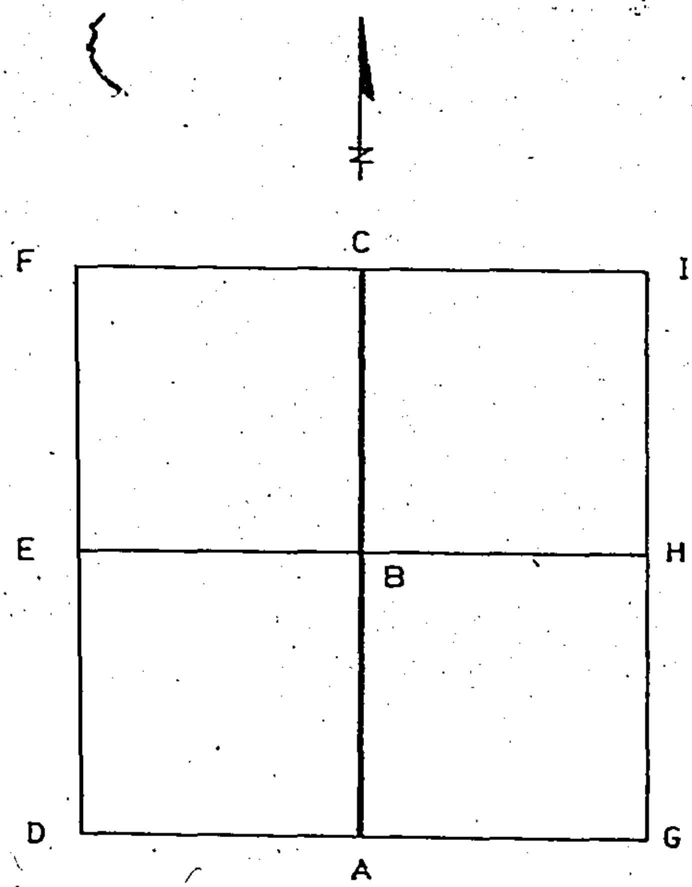


FIG. 3-1 A HYPOTHETICAL NORTHBOUND DIRECTIONAL CORRIDOR

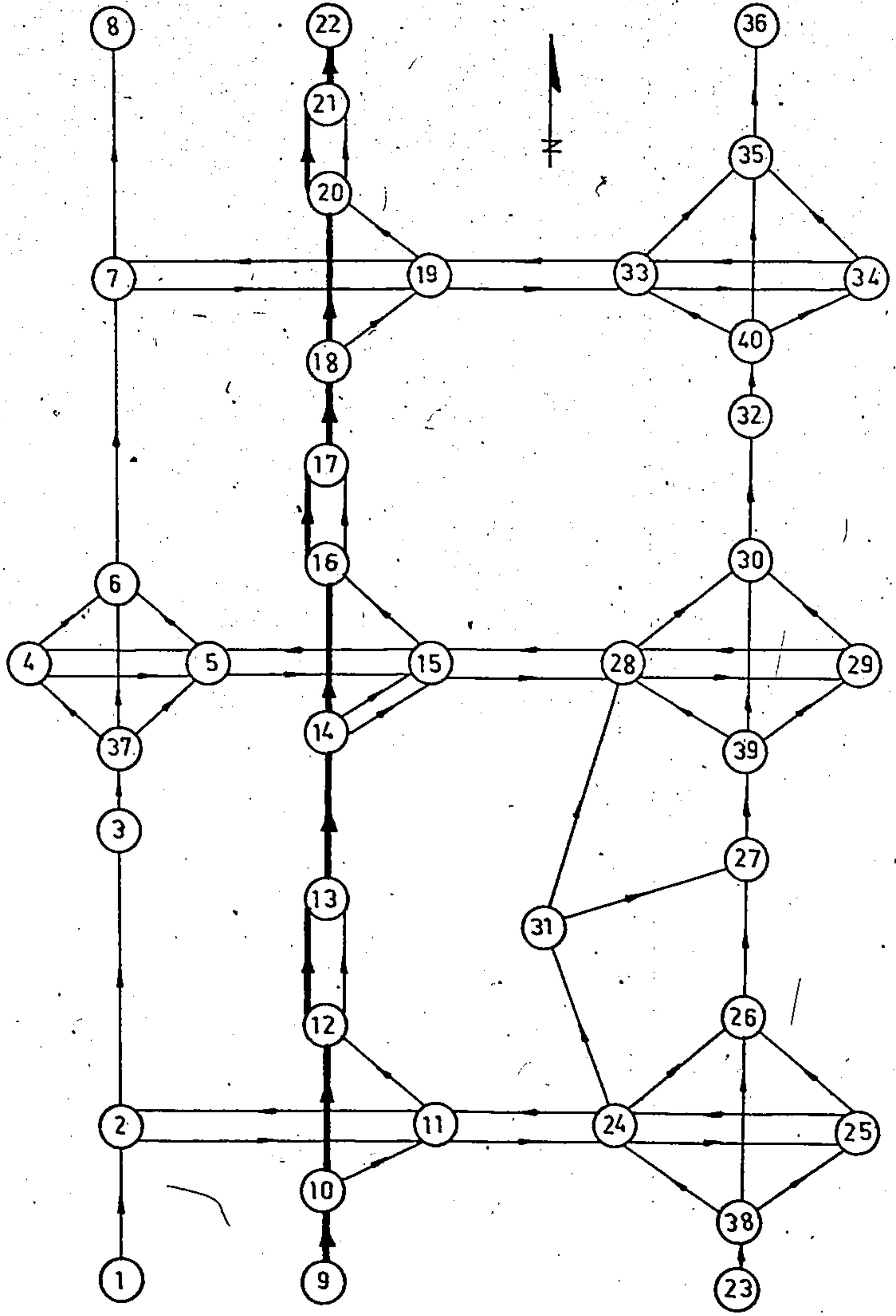


FIG. 3-2 LINK-NODE REPRESENTATION OF THE HYPOTHETICAL CORRIDOR

TABLE 3-1 DESCRIPTION OF NETWORK COMPONENTS

Network Component	Description of Links
a-Freeway links	9-10 13-14 17-18 21-22
b-Freeway merging links	10-12 14-16 18-20
c-Freeway dummy merging links	12-13 16-17 20-21 (left links)
d-Exit-ramps	10-11 18-19 14-15 (both)
e-Merging links for entrance-ramps	11-12 15-16 19-20
f-Merging dummy links for entrance-ramps	12-13 16-17 20-21 (right links)
g-Arterial links	1- 2 2- 3 6- 7 7- 8 26-27 30-32 35-36
h-Right turning links at intersections	37- 5 5- 6 38-25 25-26 39-29 29-30 40-34 34-35
i-Left turning links at intersections	37- 4 4- 6 38-24 24-26 39-28 28-30 40-33 33-35
j-Through flow links at intersections	37- 6 5- 4 4- 5 38-26 25-24 24-25 39-30 29-28 28-29 40-35 34-33 33-34
k-Dummy approach links at intersection	3-37 23-38 27-39 32-40
l-Connecting streets links	24-11 11-24 2-11 11- 2 5-15 15- 5 15-28 28-15 7-19 19- 7 19-33 33-19
m-Links connecting an O-D node to the network	24-31 31-27 31-28

Although the gross network features may be easily described using conventional link-node designation, more detailed characteristics such as inclusion of turning costs, the effect of turning movements on intersection approach capacity, freeway merging and weaving--etc., require special modelling with additional link-node representation. A description of the special modelling adopted in Fig. 3-2 is given below.

a) At-grade Intersections

Fig. 3-3 shows the modelling of an approach to an at-grade intersection. In this figure the movements at the intersection are represented by links 5-3, 5-4, and 5-2. A dummy link 1-5 having zero length is used to account for the mutual effect of the different movements on intersection capacity. The magnitudes of the individual movements are combined to form a weighted total of equivalent through flow which is assigned to link 1-5. In this way the total through equivalent flow can be limited by the capacity of link 1-5 which is represented in terms of through flow vehicles.

In obtaining the minimum paths, the cost of performing a certain movement at the intersection is composed of the cost of using the approach, which depends upon its degree of saturation, and the cost of this specific movement.

b) Merging Sections

The modelling of a merging section is illustrated in Fig. 3-4. Links 1-3 and 2-3 represent the merging

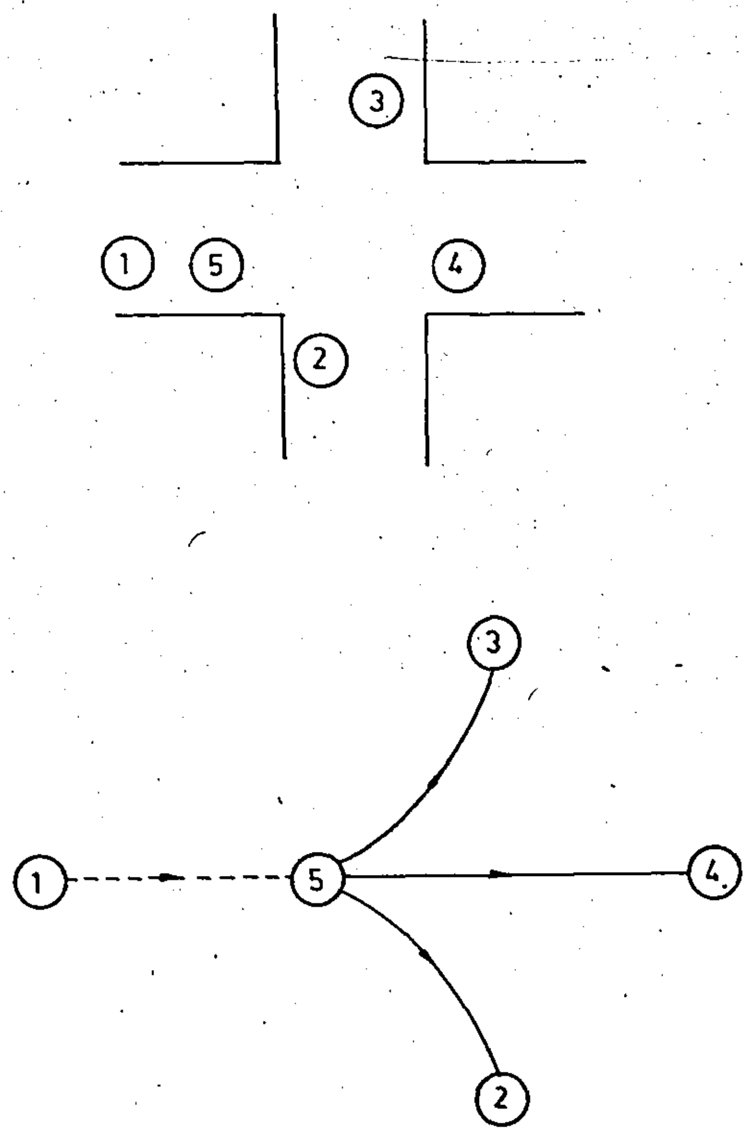


FIG. 3-3 LINK-NODE REPRESENTATION OF AN AT-GRADE INTERSECTION

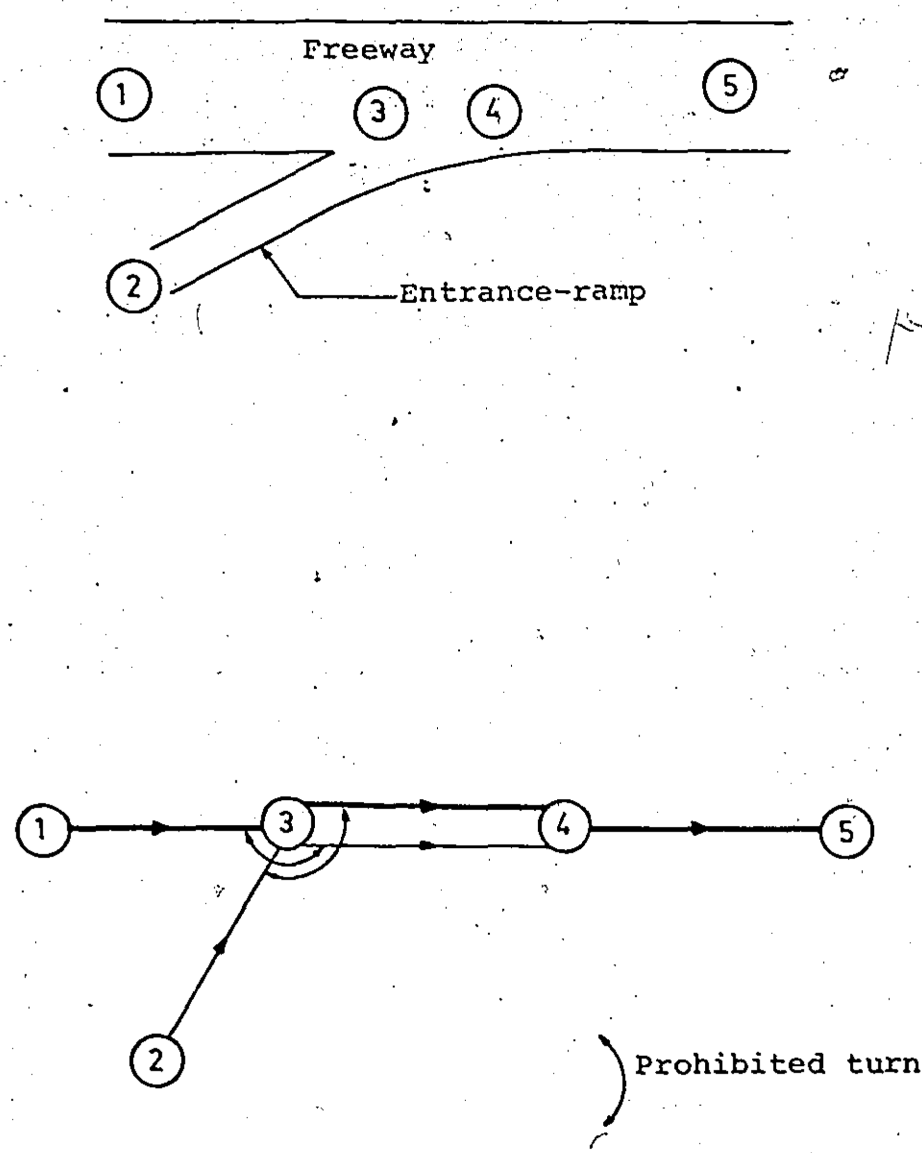


FIG. 3-4 LINK-NODE REPRESENTATION OF A FREEWAY MERGING SECTION

approaches of the freeway and the ramp respectively. Link 4-5 is the downstream merging link. The two dummy links 3-4 are used to represent arbitrarily short sections at the downstream ends of the merging approaches. The dummy links are given certain capacities to accept vehicles. These capacities are then used to regulate the ability to discharge vehicles onto the downstream merging link.

The representation shown in Fig. 3-4 requires that turn prohibitions be applied between links 1-3 and 3-4 (lower), and between links 2-3 and 3-4 (upper).

c) Weaving Sections

A weaving section can be modelled as shown in Fig. 3-5. The two links 4-5 represent the non-weaving and weaving sections. The upper link is used only by through flow. The entrance-ramp flow f_2 and exit-ramp flow f_4 use only the lower link. This link might also be used by through flow depending on the cost of the two links 4-5.

The above movements are performed through consideration of turn prohibitions between entrance-ramp dummy link 3-4 (lower) and non-weaving link 4-5 (upper) and between that non-weaving link and exit-ramp link 5-6.

Weaving section capacity and the effective number of lanes can be estimated using different methods [13]. The capacity of the non-weaving section can be estimated using Highway Capacity Manual [7].

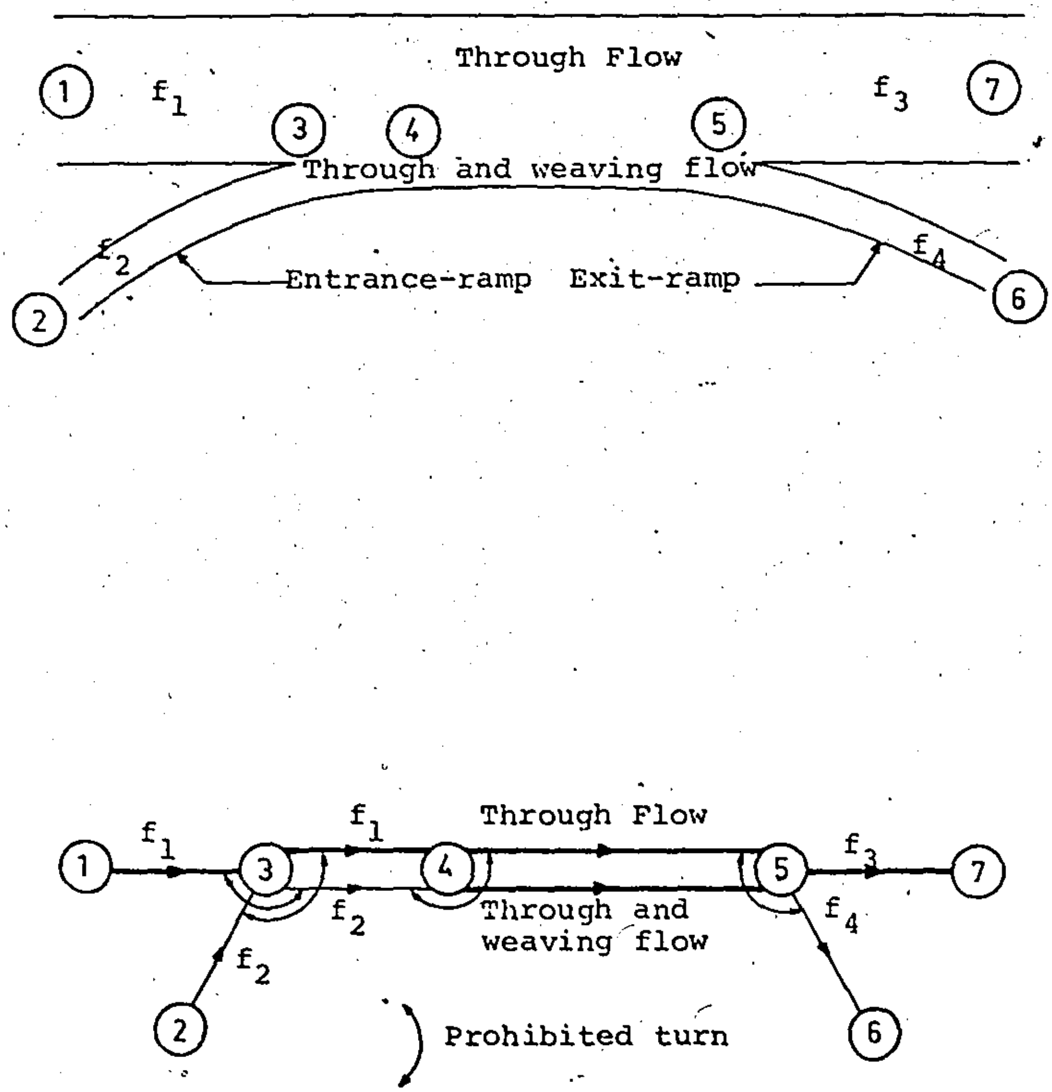


FIG. 3-5 LINK-NODE REPRESENTATION OF A WEAVING AND MERGING SECTION

d) Interchange Exit-Ramps

The upper portion of Fig. 3-6 shows an interchange having two exit ramps, 1 and 2. A simplified link-node representation of this interchange is shown in the lower portion of the figure. The two ramps are represented by two links having the same upstream and downstream nodes. The paths between links 1-2 (exit-ramp 2) and 2-4 and between links 1-2 (exit-ramp 1) and 2-3 are illogical and therefore turn prohibitions are to be applied for these movements.

In the above modelling the distance between exit-ramps is assumed negligible which is practical in most situations.

3.3 Assumptions and Approximations

Having described how to represent a corridor network, the assumptions adopted for the specific analysis of traffic assignment must be identified. The main assumptions and approximations used to treat queues, costs, intersection capacity, and other operational details are discussed in this section.

a) Queue Dissipation

Fig. 3-7 illustrates the approximation of queue dissipation as shown by the dotted line. A queue q_{n+j-1} which dissipates in time slice n+j is assumed to decrease at a constant rate during the whole length of slice. The error of this approximation is the area between the dotted line and the adjacent solid line which represents real queue behaviour. This error has an upper limit of

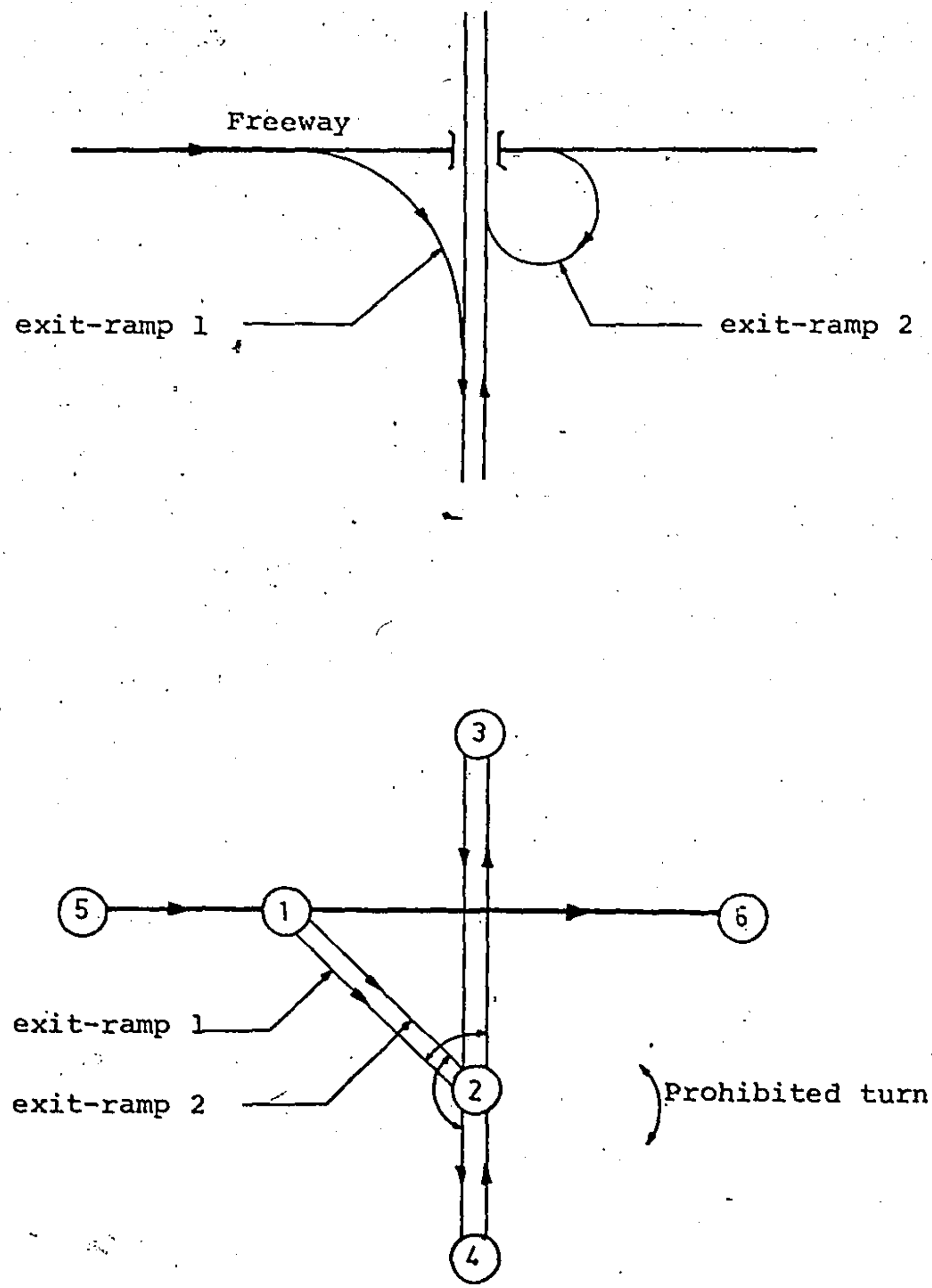


FIG. 3-6 SIMPLIFIED LINK-NODE REPRESENTATION OF INTERCHANGE EXIT-RAMPS

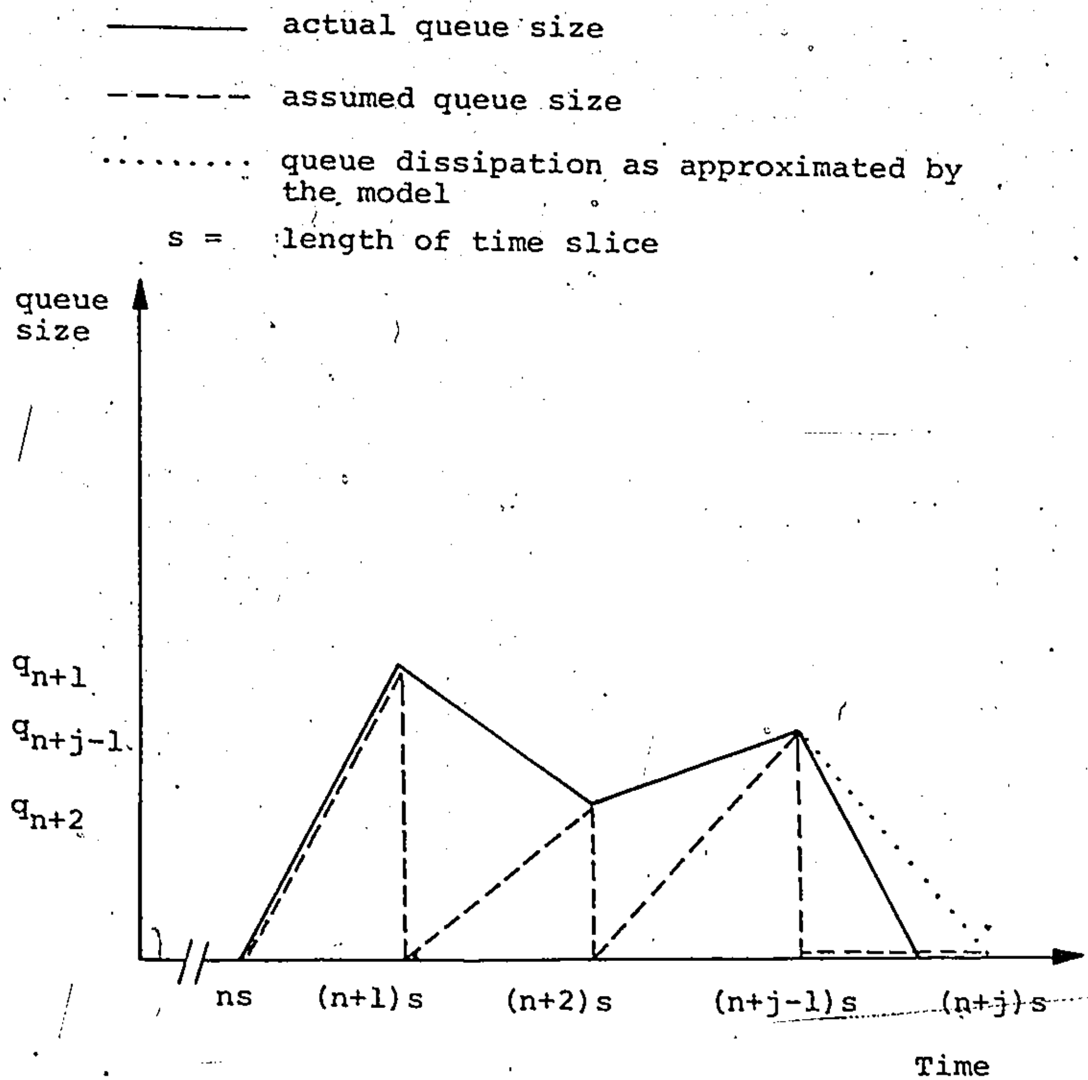


FIG. 3-7 EVOLUTION AND DISSIPATION OF A QUEUE

$\frac{s}{2} \cdot q_{n+j-1}$.

b) Queue Evolution

A queue formed in a given time slice is assumed to be fed back to the network in the next time slice. This is represented by the dashed lines in Fig. 3-7. By the assumption of queue dissipation during the whole length of the time slice n+j, the total queueing time obtained would be half its actual value assuming time slices of equal lengths. To account for that the calculated value of queueing cost is doubled.

c) Driver's Knowledge of Travel Costs

It is assumed in the model that the driver knows the travel and queueing costs for all the links in the network in each time slice. His choice of the shortest path is based on the travel cost information of that time slice. If the shortest path contains queues, he will go as far as the first congested point in the path and reselect the remainder of his path based on the new travel cost information of the next time slice.

d) Unlimited Queue Storage on Surface Streets

It is assumed that surface streets have sufficient space to store queues and therefore they do not spill back through major intersections. However, allowance is made for freeway and ramp queues to extend back onto freeway, ramp, and surface street links.

e) Constant Turning Equivalents

Each type of movement at an intersection is assumed to have a constant through flow equivalent independent of the number of vehicles making that movement. This assumption is essentially true for the small ranges of flows which might be expected in the peak periods. This means that the equivalent through flow F' is obtained as:

$$F' = F_L \cdot E_L + F_R \cdot E_R + F_T$$

where: F_L : Left turning flow

E_L : Through flow equivalent for left turns

F_R : Right turning flow

E_R : Through flow equivalent for right turns

F_T : Through flow

The flow F' is assigned to the dummy approach link of the intersection as described in section 3-2.

f) Flow-Cost Relationship

The function relating average unit travel cost to flow on each link is an increasing function as shown in the upper portion of Fig. 3-8. This relation is approximated by piecewise-constant components. According to this approximation, a link is replaced by three imaginary "sublinks" in parallel, each having a constant unit cost as illustrated in the lower portion of Fig. 3-8.

g) Unit Queue Cost

Fig. 3-9 shows a linear relation between queue size and unit queue cost. If a queue of size CSQ (capacity to serve queueing) can be served in a time slice of length s (hours), the queueing cost to be paid by a user at the

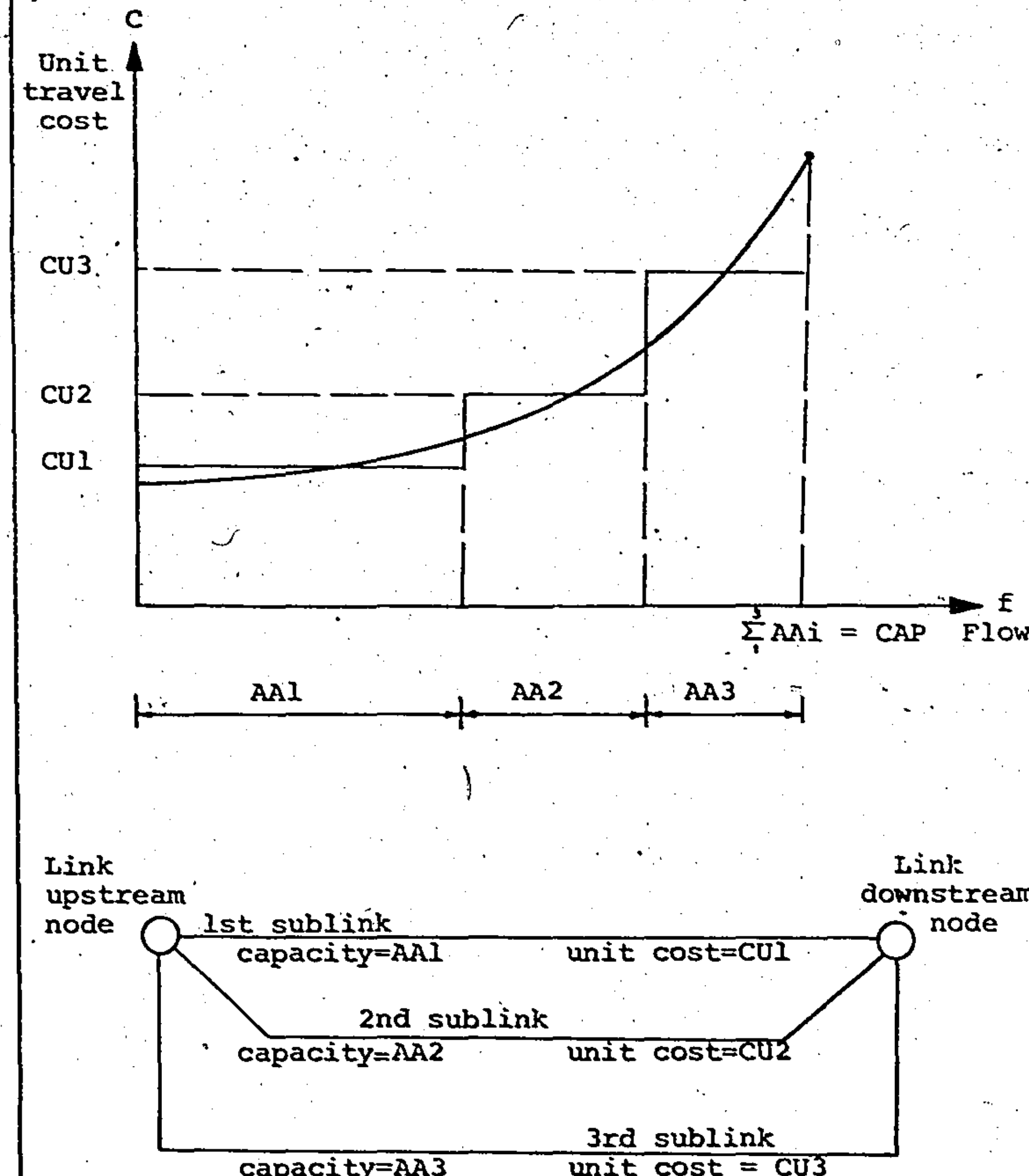


FIG. 3-8 PIECEWISE-LINEAR APPROXIMATION OF THE FLOW-COST RELATIONSHIP

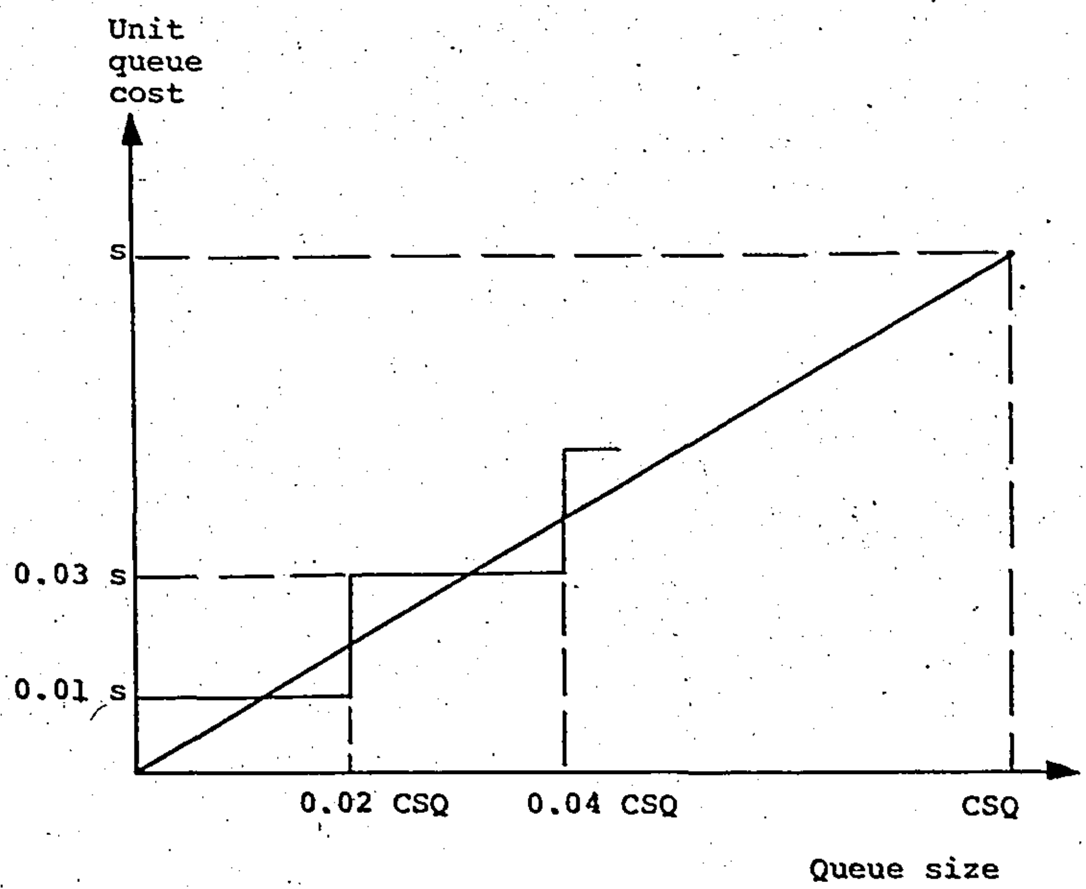


FIG. 3-9 PIECEWISE-CONSTANT COMPONENTS OF THE QUEUE-COST RELATIONSHIP

end of a queue of size q (vehicles) can be obtained as follows:

$$\text{Queueing cost (seconds)} = \frac{q.s}{CSQ} \times 3600$$

This relation is also approximated by piecewise-constant components having capacity limits of 0.02 CSQ and cost increments of 0.02 s. These capacity components are used so that the assignment to queueing links can be limited, and thus accounting for the queueing cost of assigned vehicles in a better way. In order to avoid excessive iterations, the unit cost is updated after each assignment increment and the allowable capacity component for use in the next increment is set equal to 0.02 CSQ.

3.4 Treatment of Flows, Queues, Costs and Capacities

In the previous sections the method adopted for link-node representation of a corridor and the basic assumptions considered in the assignment procedure were given. Also, the flow-cost and queue-cost relationships were established. Now, the mode in which the above aspects are incorporated within the model must be described. This section presents a detailed description of the model treatment of queue storage on the roadways, link costs, and weaving and merging capacities.

3.4.1 Storage of Queues

When a link reaches its capacity it forms a bottleneck and the vehicles wishing to use that link will form a queue starting from the upstream end of the link. The tail of the queue backs upstream as the number of queued vehicles increases.

If the queue backs through an intersection or a diverge section, it will affect the drivers who are not users of the bottleneck and might leave the queue upstream of the bottleneck.

The model takes into account the effect of any physical queue on traffic using upstream links by keeping track of the number of queued vehicles. If the physical queue capacity is reached, the upstream node of the queueing link is considered congested only for those upstream links that feed directly into the queueing link.

Any vehicles physically queued on a link are reassigned toward their destinations in the next time slice. The origin of this queue is approximated by the downstream node of the queueing link. However, the minimum paths for the queued demand on any link are obtained in such a way that this link is the first link considered in the minimum paths (the logic of this procedure is discussed in the next chapter).

The above approximation has the effect of not backing queues upstream if vehicles must queue on their final link. This occurs because the model considers them as having reached their destination. Therefore, neither the queueing time of the final link queue nor its probable effect on other queues will be considered.

Due to the manner in which the model assigns traffic, it is possible that some of the queued demand from a previous time slice may not be served in the current time slice. This can only occur if the first downstream link in the path reaches capacity or has a queue spilling back. The unassigned portion remains concentrated on the downstream node of the queueing link, but is not included in the physical queue of that link.

3.4.2 Representation of Link Costs

The total unit cost on a link consists of two components:

a) Unit flow cost - that cost which must be paid for travel on the non-queueing portion of the link.

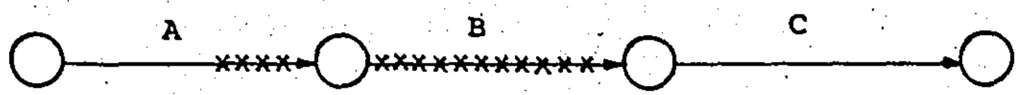
b) Unit queue cost - that cost which must be paid for travel on the queueing portion of the link.

The unit flow cost is determined from the flow-cost relationship in Fig. 3-8 and represents the unit cost for the whole length of the link. If part of the link contains a queue, the unit cost should be factored down to represent only the unit cost for the non-queueing portion of the link.

As described in section 3-3 the unit queue cost is calculated as $\frac{q.s}{CSQ} \times 3600$. The discharge rate CSQ of the link, is estimated in the model according to existing network and traffic conditions. It is always proportional to the saturation flow of the links serving the queue.

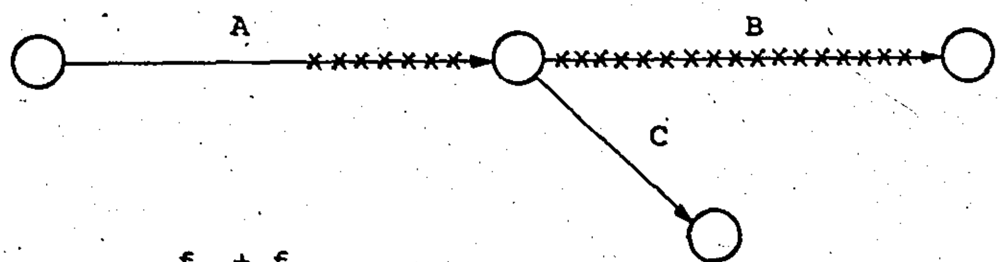
For the queueing example in Fig. 3-10, the discharge rate of link B, CSQ_B, is equal to the saturation flow capacity of the bottleneck C, CAP_C. Consequently, CSQ_A is equal to CSQ_B and not the capacity of link B.

Fig. 3-11 shows a queue built on the upstream link A of a diverge. This would occur if either link B or C reached capacity, or has a queue which spilled back onto link A. In



$$CSQ_A = CSQ_B = CAP_C$$

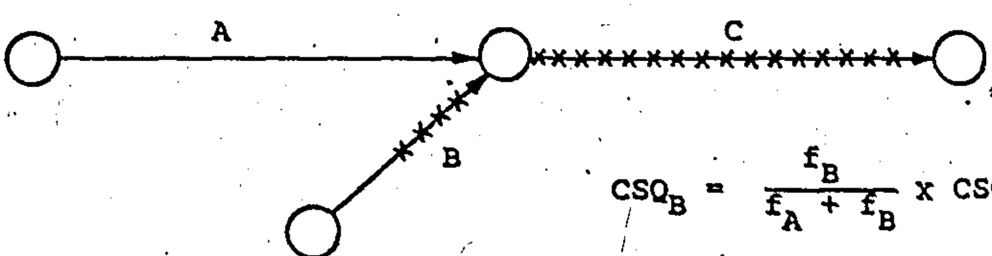
FIG. 3-10 QUEUE SPILL ONTO LINK A FROM LINK B



$$CSQ_A = \frac{f_B + f_C}{f_B} \times CSQ_B$$

$$\ngtr CAP_A$$

FIG. 3-11 QUEUE FORMED ON A DIVERGE SECTION



$$CSQ_B = \frac{f_B}{f_A + f_B} \times CSQ_C$$

$$\ngtr CAP_B$$

FIG. 3-12 QUEUE FORMED ON A MERGE SECTION the latter case where the queue on B spills onto link A, an estimate of the relative fractions of the queue components on A can be computed as $\frac{f_B}{f_B + f_C}$ and $\frac{f_C}{f_B + f_C}$ where f_B and f_C are the assigned flows on the respective links. Link B would accept vehicles at rate of CSQ_B and therefore the queue on A would discharge at approximately the rate $\frac{f_B + f_C}{f_B} \times CSQ_B \leq CAP_A$. Similarly, if link C has a queue, the discharge rate from A will be $\frac{f_B + f_C}{f_C} \times CSQ_C \leq CAP_A$.

In the case of a queue forming at a merge section as shown in Fig. 3-12, CSQ_B can be approximated according to the flows on the merging links. If a queue forms on link B due to a queue on C, CSQ_B is estimated as $\frac{f_B}{f_A + f_B} \times CSQ_C \leq CAP_B$.

In the case where the queue on B is formed due to link C reaching capacity, CSQ_C in the above expression is replaced by CAP_C.

3.4.3 Weaving Section Capacity

If one wishes to account for weaving in the assignment procedure, a dynamic method would be required to estimate the final weaving capacities on the basis of assigned flows. Since the effects of weaving are generally unknown, an appropriate method of calculating weaving capacities does not exist. Therefore, inclusion of such a procedure in this model has not been attempted and is left for future research. However, there is allowance in the model for providing pre-estimates of weaving capacities in each time slice.

3.4.4 Merging Section Capacity

Capacities of merge approaches are affected by variation in the capacity of the downstream merging link, e.g. due to weaving. In addition, these capacities are dependent on one another's flow and are complementary. If both approaches have demands greater than their capacities to discharge vehicles, queues will be formed upstream of each merge. In this case, the capacity to discharge vehicles for each approach is defined as the "entitlement", since it is always available for the approach. If one approach does not use all of its entitlement, the excess can be used, as required, by the other.

For the cases where the total merge capacity is independent of flow composition in merge approaches, the relationship between complementary capacities of merge approaches is shown in Fig. 3-13 and must satisfy $CM_1 + CM_2 = C$, where C is the total merge capacity and CM_1 and CM_2 are the respective merging capacities of approaches 1 and 2. The point (E_1, E_2) represents the mutual stable capacities where each approach receives its entitlement. While the entitlement of each approach represents a lower bound of its capacity, an upper bound of this capacity is defined by the approach ability to accept vehicles at its upstream end. For example, the point C_1 represents the capacity to accept vehicles for approach 1 and hence its maximum capacity is limited in this value. A set of demands (D_1, D_3) which falls below the line of total

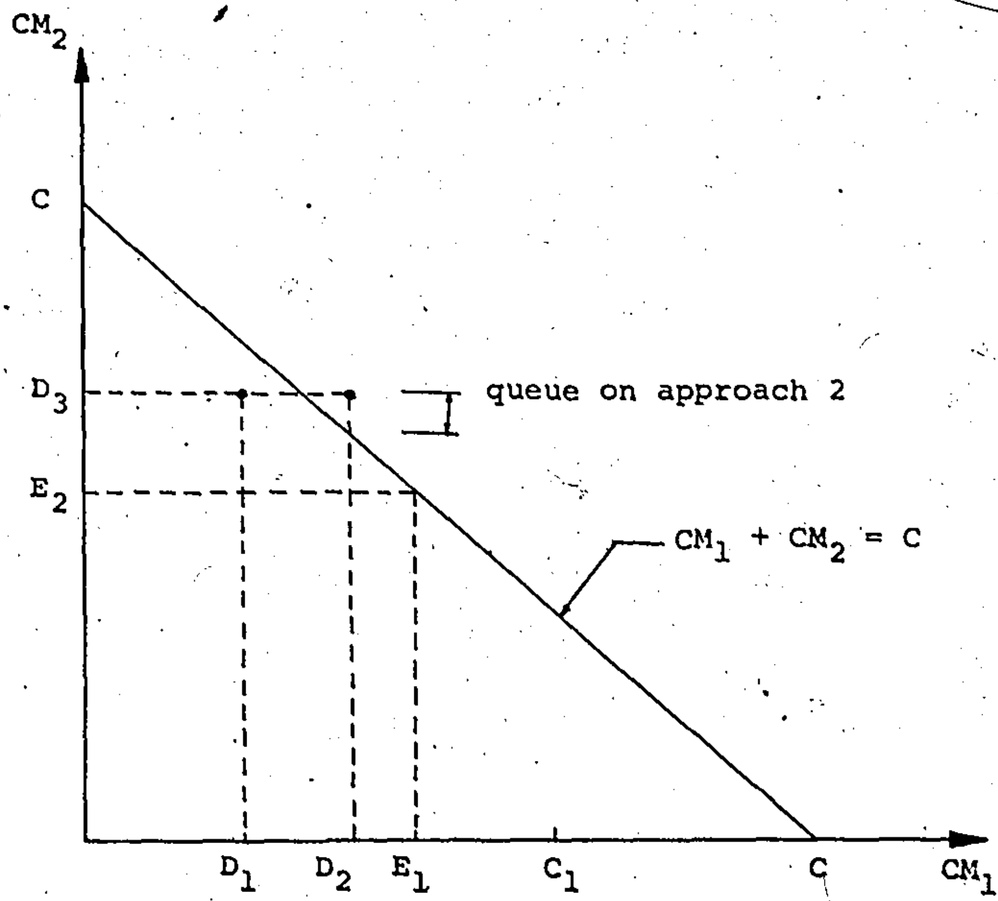


FIG. 3-13 RELATIONSHIP BETWEEN CAPACITIES OF MERGE APPROACHES

merge capacity results in each approach receiving sufficient capacity because the unused entitlement of $E_1 - D_1$ for approach 1 is large enough to serve the excess demand of $D_3 - E_2$ on approach 2. However, if the total demand exceeds the total merge capacity C, a queue will be formed. Considering the demand set (D_2, D_3), the excess demand of approach 2, $D_3 - E_2$, is greater than the unused entitlement, $E_1 - D_2$ of approach 1. In this case, approach 2 can use only $E_1 - D_2$, and the remainder $(D_3 - E_2) - (E_1 - D_2) = D_3 + D_2 - C$ will form a queue on that approach. It should be noted that although approach 2 experiences the same demand for both cases, a queue is formed only under the latter conditions.

The approach capacities can be estimated from entitlements and flows using the procedure described above.[6] However, the approach flows cannot be assigned until the capacities are known. In order to solve this problem, the following two routines are used:

a) Capacity-borrowing Routine

This routine is used after each increment in the assignment. If one approach reaches its capacity while the other has an excess, a certain amount of the excess is reserved, and some predetermined fraction of the remainder (if any) is given to the former approach. The reserve is calculated to be of sufficient quantity to complete the time slice at the present rate of assignment, plus an additional amount which accounts for possible approach queues of the previous iteration.

Although the above procedure would ideally account for the sharing of merge capacity, some problems exist in using it. The flows are not assigned at a constant rate, and therefore the extrapolation of assigned flows might be poor estimates of those existing at the end of the iteration. Another difficulty is that if one approach link becomes capacitated during any increment of the iteration, the borrowed capacity will not be used and queues might form despite the availability of that capacity. This difficulty is overcome using the following routine.

b) Entitlement-Updating Routine

The entitlement-updating routine is provided for optional use in the program. It is used after each iteration to calculate new entitlements for merge approaches on the basis of approach demands assigned in the previous iteration. If both the demands are greater or less than the respective entitlements, no borrowing will occur. If one approach has a demand greater than its entitlement and the other's demand is below its entitlement, two possibilities exist. If the required capacity of one approach is greater than the excess capacity of the other, the entire excess is transferred. If it is less, the excess is shared to give both approaches equal excesses.

If merge capacity is left unused and queues exist in spite of the entitlement-updating routine, a further step is required. From the assigned flows and queues, the demands of the merge approaches are determined and new entitlements are calculated. Using these entitlements, a new run is performed without considering either of the two routines. Furthermore, if the analyst can determine merge demands sufficiently well without the routines, this final step can be used directly.

CHAPTER 4

SPECIAL FEATURES

4.1 Introduction

In the previous chapter, the basic aspects required for the assignment of traffic to a network corridor were discussed. The purpose of this chapter is to describe some special features contained in the assignment procedure and how this procedure is actually performed in the model. Finally, the fields to which the model might be usefully applied are investigated.

One concern involves consideration of traffic diversion due to the presence of queues on minimum paths. The assignment method of Homburger [8] assigns traffic to links which are not saturated. Any link reaching capacity is omitted from the minimum path and assigned no more flow. Yagar's method leaves any capacitated link in the network and accounts for queueing costs on the upstream links of that link. Therefore, the minimum paths may contain some cost in queues. It is felt that this queueing minimum path might not be considered the best path for all network drivers because of the queueing situation, and some of them may divert their trip to alternate non-queueing path. Such a process must obviously include a procedure to divert the trips from the queueing path to the non-queueing one. Such a traffic diversion procedure is described in this chapter as an attempt to account for the impact of queueing costs on minimum paths.

Another important aspect incorporates the possibility of illogical paths which might occur at various sections within the network when choosing a minimum path. The form of these illogical paths is investigated in this chapter, along with a suggested method for identifying them automatically in the model.

As an attempt to reduce network coding effort, a method is proposed for computing turning volumes without the need for turning links in the network. Possible uses of this method are discussed.

4.2 Traffic Diversion

For each increment in the assignment, minimum paths are obtained. The minimum path may contain some links with queues. In this case, it would seem reasonable that a certain percentage of demand would divert from such a queueing minimum path to another path having no queues, even if the latter one had a greater travel cost.

To establish the logic of this procedure consider the following definitions:

a) A first minimum path

It is the minimum path obtained considering all links in the network, including those links with queues.

b) A second minimum path

It is the minimum path obtained considering only those links without queues.

Those definitions imply that traffic diversion will occur if and only if both of the following conditions are satisfied:

a) Presence of queues in the first minimum path

b) Availability of a second minimum path

If these two conditions are fulfilled, a percentage P of the remaining demand will divert from the first minimum path to the second minimum path. Therefore, the first minimum path will receive only the portion (1-P). If the second minimum path had the same trip cost, the entire demand will be assigned to it.

The percentage of vehicles diverted to the second minimum path is obviously affected by how drivers evaluate one unit cost of queueing in terms of non-queueing cost. Now assume that we have a first minimum path with a queueing cost of Q units as shown in Fig. 4-1. The first minimum path has a trip cost of T_1 units while the trip cost of the second minimum path is T_2. If a percentage P of drivers diverts from the first minimum path they will do so if their evaluation of one unit of queueing cost is V or more units of non-queueing cost. The minimum equivalent queueing cost V required for diversion is defined as follows:

$$V = \frac{T_2 - (T_1 - Q)}{Q}$$

Therefore

$$V = 1 + \frac{\Delta T}{Q}$$

where ΔT is the difference between the total travel cost of the second and first minimum paths respectively.

$$V = 1 + \frac{\Delta T}{Q}, \quad \Delta T = T_2 - T_1$$

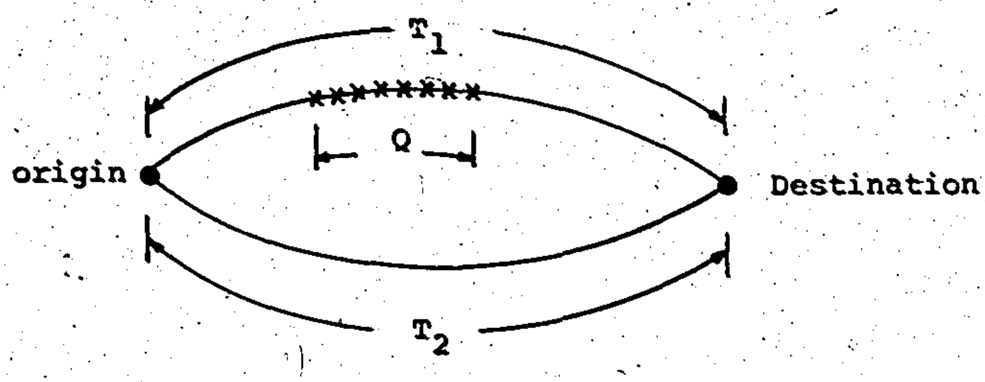


FIG. 4-1 DERIVATION OF MINIMUM EQUIVALENT QUEUEING COST V

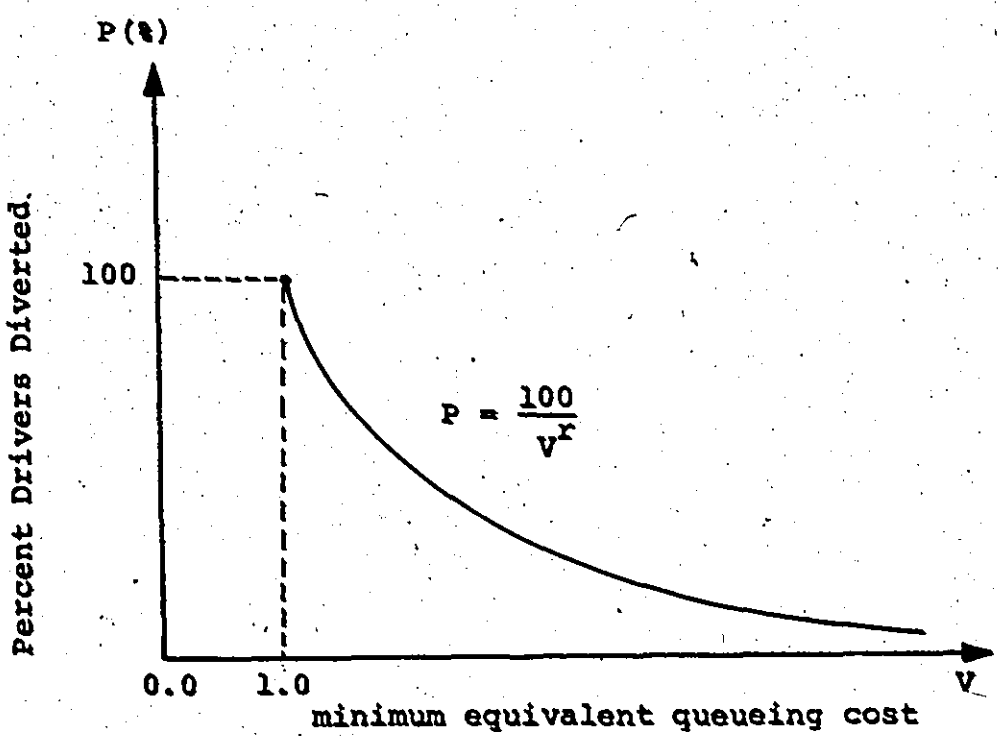


FIG. 4-2 THE DIVERSION RELATIONSHIP

The value of V gives an equivalent non-queueing cost which results in a trip cost for the first minimum path equal to the trip cost of the second minimum path. Therefore, any driver having an equivalent queueing cost of V or more will divert to the second minimum path.

To establish the relation between P and V, at the limits we have:

$$P = 100 \text{ for } V = 1.0 \quad (Q = \infty \text{ or } \Delta T = 0)$$

$$P = 0 \text{ for } V = \infty \quad (Q = 0 \text{ or } \Delta T = \infty)$$

One diversion relationship that could satisfy these constraints is:

$$p = \frac{100}{V^r}$$

Where r is a constant controlling the diversion case.

The extreme values of $r = \infty$ and $r = 0$ are equivalent to the no diversion and total diversion cases respectively, while $0 < r < \infty$ results in partial diversion. The above relationship is illustrated in Fig. 4-2.

For a given network, the value of r might differ from one path to another according to trip purpose, trip length, time of day, driver preferences, etc. For such cases a different diversion curve would be required at each queueing section, e.g. entrance ramp. To establish these diversion curves a field survey is necessary. However, for purposes of this development, the value of r is assumed constant for the entire network.

In the model calibration, one possible method to determine the value of r for a given network, would be to choose that value which provides the minimum Root Mean Square (RMS) error of the predicted and observed queues lengths, where RMS is defined as:

$$RMS = \sqrt{\frac{\sum_{1}^{n} (y_i - x_i)^2}{n}}$$

where n = number of queueing links considered in the calibration

y_i = predicted queue length on each queueing link

x_i = observed queue length on each queueing link

Having been calibrated, the model can be applied to predict flows and queues that would result from any implementation of a control strategy such as ramp metering.

4.3 Turn Prohibitions

One can consider turn prohibitions as those imposed for control means, or those imposed to prevent illogical paths in the network. The former can be used at intersections where left and/or right turns are to be prohibited. This type of control is necessary if right turning vehicles interfere excessively with heavy pedestrian traffic or if left turning vehicles interfere with heavy opposing traffic. It also might be essential when the turning vehicles create congestion problems on the downstream links of those turns.

On the other hand, turn prohibitions are necessary to avoid illogical paths in certain parts of the network. For example, consider an interchange configuration as shown in Fig. 4-3(a). Movements are not possible from the exit-

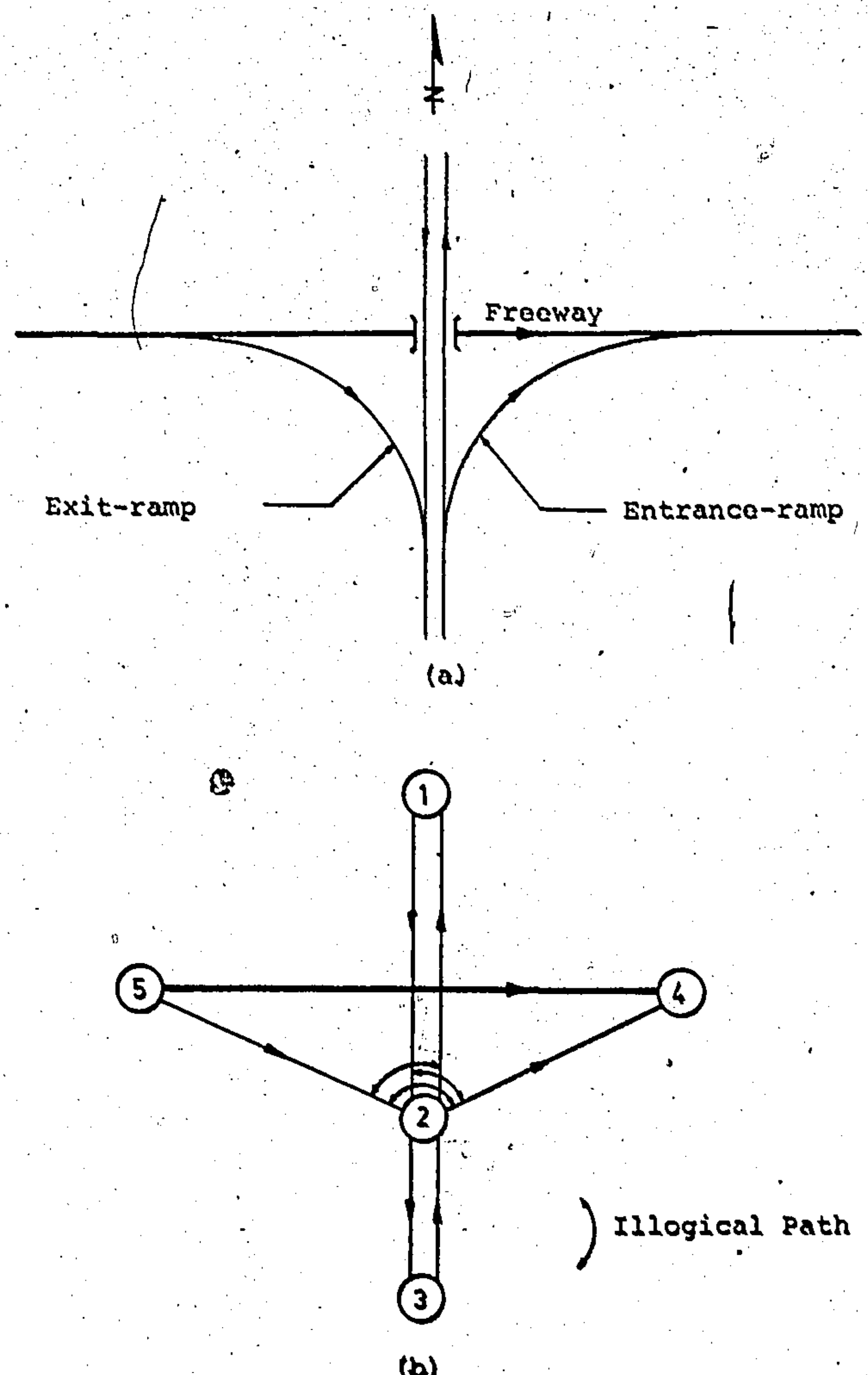


FIG. 4-3 ILLOGICAL PATHS AT AN INTERCHANGE

ramp to the northbound direction, or from the southbound direction to the entrance-ramp. Therefore, in the link-node representation shown in Fig. 4-3(b) the paths between links 5-2 and 2-1 and between links 1-2 and 2-4 are illogical. Of course, the path between links 5-2 and 2-4 is also illogical.

Fig. 4-4 illustrates another example of illogical paths which might occur at intersections. Link 1-3 is provided for right turns, link 1-4 for left turns, and link 1-2 is a through flow link. Link 3-4 is a through flow link for another approach. If the cost of link 1-3 plus the cost of link 3-4 is less than the cost of link 1-4, the minimum path will be 1-3-4 instead of 1-4. Similarly, the same concept can be applied for other movements, and five other illogical paths can be found.

Turn prohibitions are also required for avoidance of illogical paths in link-node representation adopted for sections of merging, weaving, and interchange exit-ramps as previously discussed in section 3-2.

Additionally, illogical paths might occur when any node is an O-D node. In Fig. 4-5, node 7 has exogenous flows entering via nodes 1, 2, and 3, and leaving via nodes 4, 5, and 6. As the exogenous links from nodes 1 to 6 are joined to the single node 7, illogical paths might occur if these links were used in the minimum paths for through flows.

A complete description of the method adopted for automatically determined turn prohibitions for network control and illogical paths is presented in Appendix A. In an

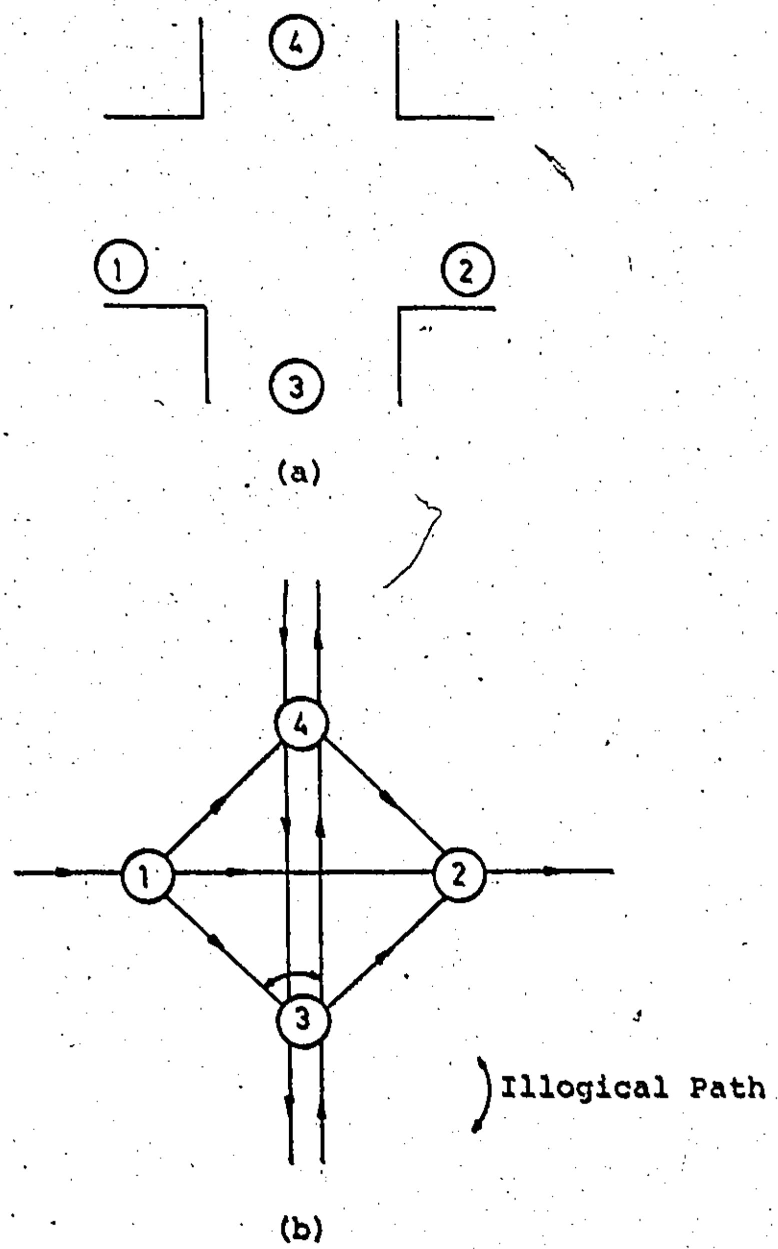


FIG. 4-4 EXAMPLE OF ILLOGICAL PATHS AT AN INTERSECTION

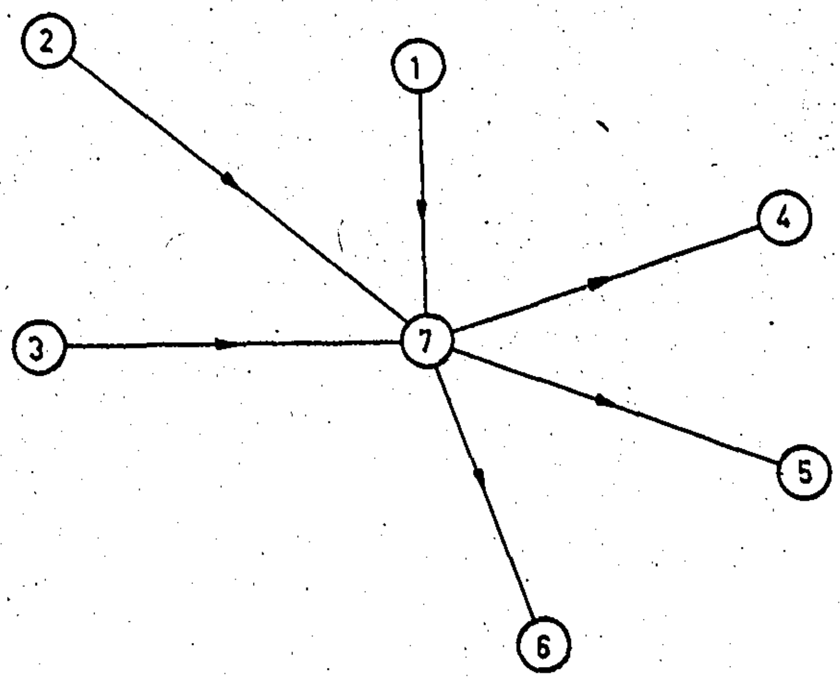


FIG. 4-5 ILLOGICAL PATHS AT AN ORIGIN-DESTINATION NODE

attempt to minimize the computer time, illogical paths through O-D nodes are automatically determined using another method. This method is also given in Appendix A.

The illogical paths described above are those which might occur when obtaining minimum paths in each time slice. These illogical paths are avoided by the developed minimum path algorithm with turn prohibitions. However, there is still another important type of illogical paths which requires special treatment. This type might occur for any queued demand at an intermediate node between the old path of previous time slice, and the new path of present time slice. Fig. 4-6 illustrates this type of illogical path. Suppose that in a given time slice, demand is assigned from origin node 5 to destination node 4 on the minimum path 5-6-2-3-4. Since link 2-3 reaches capacity, this demand queues on link 6-2. Assume that the queued demand is concentrated on downstream node 2 for the purpose of assignment in the following time slice, where node 2 is considered as an origin. If it happens in the following time slice that the cost along the remainder of the old path is increased such that the total cost along links 2-6, 6-7, 7-3 and 3-4 is less than the total cost of links 2-3 and 3-4, the minimum path will be 2-6-7-3-4. This means an illogical path would occur between the old path and the new path at node 2, through execution of a U-turn. The same concept can be applied if demand queued at a turn prohibition node.

To solve this problem, it is necessary to overlap the

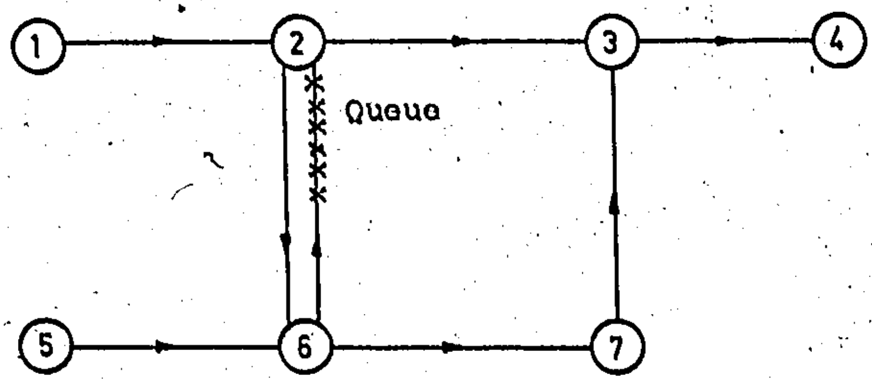


FIG. 4-6 EXAMPLE OF TIME SLICES-RELATED ILLOGICAL PATHS

old and new minimum paths of any queued demand. In other words if the last link of the old minimum path (that link on which queue is formed) is considered as the first link in obtaining the new minimum path, such illogical paths will not occur. Therefore, the procedure adopted for solving this problem considers the origin as the upstream node of the link on which the queue exists. When minimum paths are obtained from that origin, the procedure is performed as if that link were the only link out of that node. In addition, when demand is assigned from that origin, the first link receives no assignment. As an example, in Fig. 4-6 node 6 is considered as an origin with only link 6-2 out of that node and the flow is not assigned to that link. If link 6-7 has another queued demand, node 6 is considered as an origin for that queued demand, but with only link 6-7 out of that origin. In this way the illogical paths are avoided with minor increase in the computer running time due to the increase in the number of origins considered.

4.4 Turning Volume Calculation

As described in the previous chapter, turning links are used at major intersections to account for turning costs. Obviously, the turning volumes can be obtained directly as the volume on those links. However, if one wishes to determine the turning volumes at other intersections which do not have turning links, a different method is required. For example, at a simple diamond interchange information on turning volumes might be useful at the intersection of its ramps and surface streets before the implementation of any intersection control was attempted.

Consequently, a method was developed to compute turning volume without using turning links. This method is described in detail in Appendix B, with an illustrative numerical example.

In order to make it easy for the user, turning volumes are output in a way which saves time, effort, and confusion. Fig. 4-7 shows a sample of that output for a typical four-leg intersection.

For further reduction in the link-node representation of this model, this method can be used in conjunction with the developed minimum path algorithm with turn prohibitions to account for turning costs at major intersections without using turning links. However, this has not been done and is left for further research.

4.5 Basic Logical Sequences of the Model

In the previous chapter the different aspects involved in assigning time-varying demands to a corridor were explained. In addition, special activities of the model have been described in the previous sections of this chapter. With this basic information in mind, it is now possible to identify the logical sequences of the model when performing these aspects and activities. A flow chart showing these logical sequences is given in Fig. 4-8.

It can be seen that network turn prohibitions and illogical paths through O-D nodes are determined once at the beginning

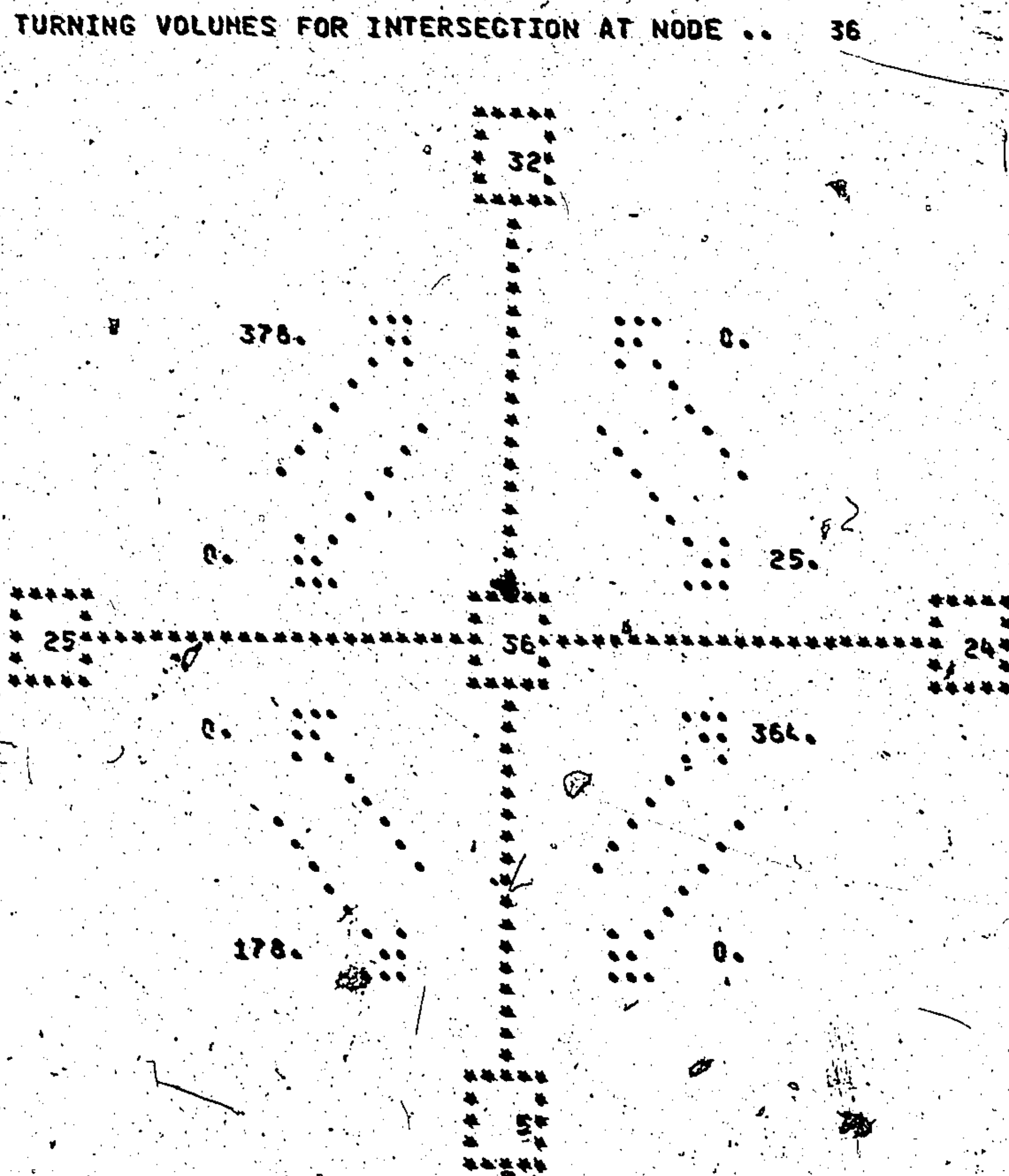


FIG. 4-7 SAMPLE OF COMPUTER OUTPUT OF TURNING VOLUMES

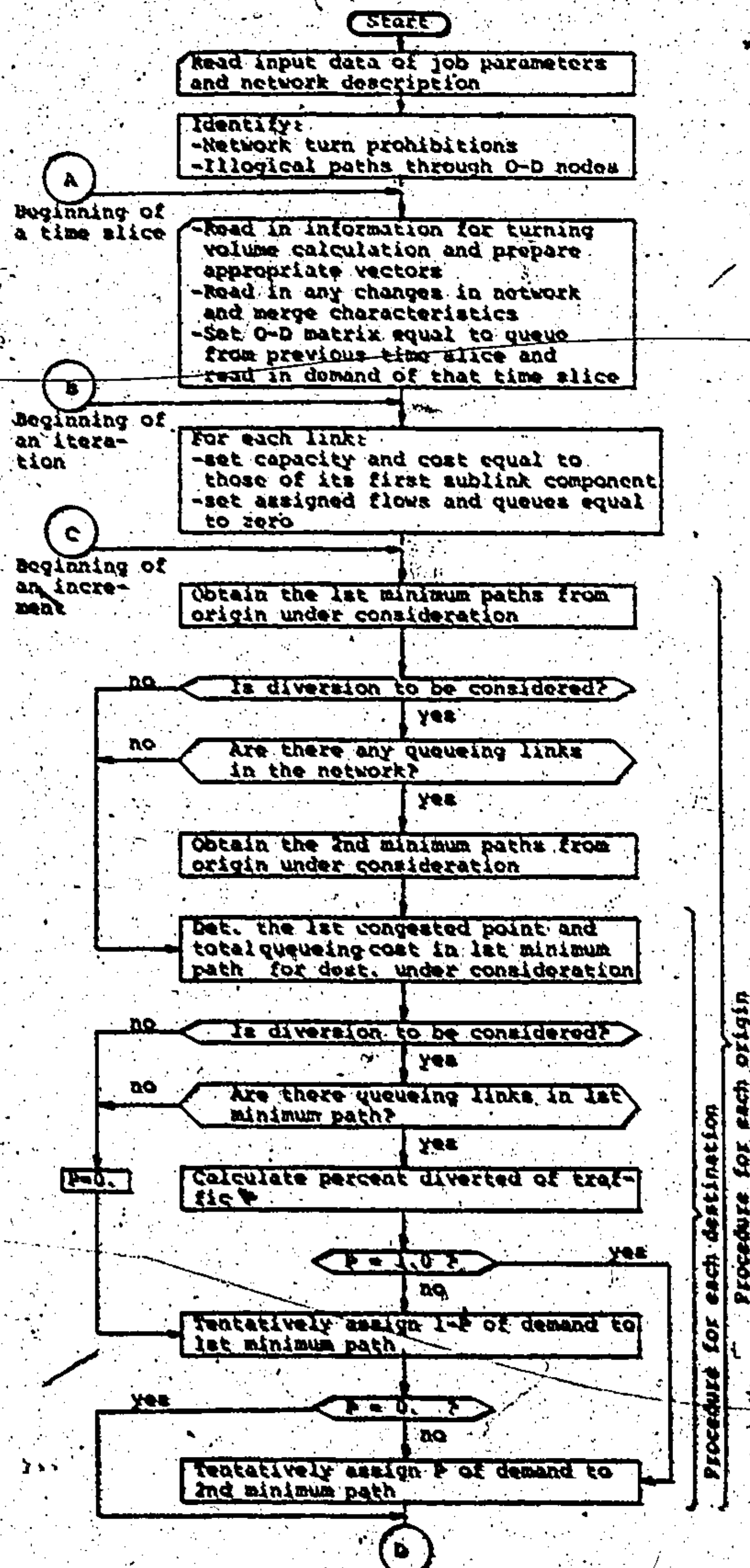


FIG. 4-8 BASIC LOGICAL SEQUENCES OF THE MODEL

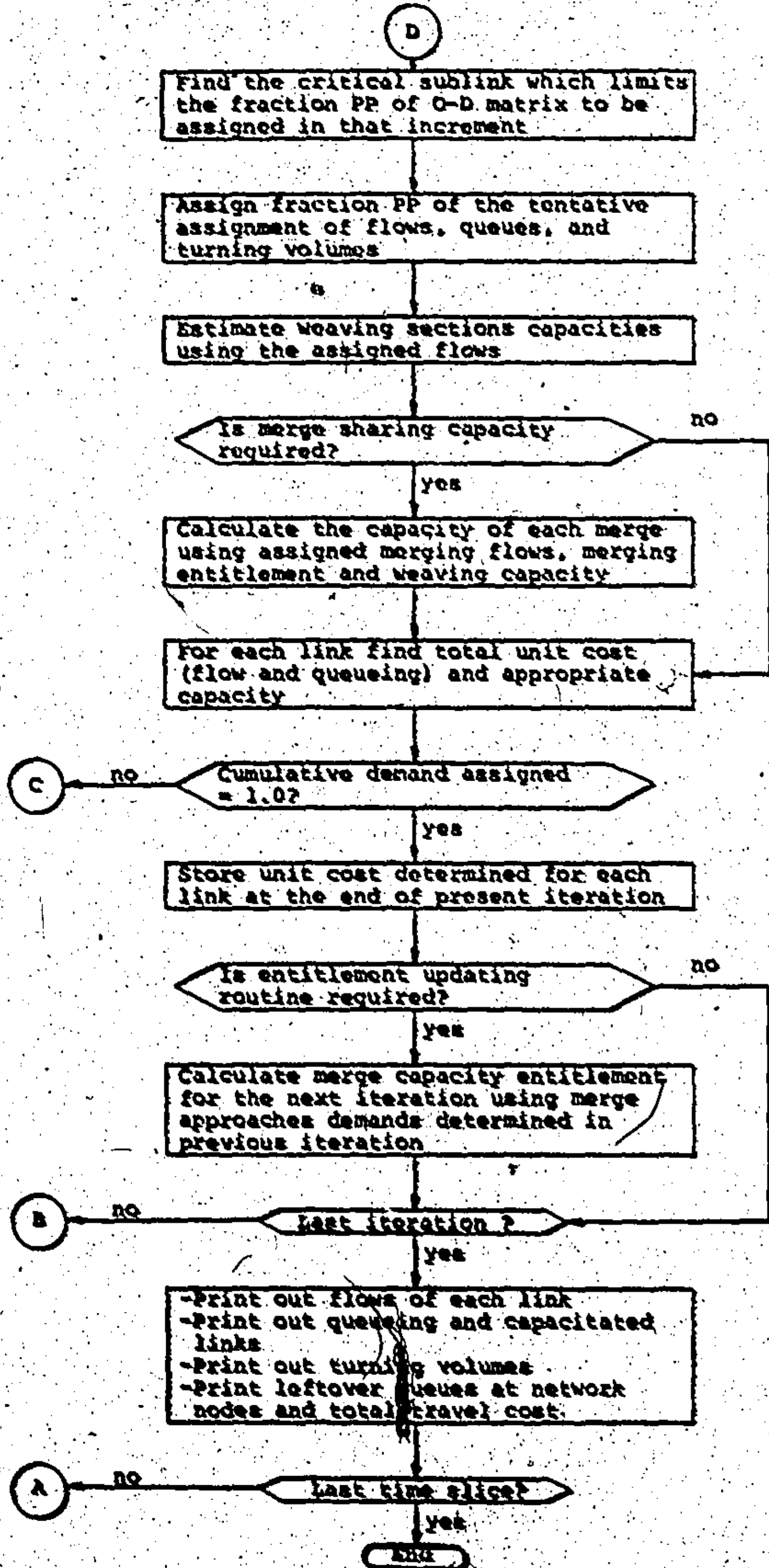

D
Find the critical sublink which limits the fraction PP of O-D matrix to be assigned in that increment
Assign fraction PP of the tentative assignment of flows, queues, and turning volumes
Estimate weaving sections capacities using the assigned flows
Is merge sharing capacity required?
no
yes
Calculate the capacity of each merge using assigned merging flows, merging entitlement and weaving capacity
For each link find total unit cost (flow and queueing) and appropriate capacity
C
no
Cumulative demand assigned = 1.0?
yes
Store unit cost determined for each link at the end of present iteration
Is entitlement updating routine required?
no
yes
Calculate merge capacity entitlement for the next iteration using merge approaches demands determined in previous iteration
B
no
Last iteration ?
yes
-Print out flows of each link
-Print out queueing and capacitated links
-Print out turning volumes
-Print leftover queues at network nodes and total travel cost.
A
no
Last time slice?
yes

FIG. 4-8 (CONT.)

of the programs. Since turn prohibitions include both those for network control and for illogical paths, network control such as no left turn will be applied for all time slices. (Slight modifications in the computer program would be required if such control was to be applied only for certain time slices).

Traffic diversion is provided as an option in the model. This has been done to enable the analyst to compare the results in both cases. Also, the no diversion case is equivalent to using a value of infinity for the constant r in the diversion model, i.e. all drivers evaluate queueing time exactly as non-queueing time.

Traffic is assigned to the first minimum path until it reaches the first congested point. At this point it joins the queue waiting upstream, and will be served in the next time slice. The first congested point is approximated by the downstream node of the link on which queue is formed. The stored queue is considered as a demand in the next time slice. That queued demand is assigned starting from the downstream node, although the minimum path is obtained from the upstream node in an attempt to avoid illogical paths. On the other hand, flows are assigned to all links of the second minimum path, since there is no congestion on this path.

After each iteration, the unit costs of links are stored to be used in the following iteration. These could be weighed in with the costs corresponding to partially assigned flows during that iteration. In addition, it is also possible to weigh the travel costs of the previous time slice into the first iteration of the present time slice when there is not significant variation in demands of the different time slices.

4.6 Fields of Application

Before application, the model must be calibrated to determine the best value of the constant r in the diversion model as described in section 4-2. It could then be applied for testing and predicting the effect of control strategies related to ramp-metering, either fixed rate or traffic-responsive.

In fixed-rate metering a given rate of vehicles is allowed onto the merging section. This rate is set for the ramp entitlement capacity with no allowance for sharing of merge capacity. When the total merge capacity is constant freeway merging capacity will also be constant and equal to the total merge capacity minus the metering rate assuming it is fully utilized. These fixed metering rates can be determined by employing the capacity-borrowing routine which will give each approach its desired entitlement, but transfer the excess of one approach to the other approach, if needed. Allowance is provided in the model for altering the fixed metering rate for each ramp at the beginning of each time slice.

In a traffic-responsive metering scheme, the ramp flow rate is limited as a function of the freeway merging flow. The freeway approach is given first a complete merging priority and then the remaining merging capacity is given to the entrance-ramp vehicles. This can be modelled by giving all the merging capacity to the freeway and none to the ramp, and thus allowing the freeway to share whatever excess it had. However, in both types of metering the capacity-borrowing routine can give misleading results due to difficiencies discussed in section 3-4.

In both cases, the model can not be used to give the optimal metering rates directly, but rather it can be used for testing different control metering strategies and choosing the best strategy to be employed. In such a case the model can accurately predict queue lengths on, and upstream of, the metered ramps, where it accounts for traffic diversion from ramp queues to available alternative routes. With diversion consideration, the investigator could draw subjective conclusions on the availability of queue-free routes and any network controls that must be considered prior to the employment of a certain control metering strategy.

CHAPTER 5

APPLICATION OF THE MODEL

5.1 Introduction

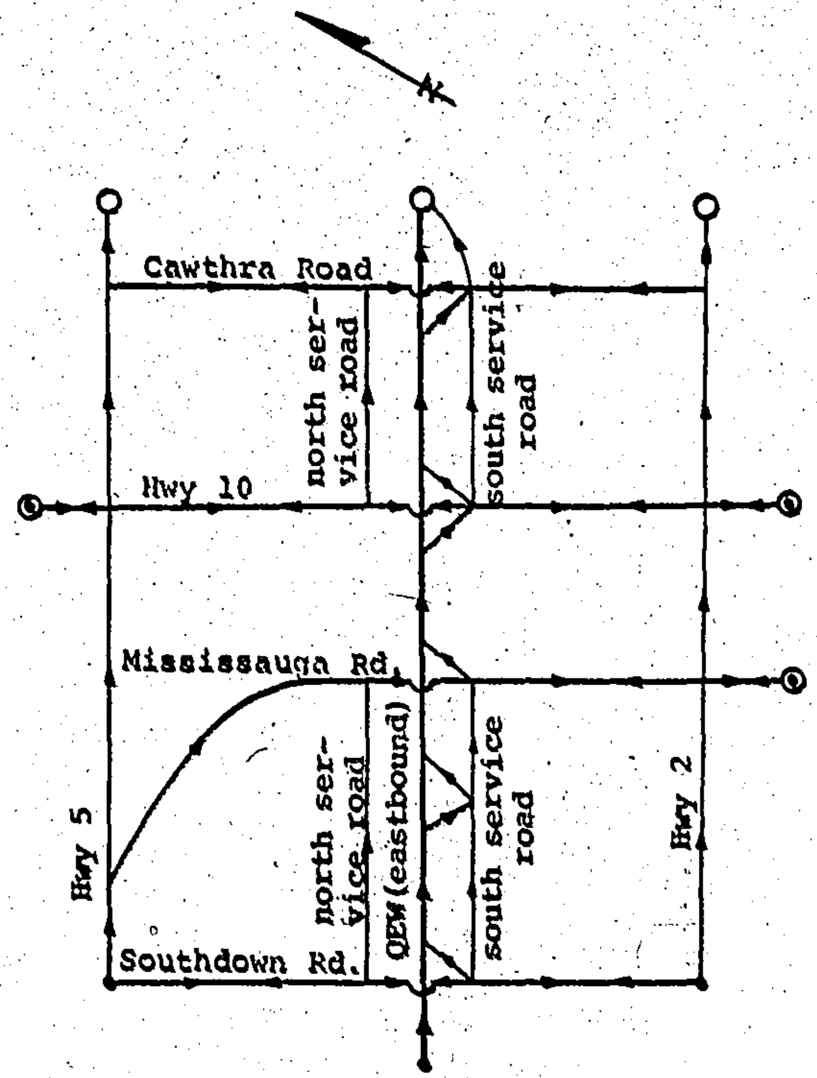
An application of the model to a real corridor has been made using hypothetical data for origin-destination (O-D) matrices and network characteristics (capacity, cost, etc.). A description of the study corridor and the associated data describing its characteristics are presented. For illustrative purposes of this chapter only one time slice of a 15-minute length is used. The purpose of this application is to investigate the following:

a) Comparing the results when traffic assigns itself approximately according to Homburger's method of total diversion ($r=0$), with those when it assigns itself according to Yagar's method of no diversion ($r=\infty$). Both results then are compared when assignment is made according to a combination yielding partial diversion ($0<r<\infty$)

b) The effect of diversion consideration on surface streets resulting when certain metering control strategies are employed.

5.2 Study Corridor

The study corridor shown in Fig. 5-1 is located in Mississauga. It consists of Queen Elizabeth Way (Q.E.W.) and its service roads, Highway 5, and Highway 2 in the longitudinal



• Origin

◎ Origin-Destination

○ Destination

FIG. 5-1 STUDY CORRIDOR

direction of the corridor. The joining roadways are Southdown Road, Mississauga Road, Highway 10, and Cawthra Road.

The link-node representation of the corridor is shown in Fig. 5-2.

At the Highway 10-Highway 5 intersection, links are used to account for the cost of turning movements. A dummy link 27-38 is used for the west approach to incorporate the effect of turning movements on intersection capacity.

To obtain the turning volumes at intersection 34, say, no turning links are required and these volumes are calculated in the program.

Turn prohibitions used for avoidance of illogical paths are determined automatically and printed out by the program. These are indicated in Fig. 5-2.

It should be noted that this study corridor is not sufficient for a complete analysis of the problems inherent in Q.E.W. traffic operations and control. Such analysis should include Dixie and West Mall Roads and their interchanges with the Q.E.W., as well as careful choice of additional origins and destinations which might be connected to the corridor by dummy links. Also, other major intersections might be identified and represented as shown by the Highway 5-Highway 10 intersection.

5.3 Flow Characteristics and Demands

The flow characteristics of the corridor links were chosen to be as close to actual characteristics as possible

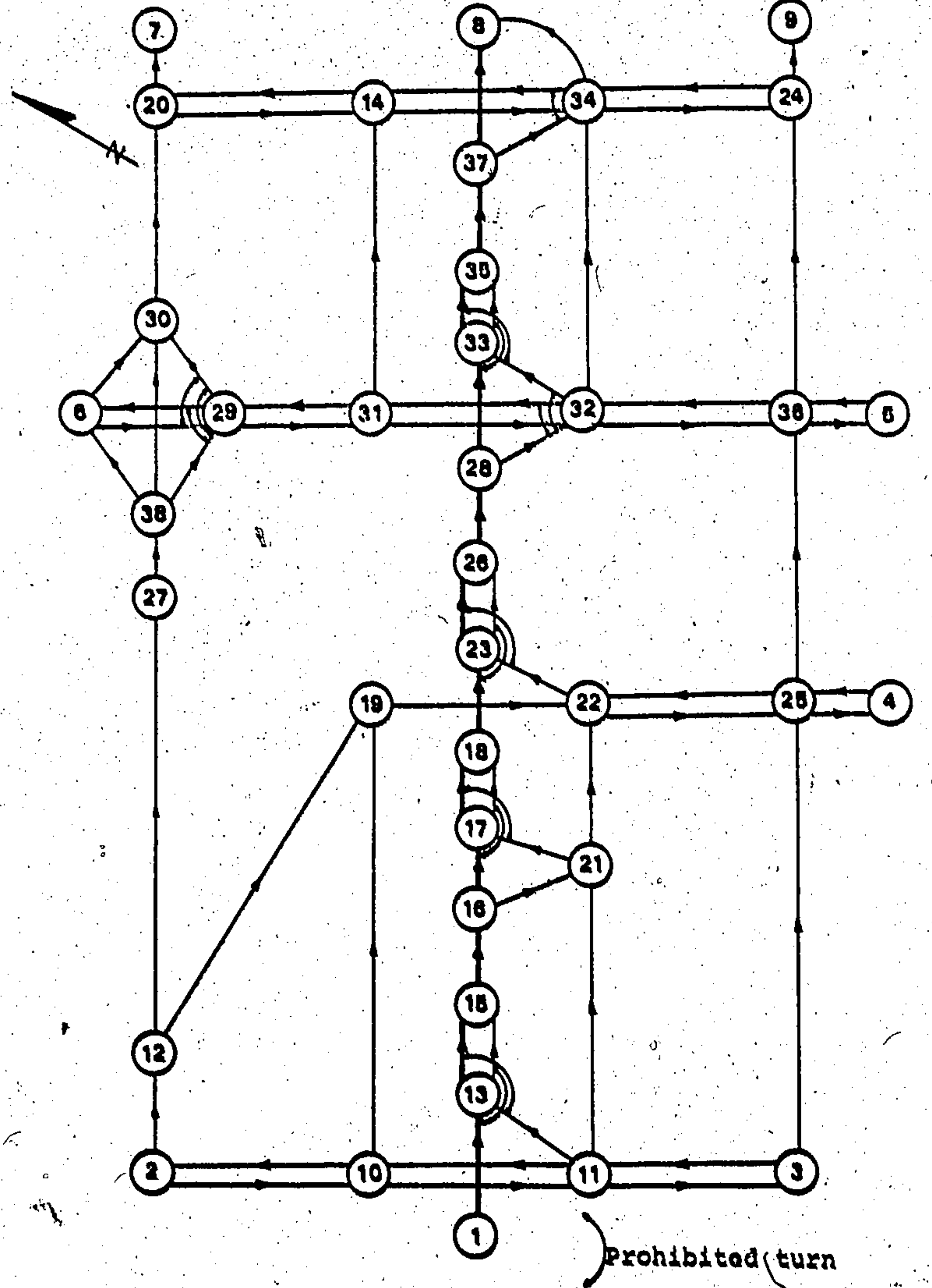


FIG. 5-2 LINK-NODE REPRESENTATION OF STUDY CORRIDOR

and are shown in Table 5-1. Each of these links is identified by its upstream and downstream node numbers. The relationship of travel cost vs. flow of each link is represented by three constant components. Therefore, each link has three sub-link capacities with respective constant unit flow costs. Link capacity is expressed in vehicles per time slice, while unit flow cost is in seconds. Associated with each link is a value representing its physical queue capacity (expressed in number of vehicles) and another value for equivalent through flow. This latter value must be equal to 1.0 for all network links except those included for left or right turns where equivalent through flows are provided. In the last four columns information on the upstream links feeding each link directly is provided. The identification numbers of these links are given.

It should be noted that if there is a turn prohibition between an upstream link and a downstream link of a node, the former is not provided in list of upstream links feeding directly the latter. For example, a turn prohibition exists between links 37-34 and 34-14, ie. between links 68 and 61. In Table 5-1, link 68 is not provided in list of upstream links for link 61.

In Table 5-2, the data for dummy merging sections of the study corridor is given. Entitlement capacity is the approach capacity to accept vehicles. The former is set equal to the metering rates when ramp control strategies are employed.

The hypothetical origin-destination demand matrix for the

TABLE 5-1 LINK-FLOW CHARACTERISTICS OF STUDY CORRIDOR

Link no.	Link		Sublink Capacities			Sublink unit costs (seconds)			Thr. flow equiv.	Storage cap.	Upstream links numbers			
	From	To	1	2	3	1	2	3			1	2	3	4
1	1	13	1800	100	100	5	6	9	1	60				
2	2	10	300	100	100	40	60	80	1	300				
3	2	12	400	200	150	15	20	25	1	100	10			
4	3	11	300	100	100	40	60	80	1	400				
5	3	25	400	200	150	90	100	110	1	600	13			
6	4	25	250	100	50	15	20	25	1	50				
7	5	36	300	100	100	15	20	25	1	50				
8	6	29	500	0	0	5	5	5	1	50				
9	6	30	300	0	0	5	5	5	1	50				
10	10	2	300	100	100	40	60	80	1	300	14			
11	10	11	500	0	0	10	10	10	1	50	2			
12	10	19	250	0	0	270	270	270	1	620	2	14		
13	11	3	300	100	100	40	60	80	1	400	11			
14	11	10	500	0	0	10	10	10	1	50	4			
15	11	13	500	0	0	20	20	20	1	40	4	11		
16	11	21	150	100	50	80	100	130	1	600	4	11		
17	12	19	250	100	50	130	160	210	1	600	3			
18	12	27	400	200	150	100	140	180	1	700	3			

TABLE 5-1 (Continued)

Link no.	Link		Sublink Capacities			Sublink unit costs (seconds)			Thr. flow equiv.	Storage cap.	Upstream links numbers			
	From	To	1	2	3	1	2	3			1	2	3	4
19	13	15	1200	0	0	1	2	3	1	0	1			
20	13	15	350	0	0	10	10	10	1	0	15			
21	14	20	300	100	100	40	60	80	1	300	51	61		
22	14	34	500	0	0	10	10	10	1	50	51	31		
23	15	16	1300	100	100	50	60	90	1	575	19	20		
24	16	17	1300	100	100	2	3	3	1	25	23			
25	16	21	350	0	0	20	20	20	1	20	23			
26	17	18	1200	0	0	1	2	3	1	0	24			
27	17	18	450	0	0	10	10	10	1	0	32			
28	18	23	1300	100	100	3	4	5	1	40	26	27		
29	19	22	250	100	50	2	3	5	1	50	12	17		
30	20	7	400	200	150	40	60	80	1	380	21	50		
31	20	14	300	100	100	40	60	80	1	300	50			
32	21	17	500	0	0	20	20	20	1	20	16	25		
33	21	22	150	100	70	10	20	25	1	20	16	25		
34	22	23	500	0	0	20	20	20	1	40	33	41	29	
35	22	25	250	100	50	40	60	80	1	400	29	33		
36	23	26	1200	0	0	1	2	3	1	0	28			

TABLE 5-1 (Continued)

Link no.	Link		Sublink Capacities			Sublink unit costs (seconds)			Thr. flow equiv.	Storage cap.	Upstream links numbers			
	From	To	1	2	3	1	2	3			1	2	3	4
37	23	26	450	0	0	10	10	10	1	0	34			
38	24	9	400	200	150	40	60	80	1	380	65	62		
39	24	34	300	100	100	50	60	80	1	400	65			
40	25	4	250	100	50	15	20	25	1	300	5	35		
41	25	22	250	100	50	40	60	80	1	400	5	6		
42	25	36	400	290	250	15	20	25	1	300	5	35	6	
43	26	28	1300	100	100	10	15	20	1	270	36	37		
44	27	38	500	300	200	2	3	5	1	0	18			
45	28	32	450	0	0	20	20	20	1	20	43			
46	28	33	1100	100	100	4	6	10	1	110	43			
47	29	6	500	0	0	5	5	5	1	50	52			
48	29	30	200	0	0	5	5	5	1	0	52			
49	29	31	300	100	100	40	60	80	1	300	8	70		
50	30	20	400	200	150	40	60	80	1	400	9	71	48	
51	31	14	150	100	50	60	80	100	1	400	49	54		
52	31	29	300	100	100	40	60	80	1	300	54			
53	31	32	500	0	0	10	10	10	1	50	49			
54	32	31	500	0	0	10	10	10	1	50	66			

TABLE 5-1 (Continued)

Link no.	Link From	Link To	Sublink Capacities 1	Sublink Capacities 2	Sublink Capacities 3	Sublink unit costs (seconds) 1	Sublink unit costs (seconds) 2	Sublink unit costs (seconds) 3	Thr. flow equiv.	Storage cap.	Upstream links numbers 1	Upstream links numbers 2	Upstream links numbers 3	Upstream links numbers 4
55	32	33	500	0	0	20	20	20	1	50	66	53		
56	32	34	150	100	50	90	120	140	1	400	45	53	66	
57	32	36	300	100	100	40	60	80	1	400	45	53		
58	33	35	1200	0	0	1	2	3	1	0	46			
59	33	35	450	0	0	10	10	10	1	0	55			
60	34	8	300	200	100	60	80	100	1	380	56	68	22	39
61	34	14	500	0	0	10	10	10	1	50	56	39		
62	34	24	300	100	100	40	60	80	1	400	56	22	68	
63	35	37	1000	150	200	25	30	50	1	250	58	59		
64	36	5	300	120	100	15	20	25	1	50	42	57		
65	36	24	400	200	150	70	80	90	1	400	42	57	7	
66	36	32	400	200	150	40	60	80	1	400	7	42		
67	37	8	1000	50	100	30	40	60	1	380	63			
68	37	34	350	0	0	20	20	20	1	20	63			
69	38	6	200	0	0	5	5	5	1	0	44			
70	38	29	200	0	0	5	5	5	1	0	44			
71	38	30	400	200	150	5	5	5	1	0	44			

TABLE 5-2 ORIGINAL MERGING SECTION CAPACITIES

Merge No.	Merge description	Entitlement Capacity (veh/time slice)		Ultimate Capacity (veh/time slice)	
		Freeway approach	Ramp approach	Freeway approach	Ramp approach
1	Southdown Road	1320	375	1400	450
2	Mississauga 1	1320	375	1400	450
3	Mississauga 2	1320	375	1400	450
4	Highway 10	1320	375	1400	450

TABLE 5-3 ORIGIN-DESTINATION DEMANDS (veh/time slice)

Origin Node Number	Destination Node Number 4	5	6	7	8	9
1	100	160	140	100	610	60
2	100	100	100	100	300	100
3	75	90	0	85	170	90
4	0	0	100	140	150	0
5	0	0	0	200	240	50
6	0	100	0	100	340	100

15-minute time slice is shown in Table 5-3. These demands were chosen in such a way as to produce queueing on the freeway so that all aspects of the analysis could be investigated.

5.4 Results

Results of the application were obtained for two main phases:

a) Calibration Phase

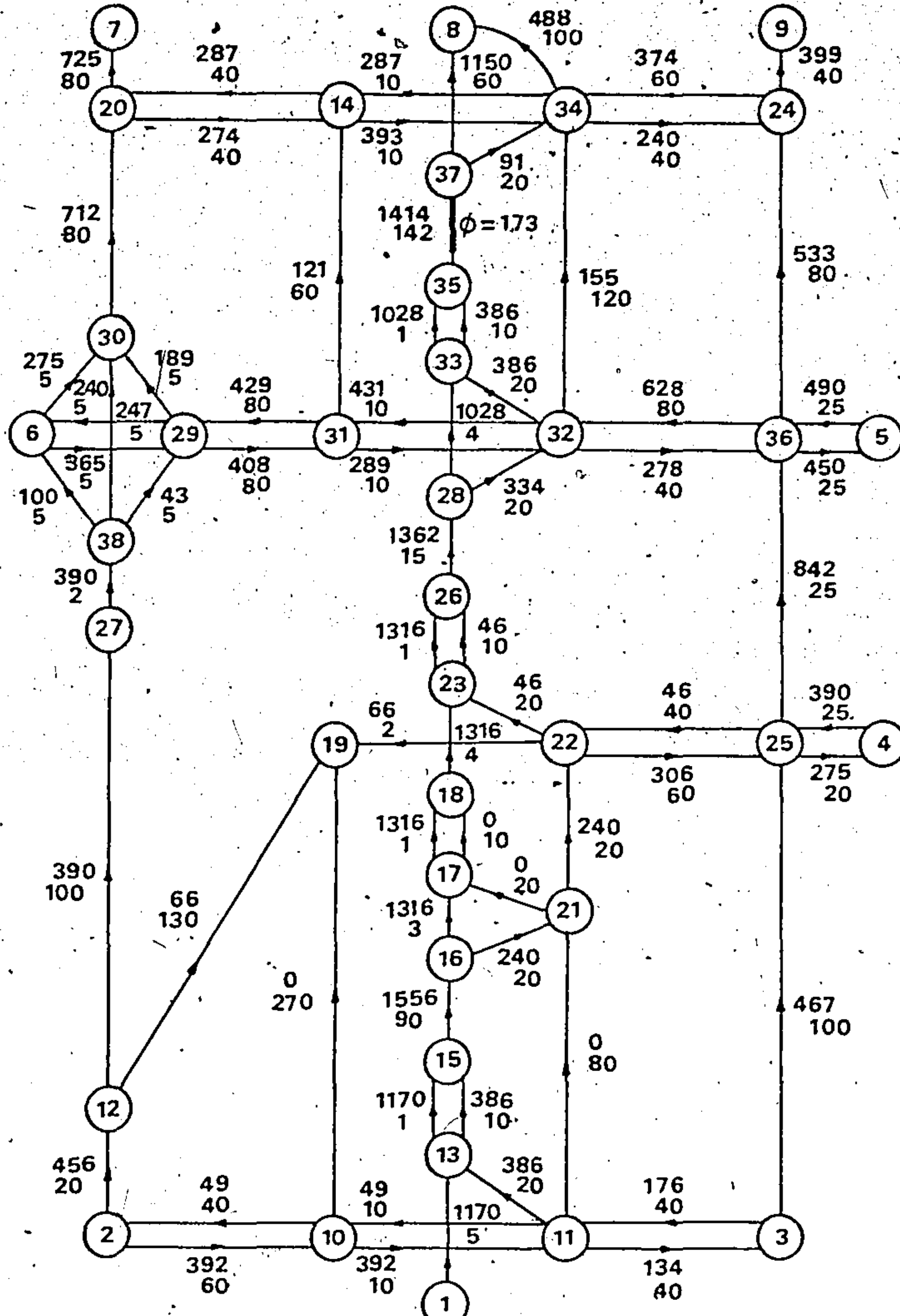
The purpose of this phase was to calibrate the diversion model to determine the best value of the constant r for use in this model. Different values of r were assumed and the resulting flows and queues were obtained. If data were available for actual flows and queues, the best value of r would have been chosen in this manner. For purposes of illustration, one value of r was arbitrarily assumed for use in the control phase.

b) Control Phase

Having been determined, the best value of r was used in applying the model for predicting flows and queues for two ramp metering strategies. Also, the predicted flows and queues were obtained for the no diversion case as a matter of comparison.

5.4.1. Calibration Phase Results

In this case the O-D demands were first assigned to the network without diversion consideration ($r=\infty$) or ramp control. The resulting flows, queues and unit travel costs for the time slice considered are shown in Fig. 5-3. It can be seen from this figure that a queue of $\emptyset$ = 173 vehicles is formed on freeway link 35-37 due to link 37-8 reaching capacity.



134 flow (veh/time slice)
40 travel cost (seconds)

FIG. 5-3 PREDICTED FLOWS AND QUEUES WITH NO DIVERSION AND NO RAMP CONTROL

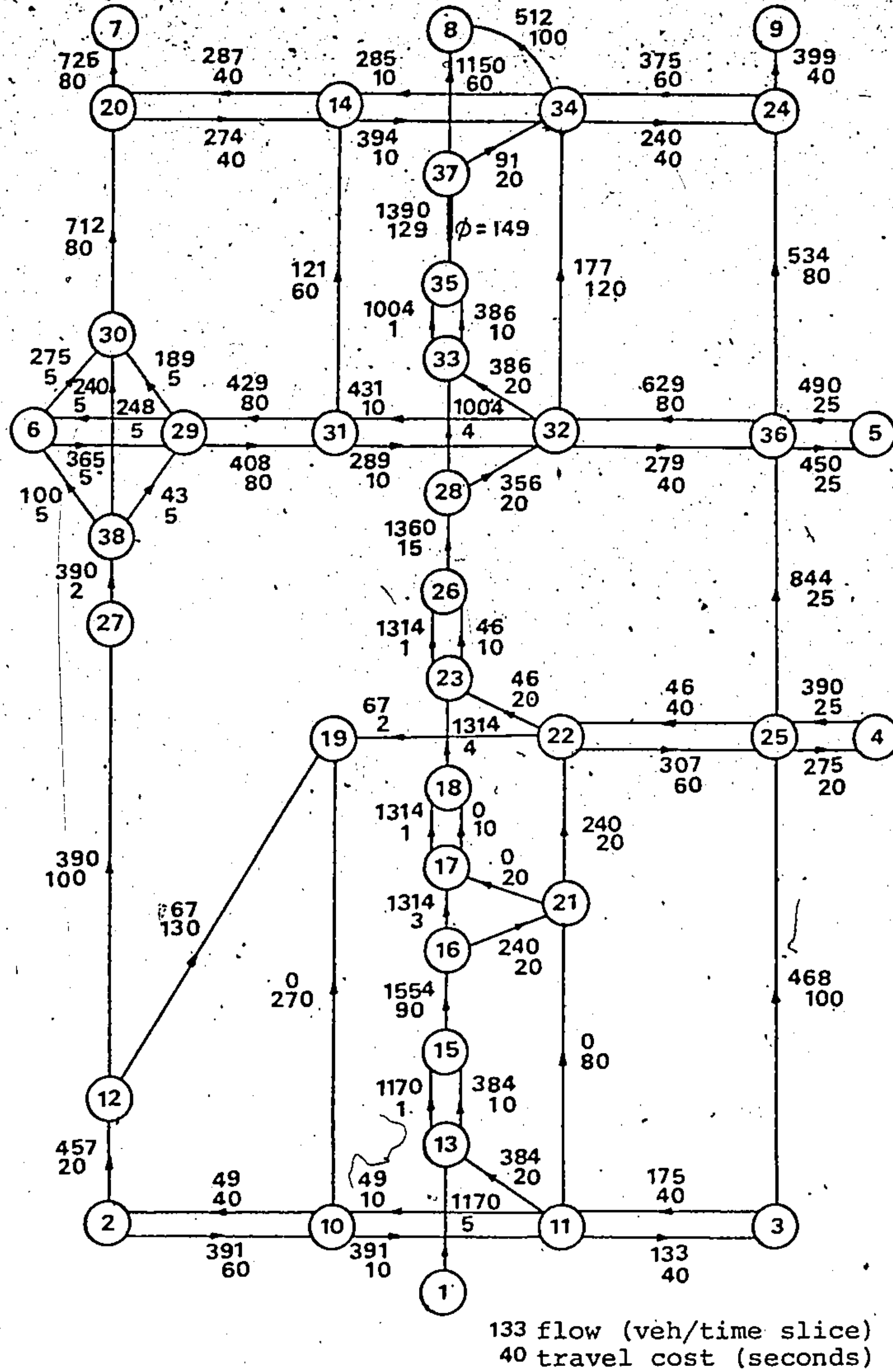
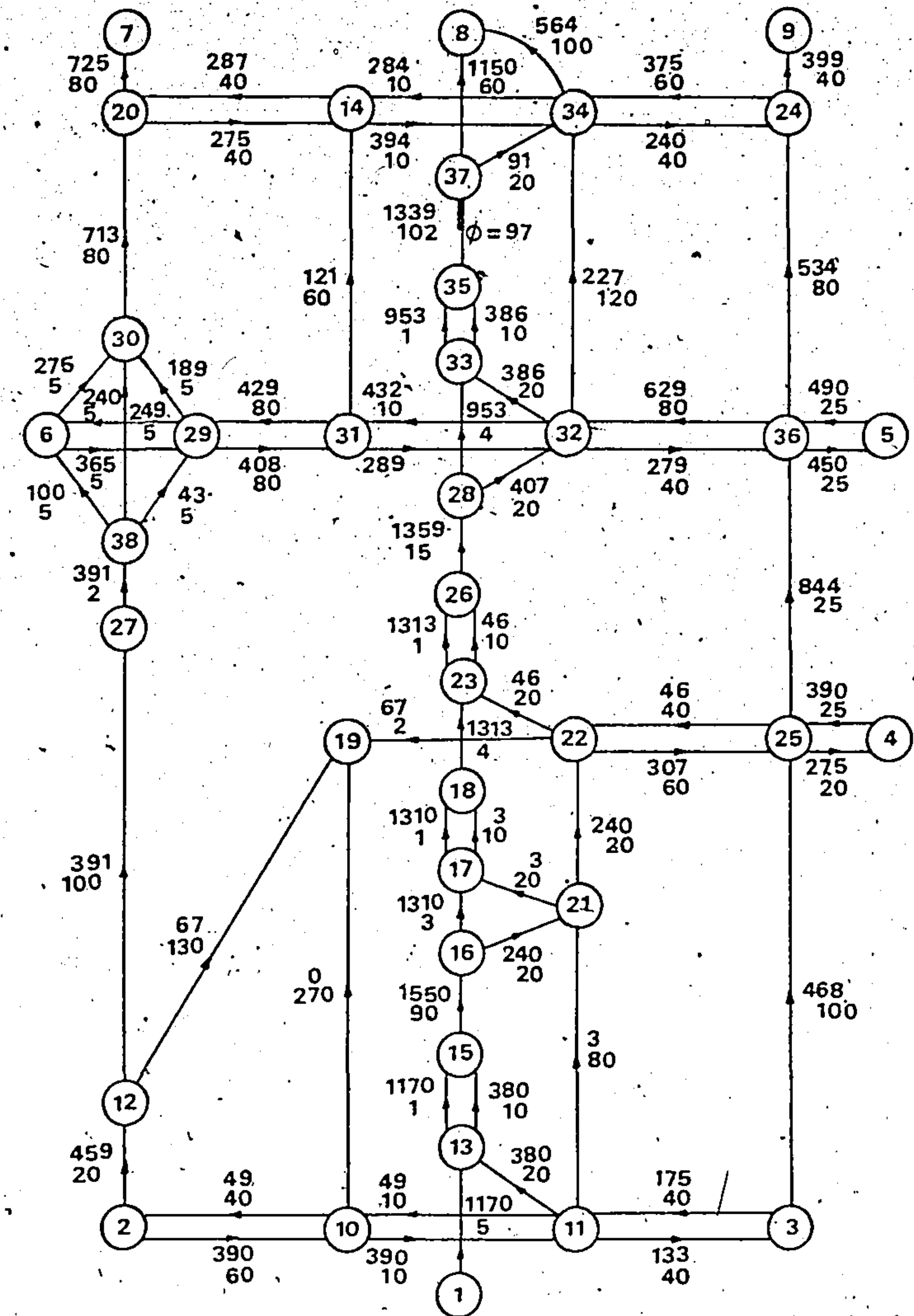


FIG. 5-4 PREDICTED FLOWS AND QUEUES WITH DIVERSION (r=3) AND NO RAMP CONTROL



133 flow (veh/time slice)
40 travel cost (seconds)

FIG. 5-5 PREDICTED FLOWS AND QUEUES WITH DIVERSION (r=0.5) AND NO RAMP CONTROL

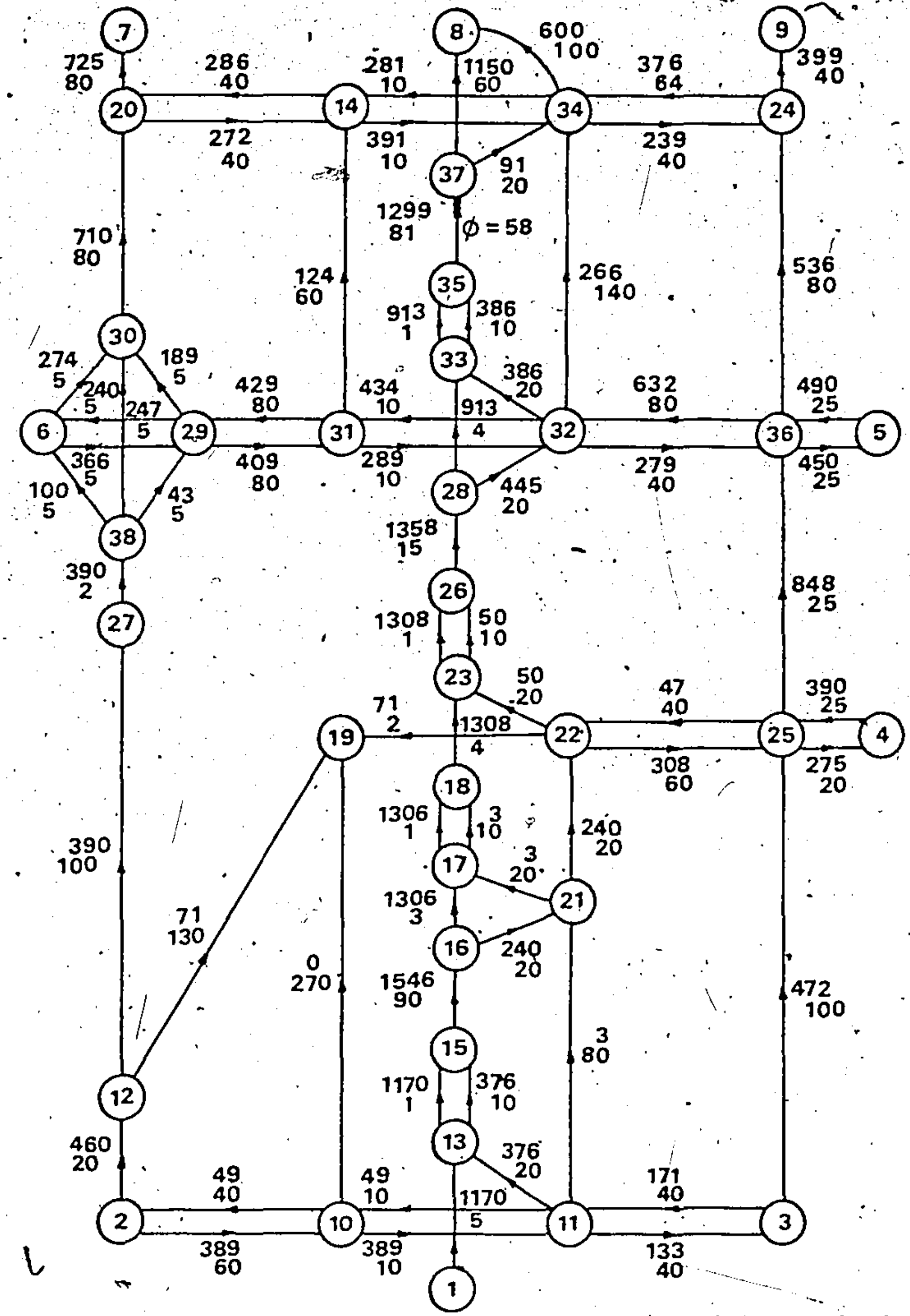


FIG. 5-6 PREDICTED FLOWS AND QUEUES WITH DIVERSION (r=0-1) AND NO RAMP CONTROL

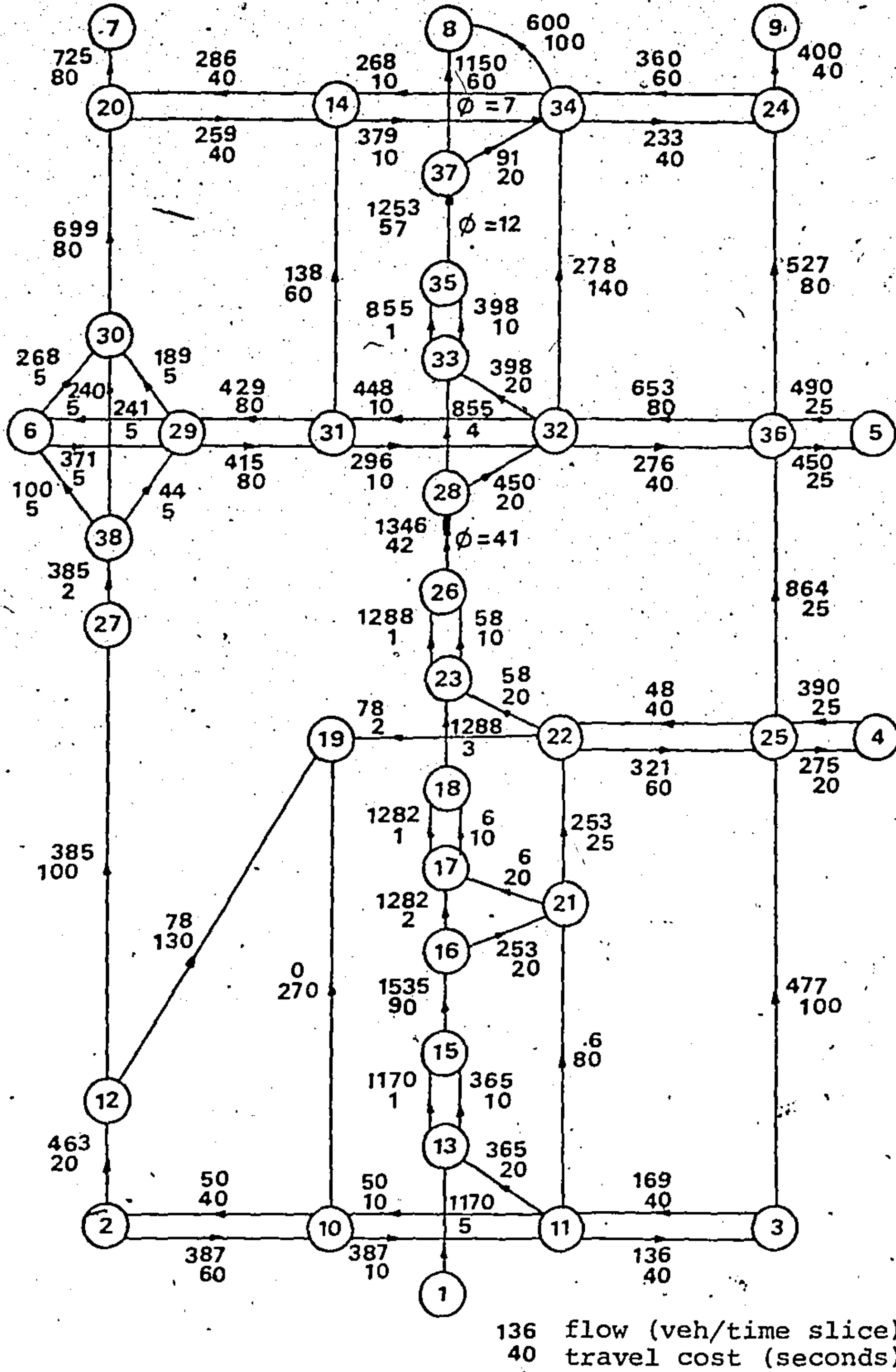


FIG. 5-7 PREDICTED FLOWS AND QUEUES WITH DIVERSION (r=.0) AND NO RAMP CONTROL

To establish the best value of the diversion model constant to be used when ramp control is employed, different values of r were assumed and the resulting flows and queues were obtained. Using values of r as 3.0, 0.5, 0.1, and 0.0, the resulting flows, queues, and unit travel costs shown in Figures 5-4 through 5-7 were respectively obtained. By way of example, the queue lengths on link 35-37 are summarized in Table 5-4 for the different values of r. Table 5-5 gives the total travel cost spent in the corridor for each value of r.

5.4.2. Control Phase Results

The purpose of this phase was to evaluate different ramp control strategies in an attempt to choose the best strategy to be employed. Additionally, the detailed effects of such a strategy on the entire corridor could be identified before the actual implementation. The two strategies shown in Table 5-6 were used. The predicted flows and queues for each were then obtained with and without diversion. In the diversion case, the value r = 0 was assumed to be best (total diversion). Figs. 5-8 through 5-11 show the predicted flows and queues for each of the four cases, and the total travel cost spent in the corridor is given in Table 5-7.

5.5 Discussion of Results

In the calibration phase, results were obtained first with no diversion considered. This is equivalent to using a value $r = \infty$ in the diversion model, ie. all drivers do not penalize queueing time and therefore they are all assigned to the first minimum paths. This results in one queue of 173 vehicles on freeway

TABLE 5-4 QUEUE LENGTH ON LINK 35-37

Value of r	Queue length (vehicles)
∞	173
3.0	149
0.5	97
0.1	58
0.0	12

TABLE 5-5 TOTAL TRAVEL COST IN THE CORRIDOR
(veh-hrs/time slice)

Value of r	Total travel cost
∞	334
3.0	330
0.5	320
0.1	313
0.0	312

TABLE 5-6 CONTROL STRATEGIES ADOPTED FOR THE TIME SLICE CONSIDERED

Merge No.	Metering Rates (veh/time slice)	
	Strategy No. 1	Strategy No. 2
1	290	300
2	0 (closed)	0 (closed)
3	0 (closed)	290
4	290	0 (closed)

TABLE 5-7 TOTAL TRAVEL COST FOR CONTROL STRATEGIES EMPLOYED (veh-hrs/time slice)

Strategy No.	Diversion (r=0)	No Diversion (r=∞)
1	315	317
2	315	320

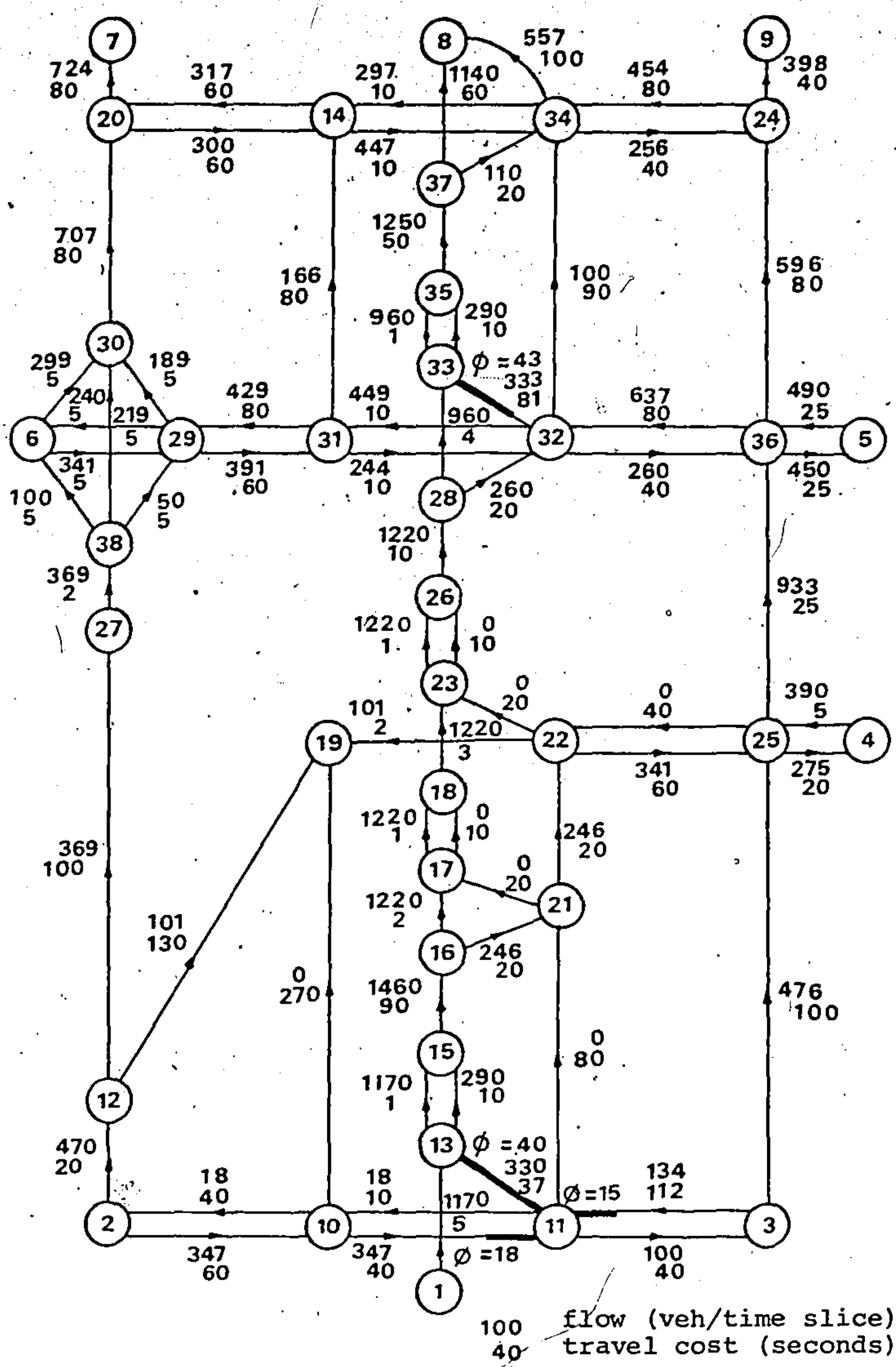
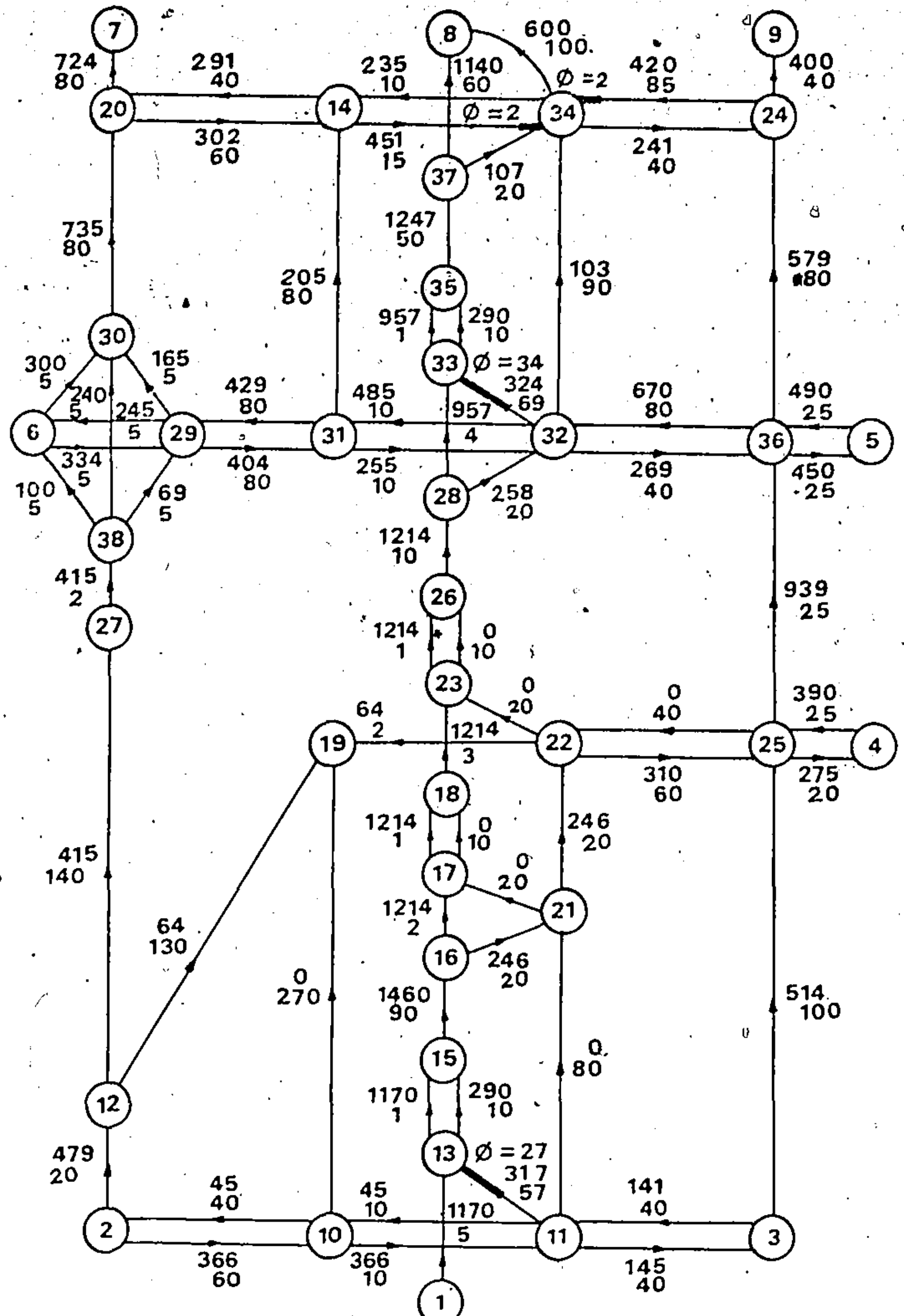


FIG. 5-8 PREDICTED FLOWS AND QUEUES WITH NO DIVERSION - RAMP CONTROL STRATEGY #1



145 flow (veh/time slice)
40 travel cost (seconds)

FIG. 5-9 PREDICTED FLOWS AND QUEUES WITH DIVERSION (r=0.0) - RAMP CONTROL STRATEGY #1

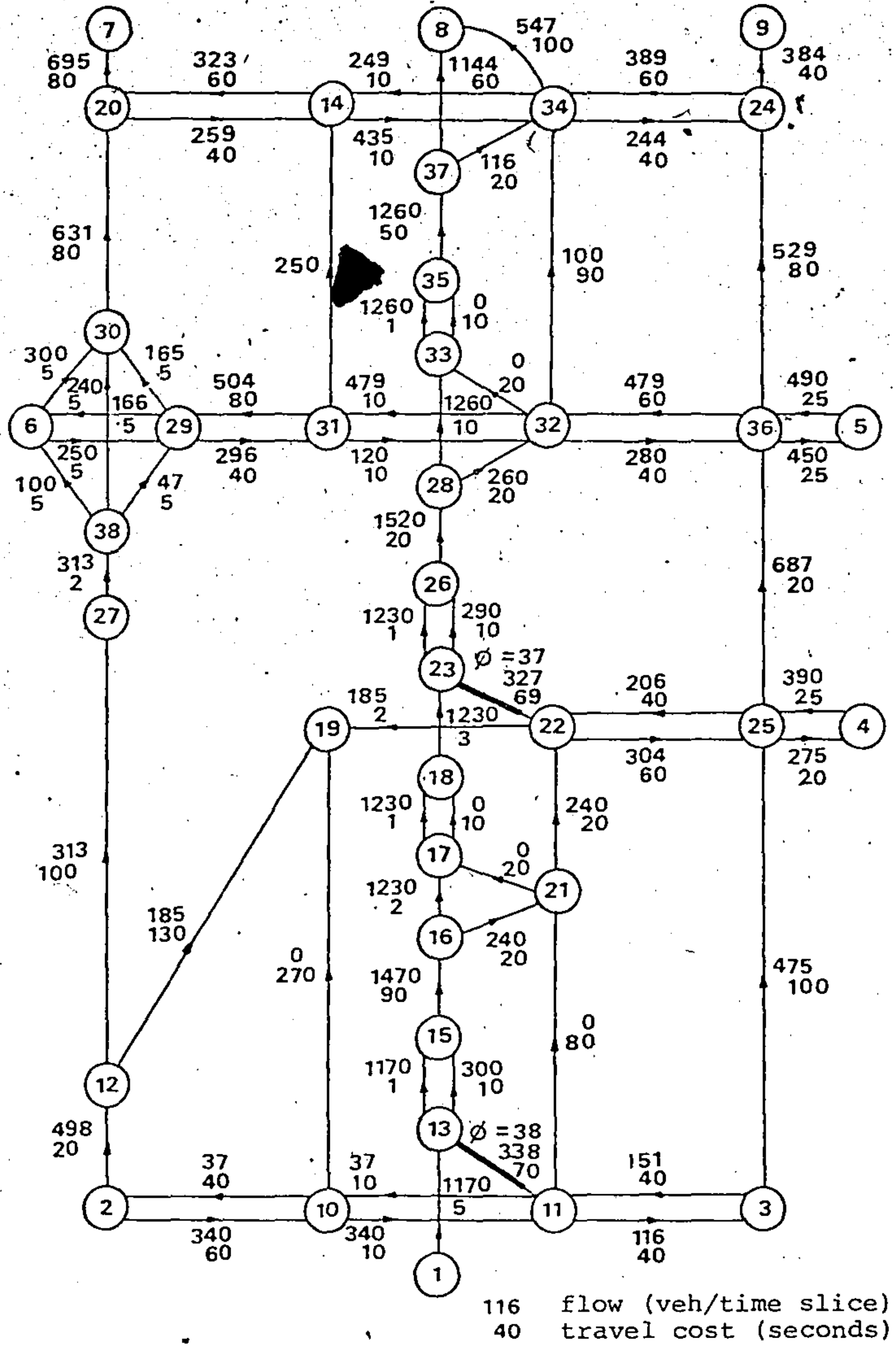


FIG. 5-10 PREDICTED FLOWS AND QUEUES WITH NO DIVERSION - RAMP CONTROL STRATEGY #2

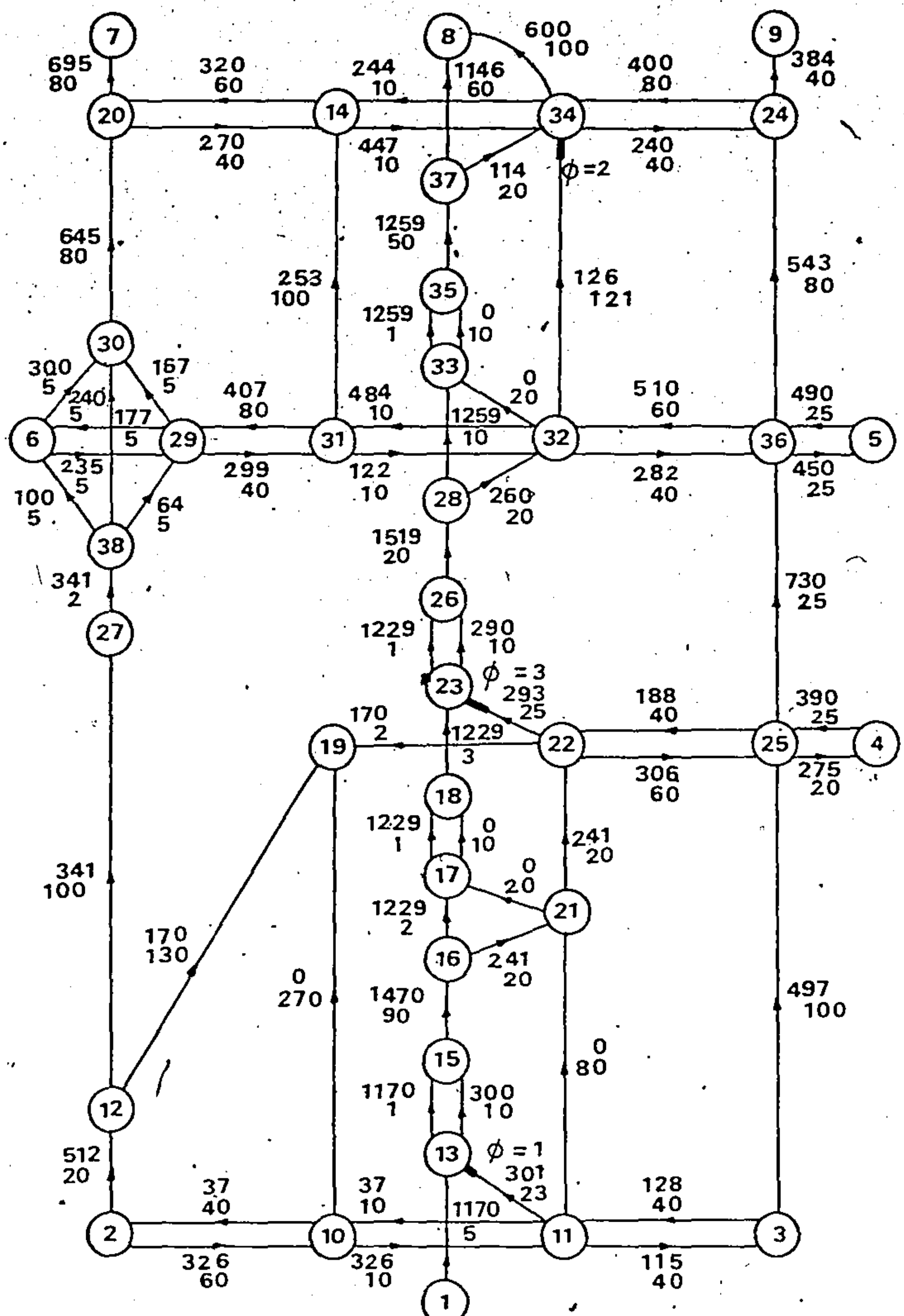


FIG. 5-11 PREDICTED FLOWS AND QUEUES WITH DIVERSION (r=0.0) - RAMP CONTROL STRATEGY #2

link 35-37. For the values r = 3.0 and r = 0.50, the queue lengths on link 35-37 decreased due to traffic diversion. For r = 0.1, the diverted traffic caused link 34-8 to reach capacity, resulting in queues on links 14-34 and 24-34. In the extreme case, where all traffic is diverted (r = 0), the assignment resulted in a queue on links 35-37, 14-34, and 26-28 since links 37-8, 34-8, and 28-32 respectively reached capacity.

The value r = 0 resulted in an assignment equivalent to Homburger's method as long as non-saturated alternative routes are available. After these routes are saturated, the assignment is performed to the saturated paths of minimum travel cost with no diversion. It is felt that this might be worthwhile to consider for some networks where drivers wish to avoid queueing paths, but are forced to use them when non-queueing paths are not available.

From Table 5-5 it is interesting to note that diversion of traffic always results in a reduction in the total travel time spent in the system although traffic is diverted to paths having longer travel costs. This occurs since the diversion procedure distributes traffic on non-saturated paths and consequently reduces the number of queueing vehicles in the network. The queueing cost is calculated on the basis that queues evolve uniformly during a given time slice and dissipate uniformly during the whole length of another time slice. Consequently, the time spent in a certain queue is the number of queueing vehicles multiplied by the length of the time slice. Therefore, under normal circumstances, the diversion of traffic reduces the queueing cost by a known amount and increases the travel cost on alternative routes by a lesser amount.

In the control phase when strategy #1 was employed with no diversion, queues were formed on links 32-33 and 11-13 with the latter spilling back through surface street links 3-11 and 10-11, as shown in Fig. 5-8. With total diversion, the situation is improved but some queues were still formed on the metered entrance-ramps. Surface street link 34-8 reached capacity causing additional queues on upstream links 24-34 and 14-34. However, this can be interpreted as follows. When metered ramps are assigned flows equal to the metering rates, no queues are formed and all drivers assign themselves to alternate routes which are not saturated. This results in surface street link 34-8 reaching its capacity. After that, drivers wishing to reach destination 8 do not find any non-saturated paths to that destination and must choose the best saturated path.

When strategy #2 was considered with no diversion, it also resulted in queues on the metered ramps. With the diversion case, almost no queues were present on the ramps and link 34-8 was saturated with a few queued vehicles. The analysis of these results is identical to that of strategy #1. In addition, when comparing the total cost of the two strategies shown in Table 5-7, it is noted that they give the same total travel cost in the diversion case. This is also less than that of the no diversion case.

It would appear from this particular example that the results of the previous cases produced queues at significantly different locations of the network. This means that if one could determine accurately the diversion characteristics, the resulting assignment would identify the actual locations which would suffer from poor operations. Appropriate actions for improvement could then be made before actual implementation of a certain strategy. For example, the no diversion case(strategy#1 indicates that the ramp queue on link 11-13 will spillback onto upstream links while the diversion case indicates that the poor operations on surface streets is at intersection 34.

It can be concluded that inclusion of traffic diversion in assignment is an essential feature for accurately predicting flows and queues in a corridor network. However, diversion characteristics must be determined precisely if one wishes to achieve results describing actual traffic behaviour in an accurate way.

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary

A model for investigating traffic assignment and control in a freeway corridor has been developed. It assigns time-varying demands according to the minimum individual travel cost principle. The model accounts for traffic diversion from congested minimum paths, with three options: total diversion, [illegible] diversion, or partial diversion. To accomplish this, a diversion model was developed which diverts drivers according to their equivalent queueing costs.

A simple link-node representation was described and a detailed explanation of network illogical paths was given along with a technique which determines automatically these illogical paths. A completely new minimum path algorithm with turn prohibitions was developed to account for these illogical paths. In addition, a method for calculating turning volumes without using turning links was developed.

An application of the model to a real corridor using hypothetical data was made. The results of the application were included to show the effects of the various diversion strategies and to illustrate basic use of the model. On the basis of the above studies the following conclusions and suggestions have been drawn.

6.2 Conclusions

1. The minimum path algorithm with turn prohibitions developed in this thesis demonstrated its efficiency in accurately accounting for turn prohibitions. It is important to note that the algorithm can be used for networks without turn prohibitions. The computer running time of the algorithm is expected to be proportional to the number of turn prohibitions, and it can, therefore, be used even for networks with a few turn prohibitions.

   The only limitation of this algorithm is that it allows a maximum of 4 entering links at any turn prohibition node. However, if desirable, more than 4 entering links could be allowed through a straight forward change in the algorithm.

2. The minimum path algorithm also proved its usefulness in allowing a simple link-node representation. If turn prohibitions were included using the conventional method of link-node representation, the increase in coding effort over the proposed method would exceed 40 percent.

3. The method of calculating turning volumes without using turning links can be utilized in other traffic assignment programs used for transportation planning to provide turning volumes at freeway interchanges and at intersections without the need for turning links. The method is obviously advantageous in those programs which use turning links for the sole purpose of obtaining turning volumes, such as TNET 90 [8].

4. The model can be applied for determining partial optimum metering rates which results in a queue-free freeway. However, this feature will only be possible when congestion occurs due to downstream merging links.
5. The model can effectively be applied for testing and evaluating metering control strategies. The best strategy can then be chosen and resulting flows and queues can be accurately predicted before the implementation of that strategy.
6. The application presented in this thesis for the Q.E.W. freeway corridor was made only for illustrative purposes. It is recommended that for real applications, the study section must bound the freeway congestion. Although the study corridor does not represent completely the actual corridor in terms of locations of origins and destinations, such items are extremely important in obtaining accurate results.
7. The total travel cost in the corridor was found to decrease with diversion consideration. Although drivers divert to non-queueing alternative routes having larger travel cost, their contribution to the total cost is in favour of the system. This is due to the large decrease in queueing cost outweighing the increases in travel costs on the alternative routes.

6.3 Suggestions for Further Research

1. The method proposed for calculating turning volumes without turning links could, with some additional work,

be used to account for turning costs in the developed minimum path algorithm with turn prohibitions. The resulting minimum path algorithm could use the turning costs as a function of the turning volumes and not simply a constant turn penalty.

2. A more sophisticated micro-analysis for merging and diverging sections could be included. The capacities of merge and diverge approaches of ramps may be calculated on the basis of freeway lane-1 volume rather than the volume of the whole freeway approach. In this case, the overall merge and diverge capacities could be obtained from the existing Highway Capacity Manual Procedures [7], or updated according to the latest techniques available.
3. The procedure adopted for turning volume calculation without turning links may be used to account for the effect of turning volumes on intersection capacity. Instead of using constant through equivalents regardless of the amount of turning volumes, through equivalents could be related to the percentage of the turning volumes with respect to the total approach volume.
4. Calibration of the diversion model is essential. In that calibration, the best value of the diversion constant for the study network must be determined. However, if data on equivalent queueing cost could be collected, the exact relation between the percent of traffic diverted and equivalent queueing cost could be constructed and calibrated. The resulting relation need not be as represented in this thesis and may in fact take another form.

5. The diversion formula used to divert some traffic from the minimum-queueing path to the minimum non-queueing path uses a constant diversion curve for the whole network. Actually there might be different curves for different paths according to trip length, purpose, driver characteristics, etc. It is felt that investigation of the effect of these factors on traffic diversion will permit the actual route choice phenomena to be correctly represented.
6. A method is required in the traffic assignment technique for dynamically estimating the final weaving capacity on the basis of assigned flows. If developed, this method could be inserted directly into the model.

APPENDICES

APPENDIX A

DESCRIPTION OF THE METHODS USED FOR DETERMINING NETWORK ILLOGICAL PATHS

Turn prohibitions are used to avoid illogical paths at interchanges, intersections, and at merging and weaving sections. They are also used to account for any network control such as prohibition of a left turn. A method is proposed for determining these turn prohibitions automatically in the program. In addition, illogical paths through origin-destination (O-D) nodes are avoided by another method using a link-delineation type of procedure.

A.1 Determination of Turn Prohibitions

The procedure of determining network turn prohibitions is dependent mainly on the data provided for the upstream links feeding each link in the network. The basic use of this data is to account for queue spillback in the assignment procedure. However, this data is also utilized for determining the upstream and downstream links of any illogical path in the network.

The procedure for determining network turn prohibitions described below, is repeated for each node in the network except O-D nodes:

a) the upstream links of that node are determined

b) Each link I of those upstream links is checked against the list of upstream links feeding each downstream link J of that node.

c) If link I is not found in that list of upstream links feeding link J, a turn prohibition between links I and J is assumed.

Using this procedure turn prohibitions are determined in ascending order of the downstream node number of the turn prohibition upstream link. This is an important facility provided to reduce the computation time of the minimum path algorithm described in Chapter 2.

To clarify the above procedure consider the interchange and its link-node representation shown in Fig. A-1. The input data for this example is given in Table A-1.

TABLE A-1: DATA PROVIDED FOR EXAMPLE NETWORK

Link no.	Nodes From	Nodes To	Upstream Links Feeding each Link			
1	1	2	10			
2	1	4	10			
3*	2	7	1			
4**	2	7	6			
5	3	4				
6	4	2	9			
7	4	3	9			
8	4	5	2	5		
9	5	4				
10	6	1				

* Freeway dummy approach link

** Ramp dummy approach link

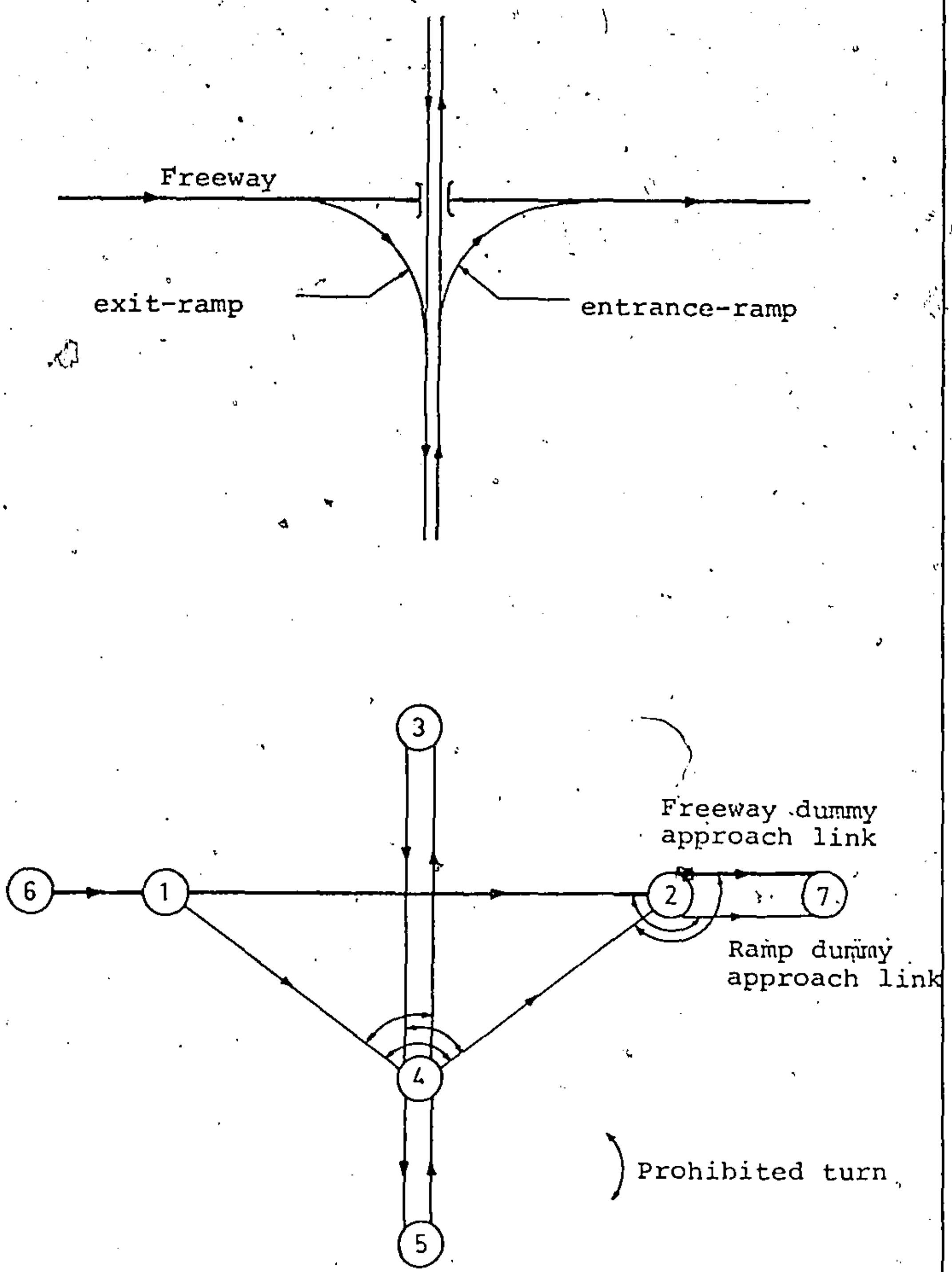


FIG. A-1 EXAMPLE OF DETERMINING TURN PROHIBITIONS

Suppose now we apply the above procedure of turn prohibitions at node 4. The following are found:

a) There are three upstream links having numbers 2, 5, and 9.

b) Each of these links is checked against the upstream links feeding each downstream link of that node. The downstream link numbers are 6, 7, and 8.

c) Link number 2 is not found in the upstream links feeding link numbers 6 and 7 and therefore, turn prohibitions are considered between link number 2 and these links. Similarly, link number 5 has turn prohibitions with link numbers 6 and 7 and link number 9 has turn prohibition with link number 8.

It should be noted that if any link reaches its capacity or has a queue reaching its physical queue capacity, the upstream node of that link is considered a congested point for only those upstream links feeding that link. For example, if link 4-2 reaches its capacity, node 4 is considered a congested point for only link 5-4. For links 1-4 and 3-4, node 4 is not considered a congested point as these links do not feed link 4-2. If the movement between links 3-4 and 4-2 and between links 1-4 and 4-3 were allowed (as in a simple diamond interchange), and link 4-3 reaches its capacity, node 4 will be considered a congested point for links 5-4 and 1-4. For link 3-4, node 4 is not considered a congested point and, therefore, the movement between links 3-4 and 4-2 is allowed. It is assumed that the queues formed on upstream links do not block the intersection at node 4 which is essentially true, especially when the intersection is signalized.

B.2 Procedure For O-D Nodes

Although the procedure used for determining turn prohibitions can be used to determine illogical paths through O-D nodes, another method is used to save computer time. The logical sequences of this method are as follows:

a) From the data provided for origins and destinations, the common numbers in both are identified. Any of these numbers represents an O-D node.

b) During the minimum path procedure when any of these common nodes are found, the links out of this node are not considered in building minimum paths. However, when the minimum paths are obtained from an O-D node, the links out of this node are considered.

Ramp closure is treated in the same manner. When any ramp is closed in a certain time slice, the corresponding network link is identified and omitted from the minimum paths. It is assumed in the case of entrance-ramp closure that no drivers will queue upstream of that ramp in that time slice.

APPENDIX B

DESCRIPTION OF THE METHOD USED IN TURNING VOLUME CALCULATION

B.1 Input Data

For each intersection I, the input data to the program is one card containing the following:

1. Intersection node number. This is stored in vector INT(I)
2. Surrounding node numbers provided in the clockwise direction. These are stored in matrix NSUR (I,4)

B.2 Vectors Calculated in the Program

NS 1(J) = storage of the upstream and downstream node numbers of turning movements at all specified intersections.

TURV(J) = tentative assignment of turning volumes in each increment of the assignment

TURV1(J) = cumulative assignment of turning volumes.J ranges from 1 to the total number of turning movements to be calculated (not more than 100)

KK(I) = identification number of the first turning movement of each intersection stored in NS1(J). The last element stored in vector KK(I) is J + 1. I ranges from 1 to the number of intersections plus 1.

B.3 Procedure

a) From NSUR(I,4), the upstream and downstream nodes numbers of each turning movement are obtained and stored in NS1(J). These movements are stored in the order of their printing on the intersection drawing.

b) For each intersection, the position in NS1(J) of its first turning movement is stored in KK(I). In addition, a value equal to the number of total turning movements plus 1 is stored in that vector

c) During the assignment to the minimum path, each node number in that path is checked with the node numbers in INT(I). Also, the preceding and following nodes are continuously stored.

d) If the node number checked in previous step is found in INT(I), the location of that node's turning movements in NS1(J) is determined using the vector KK(I).

e) From that location in NS1(J) the turning movement J which has the appropriate following and preceding node numbers are identified.

f) The demand of that path is tentatively assigned to the turning movement J in TURV(J).

g) After the tentative assignment of all paths, TURV(J) is multiplied by the maximum fraction of the demand matrix that can be assigned in that increment, and the resultant value is added to the cumulative turning volume TURV1(J).

To clarify steps a and b, the following example is provided.

B.4 Example

Suppose we have two intersections as shown in Fig. B-1. It is required to prepare vectors NS1(I) and KK(I) for calculating all turning volumes of these intersections.

The input data is provided in two cards as follows:

card no. 1	4	3	1	15	2
card no. 2	6	0	19	8	7

The first number in each card is the intersection node number. In card no. 1, the surrounding nodes for intersection 4 are provided in the clockwise direction, preferably starting with the node at the top of the intersection drawing. In card no. 2, a zero must be provided first and then the surrounding node numbers in the clockwise direction, imagining that the zero value represents the missing intersection node.

Vectors NS1(J), KK(I) will be as follows:

$$NS1(J) = \begin{bmatrix} 0203 \\ 0103 \\ 0302 \\ 0301 \\ 1502 \\ 1501 \\ 0215 \\ 0115 \\ 0807 \\ 0819 \\ 0708 \\ 1908 \end{bmatrix} \qquad KK(I) = \begin{bmatrix} 1 \\ 9 \\ 13 \end{bmatrix}$$

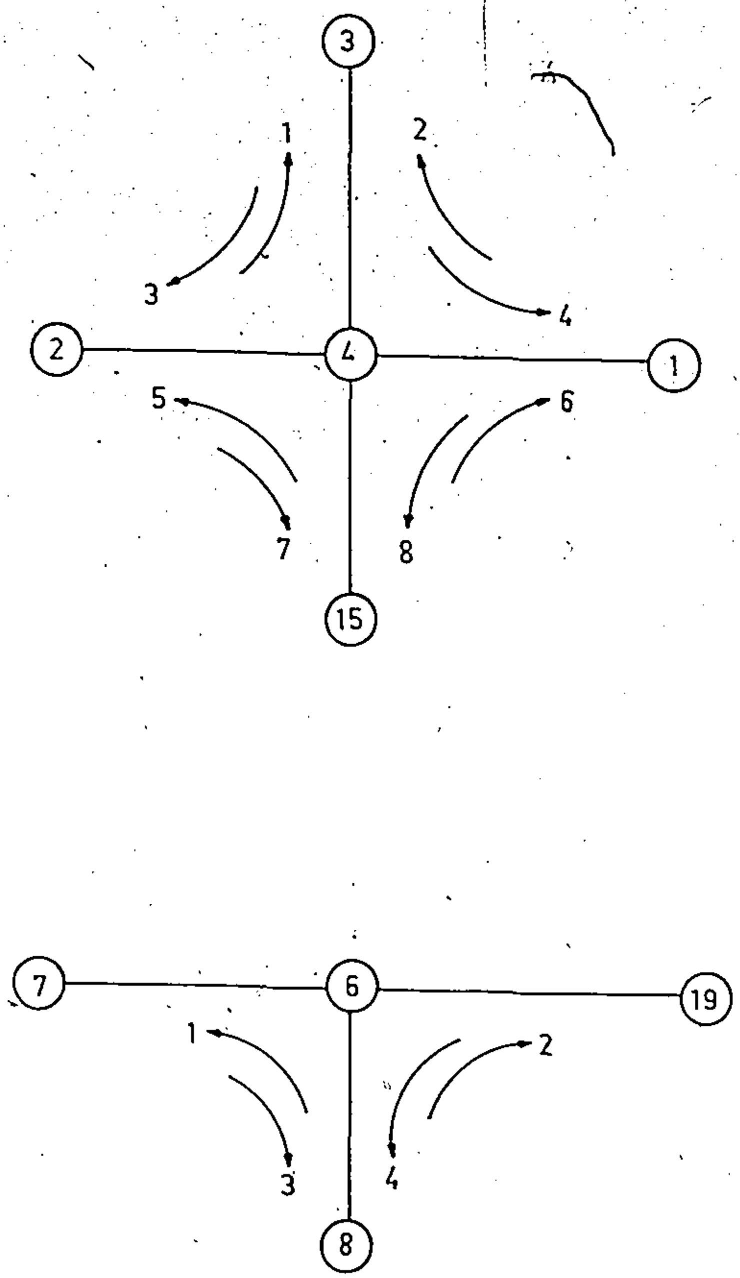


FIG. B-1 ILLUSTRATIVE EXAMPLE FOR TURNING VOLUME CALCULATION

The first two digits in NS1(J) represent the upstream node number of the turning movement while the second two digits represent the downstream node number of that movement. The turning movements are stored in the order of their printing as shown in Fig. B-1 by the numbers assigned to the turning movements. The first value 1 in kk(I) represents the position in NS1(J) of the first turning movement of the first intersection. The second value 9 represents the position in NS1(J) of the first turning movement of the second intersection. The value 13 is the total number of turning movements plus 1.

The upstream and downstream node numbers are stored in the same vector as an attempt to save computer storage. However, this procedure limits the number of nodes in the network to 99. For 100 or more nodes, a slight change is required in the program.

APPENDIX C

INSTRUCTIONS FOR PROGRAM USE

To carry out the different procedures contained in the model, a computer program was written in FORTRAN IV and provided in Appendix D. In this appendix a detailed description of the input data is provided in their order of input with some hints when necessary. An interpretation of the computer output is included.

C.1 Input Data

Following is a description of the input data in the order of their use in the computer program. There are two types of information requiring specifications; the initial data which is read only once at the beginning of the program and the time-slice data which is read for each time slice.

a) Initial Data

1- Diversion Control Parameter

1 card with format (I1). This is represented in the computer program by the parameter IDIV. If IDIV = 1, diversion occurs. If IDIV $\neq$ 1, no diversion occurs.

2- Integer Parameters

1 card with format (13I3) reads the following parameters in the given order:

NORG Number of orgins

NDEST Number of destinations

NNODE Number of nodes (includes NORG and NDEST)

NSLCE Number of time slices to be considered

NLINKS Number of links (dummy links included)

NMERGE Number of sections to be treated as merges

INTAP The lowest downstream node number of the dummy intersection approach links (these downstream nodes must be the highest numbered nodes in the network). For example, the value of INTAP in Fig. 3-2 is 37.

MIDEM Maximum number of iterations allowed in each time slice.

NIDEM Maximum number of iterations allowed to complete the sharing of merge capacity.

LIDEM Minimum number of iterations

NOMRG Parameter indicating whether merge capacities may be varied from one iteration to another (when NOMRG=0, they may be varied)

MINC Maximum number of assignment increments in an iteration. The computer program will stop if this value is exceeded.

IO Control parameter for printing out the intermediate assignment results (when IO=1, these results are output)

The network size allowed by the program is as follows:

- NLINKS $\leq$ 130
- NNODE $\leq$ 75

- NDEST $\leq$ 25
- NORG $\leq$ 25
- NMERGE $\leq$ 9
- NSLCE $\leq$ 9

Note: the number of turn prohibitions must not exceed 80 (this can be checked from the computer output of turn prohibitions)

For larger networks the dimensions must be changed in the vectors and matrices specified in the program initializing statements.

3- Real Parameters

1 card with format (6F3.2) reads in the following parameters in the given order:

SLICE Length of a time slice in hours (all time slices must have the same length)

DAMP Fraction of possible shared capacity in each increment

FRNO Fraction by which unit cost in previous slice is weighed into present slice

FRNN Fraction by which unit cost in previous iteration is weighed into present iteration

GROW Fraction by which the original O-D matrices are increased to give required demand level

R The constant term r in the diversion model

4- Number of Network Links Changed for Each Time Slice:

1 card with format (9I2). One value is given for each time slice. These are represented by the vector NLCHG (data for these links will be provided as des-

cribed in part b, item 2)

5- Number of Merges with Entitlement Changes for Each Time Slice

1 card with format (9I2). One value is given for each time slice. These are represented by the vector NMCHG (data for these merges will be provided as described in part b, item 3)

6- Number of Intersections for Which Turning Volumes to be Calculated for Each Time Slice

1 card with format (9I3). One value is given for each time slice. These are represented by the vector NINTP. The maximum number of intersections for each time slice is 10.

7- Origin Nodes

1 card with format (25I2). Identification numbers of network origin nodes. These are represented by the vector KO.

8- Destination Nodes

1 card with format (25I2). Identification numbers of network destination noedes. These are represented by the vector KD.

9- Number of Links out of each Node

1 card with format (75I1). The number of downstream links for each node treated in ascending order by node number. These are represented by the vector MA.

10- Link Descriptions

1 card is required for each link in the network in ascending order by link identification number. The

links are ordered by their upstream links numbers. Links having the same upstream node number are ordered by their downstream node numbers. If two links have the same upstream and downstream node numbers (dummy merging links or interchange exit ramps), they can be arranged in any order.

For each link I, the following information is read with format (3I3, 8F6.2, I1, 4I3) in the given order.

I	Link identification number
IB(I)	Upstream node of the link
L(I)	Downstream node of the link
AA1(I)	Capacity of 1st sublink component (vehicles per time slice)
AA2(I)	Capacity of 2nd sublink component (vehicles per time slice)
AA3(I)	Capacity of 3rd sublink component (vehicles per time slice)
CU1(I)	Unit cost of 1st sublink component (seconds)
CU2(I)	Unit cost of 2nd sublink component (seconds)
CU3(I)	Unit cost of 3rd sublink component (seconds)
PQC(I)	Physical queue capacity of the link. This value can be zero for some links, ie. dummy merging links.
FET(I)	Through flow equivalents of a unit of flow on the link
MAB(I)	The number of links immediately upstream of link I onto which the queue on I may spill.

LINKIN(I) Identification numbers of upstream links of link I onto which its queue may spill. A maximum of four links can be given.

11- Merge Descriptions

1 card is required for each merge I in the network. The following information on each merge is read in with format (13F6.0) in the following order:

MERGE (I,1) Identification number of one of the approach links of merge(I)

MERGE (I,2) Identification number of the other approach link of merge (I)

MERGE (I,3) Identification number of the downstream link of merge (I)

MERGE (I,4) The capacity entitlement of the link described as MERGE (I,1). This value is the capacity to discharge vehicles from MERGE (I,1) when MERGE (I,2) has a queue

MERGE (I,5) The capacity entitlement of the link described as MERGE (I,2). This value is the capacity to discharge vehicle from MERGE (I,2) when MERGE (I,1) has a queue

MERGE (I,6) The ultimate capacity of the link described as MERGE (I,1) when MERGE (I,2) has no flow. This is the capacity to accept vehicles to MERGE (I,1)

MERGE (I,7) The ultimate capacity of the link described as MERGE (I,2) when MERGE (I,1) has no flow. This is the capacity to accept vehicles to MERGE (I,2)

b) Data for Each Time Slice

For each time slice the following information is read in the given order.

1- Information for calculating turning volumes

1 card is required for each intersection J with format (5I3). The following information is read in the given order:

INT (J) Identification number of the intersection node

NSUR(J,4) Identification numbers of the surrounding nodes provided in clockwise direction. In the case of a T-intersection, the first number has to be zero and then the surrounding node numbers are given in the clockwise direction in the way described in Appendix B.

2- New Link information

1 card is read in for each link I whose characteristics to be changed in that time slice. The following information is read in with format (I3,6F6.2) in the given order:

I Link identification number

AA 1(I) Capacity of 1st sublink component (veh/time slice) cribed as MERGE (I,2) when MERGE (I,1) has no flow. This is the capacity to accept vehicles to MERGE (I,2)

b) Data for Each Time Slice

For each time slice the following information is read in the given order.

1- Information for calculating turning volumes

1 card is required for each intersection J with format (5I3). The following information is read in the given order:

INT (J) Identification number of the intersection node

NSUR(J,4) Identification numbers of the surrounding nodes provided in clockwise direction. In the case of a T-intersection, the first number has to be zero and then the surrounding node numbers are given in the clockwise direction in the way described in Appendix B.

2- New Link information

1 card is read in for each link I whose characteristics are to be changed in that time slice. The following information is read in with format (I3,6F6.2) in the given order:

I Link identification number

AA 1(I) Capacity of 1st sublink component (veh/time slice)

AA2(I) Capacity of 2nd sublink component (veh/ time slice)

AA3(I) Capacity of 3rd sublink component (veh/ time slice)

CU1(I) Unit cost of 1st sublink component (seconds)

CU2(I) Unit cost of 2nd sublink component (seconds)

CU3(I) Unit cost of 3rd sublink component (seconds)

The number of cards used in a time slice J must equal NLCHG(J).

3- New Merge Information

1 card is read in for each merge I to be altered in that time slice. The following information is provided with format (2I3, 2F6.2) in the given order:

I The identification number of the merge section

M1 The downstream link number of the merge section defined as MERGE (I,3)

X1 The new ultimate entitlement for the link defined as MERGE (I,1)

X2 The new ultimate entitlement for the link defined as MERGE (I,2)

The number of cards used in time slice J must be equal NMCHG(J).

4- O-D Matrix

The demands from each origin to ordered destinations are read in with Format (13F6.0). If NDEST $\leq$ 13, only 1 card is required for each origin. If NDEST > 13, use two cards.

C.2 Computer Output

Generally, two types of information are output, namely initial output and time slice-related output.

a) Initial Output

This type of output is given once at the beginning of the computer output. It includes all information which is common for all time slices. In that part, the following is given:

1- Definition of the program and summary of the main input parameters.

2- Summary of turn prohibitions in the network automatically determined by the program. This list of turn prohibitions will help the analyst to check them against those required for his network. If inconsistency occurs, this means something is incorrect in the data he input for the upstream links feeding each link. Furthermore, he can ensure that the maximum number of turn prohibitions (80) is not exceeded.

3- Output of original link information including capacity, cost and upstream links onto which queues may spill.

4- Output of original merge information including entitlement and ultimate capacities for each merge approach.

b) Time Slice-Related Output

This type of output is repeated for each time slice. It includes the four main parts described below:

1- Output of new link and merge characteristics

If the user changes any characteristics of network links or merges, this new information will be printed out at the beginning of that time slice. This includes any ramp metering rates considered in that time slice. It should be noted that if the analyst specifies any new information for links or merges in a certain time slice, this new information and not the original one will be used in the following time slice unless this information is changed in that time slice.

2- Increment-related output

This part of output is optional and may be used to print information after each assignment increment. This information includes increment number and cumulative assigned fraction of the O-D matrix. Also, the number of the critical link is printed along with the attempted and allowable assignments of that critical link. The critical link is that one with the largest ratio of attempted to allowable assignment in that increment. When this ratio is less than or equal 1.0 the iteration ends.

3- Iteration-related output

This output provides information on each network freeway and entrance ramp approach. This includes the flow assigned to each approach (veh/time slice) and the queue size (vehicles) formed on and upstream of each approach at the end of iteration.

4- Final output

The following information is printed out at the end of each time slice:

i) link-flow characteristics

For each link in the network the flow (vehicles/time slice) and travel costs (seconds) are printed. Each line in that output specifies a group of links having the same upstream node. These links are identified by that upstream node and their downstream nodes. The upstream nodes are printed in ascending order. The cumulative total cost (seconds) is printed after each group of links. When the upstream node is a congested point, the queued demand from that node to each destination is printed with the resulting cumulative total cost (seconds). It should be noted that this total cost is accumulated for time slices.

ii) Links having queues

The network links having queues after each time slice are given with the queue size (vehicles) on each link. Each link is identified by its upstream and downstream node numbers.

iii) Links reaching capacity

The network links reaching capacity after each time slice are given with the corresponding capacity. Each link is identified by its upstream and downstream node numbers. The capacity is expressed in vehicles per time slice.

iv) Turning volumes at specified intersections

For each intersection specified in that time slice the intersection node number and the surrounding node numbers are printed along with the intersection legs and turning arrows. The turning volumes (veh/time slice) are printed on the corresponding arrows.

At the end of final time slice the total cost (veh-hours) for all time slices is printed.

c) Messages of the program

The program stops if any of the following occurs:

1- The links numbers are out of order. The program prints this message: LINK INPUT OUT OF SEQUENCE

2- The upper limit of the number of increments is exceeded. The program prints the following message: INCREMENT LIMIT EXCEEDED

3- The number of links stored in the minimum path table exceeds 195 -- the program prints: LAST (IN TREE) = (the exceeding value) -INCREASE DIMENSION OF TABLE

4- The destination is not reached. The message is: NO PATH FROM (origin no.) TO (dest. no.) - LEFT DEMAND = (left demand in veh/time slice)

5- The number of upstream links feeding any turn prohibition node exceeds 4. The message is: NO. OF LINKS ENTERING THE TURN PROHIBITION NODE (node no.) MUST NOT EXCEED 4.

APPENDIX D

PROGRAM CORCON = A MODEL FOR INVESTIGATING TRAFFIC ASSIGNMENT AND CONTROL IN A FREEWAY CORRIDOR

To carry out the logical sequences of the model described in Chapter 4, a computer program named CORCON was written in FORTRAN IV language with a portion of Yagar's program [20] incorporated.

The program CORCON consists of the main executive program and the three subroutines MINPATH, FCONPT, and ASSIGN. MINPATH performs the forward procedure of the minimum path algorithm. FCONPT performs the backward procedure and determines the first congested point in the path. The assignment to minimum paths and calculation of turning volumes are carried out by ASSIGN.

Definitions and dimensions of the important variables used and a complete listing of the computer program are given in this appendix.

D.1 Program Variables

The main variables contained in the computer program are given below. Capacities are in vehicles per time slice and costs are in seconds.

A (130) tentative assignment to the link up to present time in the increment.

AA(130) allowable assignment to the link at present cost

AA1(130)	capacity of the first component of the link
AA2(130)	capacity of the second component of the link
AA3(130)	capacity of the third component of the link
AQ(100, 25)	cumulative queue to the present time in the iteration
AS(130)	assignment to the link to the present time in the iteration
ATQ(130)	physical queue tentatively on the link up to present time in the increment
BBEG(9)	fraction of O-D matrix assigned in that iteration when capacity borrowing started
C(75)	minimum cost of reaching the node in the first minimum path
C1(79)	minimum cost of reaching the node in the second minimum path
CAP(130)	capacity of the link = AA1 + AA2 + AA3
CC(75, 4)	the four upstream link costs of turn prohibition nodes considered in the first minimum path
CC1(75,4)	the four upstream link costs of turn prohibition nodes considered in the second minimum path
CSQ(130)	capacity to serve queued vehicles on the link
CU(130)	unit cost of the link used in the early tree
CU1(130)	unit cost of the first sublink component
CU2(130)	unit cost of the second sublink component
CU3(130)	unit cost of the third sublink component
D(100,25)	O-D left from each node to all destinations

DAMP	predetermined fraction of possible shared capacity in each increment
FET(130)	flow equivalent of a unit of flow on the link
FRN	amount by which previous unit costs are weighed
FRNO	amount by which unit cost of previous time slice is weighed into present slice
FRNN	amount by which unit cost of previous iteration is weighed into present iteration
GROW	fraction by which original O-D demand is increased to give required demand level
IB(130)	upstream node for the link
IC	number of queueing links in the first minimum path from origin to a given destination
ICT	number of queueing links in the network
ICOMP(130)	sublink component being used at present for that link
IDEM	number of iterations completed
IDIV	diversion control parameter. IF equals 1, diversion is considered
INT(10)	intersection node number for which turning volumes are to be calculated in that time slice
IPP	number of increments up to the present time in the iteration
ISLCE	number of present time slice
ITREE(100)	nodes from which minimum paths are built

IV	number of links to be considered for completion of minimum paths
JC	first congested point in the first minimum path
JC1	first congested point in the second minimum path
KD(25)	destination node members
KK(11)	position of first turning movement of each intersection in vector NS1(100)
KO(25)	origin node numbers
LAST	the number of links whose terminal nodes have to be considered for completion of minimum paths
L(130)	downstream node for the link
LINK(75)	link upstream of that node corresponding to its minimum cost in the present tree of the first minimum paths
LINK1(75)	link upstream of that node corresponding to its minimum cost in the present tree of the second minimum path
LINKC(130)	queueing link numbers in the network
LINKD(80)	downstream link number of turn prohibition nodes
LINKK(75, 4)	the four upstream link numbers of turn prohibition nodes considered in the first minimum path
LINKK1(75, 4)	the four upstream link numbers of the turn prohibition nodes considered in the second

	minimum path
LINKIN(130, 4)	upstream links of that link onto which queue may spillback
LINKU(80)	upstream link number of turn prohibition nodes
LIDEM	minimum number of iterations
MA(75)	number of links out of that node
MAB(130)	number of upstream links, feeding that link, onto which queue may spillback
MAT(75)	identification number of first link out of that node
MERGE(9, 13)	characteristics of each merging section I (as described below)
MERGE(I, 1)	first merging link number of merge I - approach 1, say
MERGE(I, 2)	second merging link number of merge I - approach 2, say
MERGE(I, 3)	downstream merging link number of merge I
MERGE(I, 4)	capacity entitlement of approach 1, used in the present iteration
MERGE(I, 5)	capacity entitlement of approach 2, used in the present iteration
MERGE(I, 6)	ultimate capacity of approach 1
MERGE(I, 7)	ultimate capacity of approach 2
MERGE(I, 8)	total merge capacity at the beginning of the iteration
MERGE(I, 9)	total merge capacity at the existing weaving flows

MERGE(I, 10)	ultimate entitlement at merge approach 1
MERGE(I, 11)	ultimate entitlement at merge approach 2
MERGE(I, 12)	queue on and upstream of approach 1 at the end of the previous iteration
MERGE(I, 13)	queue on and upstream of approach 2 at the end of the previous iteration
MINC	maximum number of increment in an iteration
MIDEM	mazimum number of iterations in each time slice
NAV(50)	link numbers out of O-D nodes in the network
NDEST	number of destinations
NIDEM	maximum number of iterations that allow sharing of merge capacity
NINTP(9)	number of intersections for each time slice for which turning volumes are to be calculated
NLINKS	number of links
NMERGE	number of merging sections (≯ 9)
NNODE	number of nodes
NOD(75)	storage of all network node numbers with turn prohibition nodes having negative signs
NOMRG	control parameter indicates whether merge capacity entitlement may be varied (if equals 0, they may be varied)
NORG	number of origins
NP(75)	the node upstream of that node, which corresponds to its minimum cost in the early tree in the first minimum path

NP1(75)	the node upstream of that node, which corresponds to its minimum cost in the early tree in the second minimum path
NS1(100)	upstream and downstream node numbers of turning movements that are to be calculated
NSLCE	number of time slices
NSUR(10, 4)	the surrounding node numbers of each intersection whose turning volumes are to be calculated.
NTPRO	total number of turn prohibitions
NTPRO1	number of turn prohibitions except those for merging sections
OCU(130)	unit cost of the link at the end of previous iteration
P1	percentage of demand diverted from the first minimum path to the second minimum path in an increment
PP	inverse of fraction of remaining O-D matrix that can be assigned in present increment
PPT	cumulative fraction of the O-D matrix that has been assigned
PQC(130)	physical queue capacity of that link
Q(100, 25)	tentative queue to the present time in the increment
QCOST(130)	unit queue cost of queueing on that link estimated after each increment
R	diversion constant in the diversion model
S	remaining fraction of O-D matrix

SLICE	length of time slice (hours)
SY(25)	dummy vector for reading demands that origin to all destinations
TABLE(2,200)	matrix with two rows containing links to be considered for completion of minimum paths
TC	accumulated total cost spent in the network
TQ(130)	cumulative physical queue on the link up to the present time in the iteration
TTC(25)	total queueing time (seconds) in the minimum path to that destination
TURV(100)	tentatively assigned turning volume so far in the increment
TURV1(100)	cumulative assigned turning volume up to the present time in the iteration
V	minimum equivalent queueing cost of traffic diverted from the first minimum path to the second minimum path
Z	number of links out of all O-D nodes
ZLINK(100)	identification number of the link on which queue is formed in this time slice. This is used to avoid time slices-related illogical paths
ZLINKO(100)	identification number of the link to be considered out of the source for purposes of assigning queued demand in this time slice.
ZO(25)	upstream node number of the link on which queue is formed.

D.2 COMPUTER PROGRAM LISTING

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C   INITIALIZATION ROUTINE
C   **********************
C
      REAL         PPY,S,
     *             A(130),AA(130),AA1(130),AA2(130),AA3(130),AQ(100,25),
     *             ATQ(130),BBFG(9),C(75),CSQ(130),CU2(130),CU3(130),
     *             OCU(130),MERGE(9,13),Q(100,25),SY(25)
C
      INTEGER      NDEST,NLINKS,NNODE,NORG,SOURCE,Z,Z1,Z2,Z3,Z4,Z5,
     *             ICOMP(130),ITREE(100),KD(25),KO(25),LINK(75),MAP(130),
     *             NLCHG(9),NMCHG(9),NP(75),ZLINK(100),ZLINKO(100),ZO(25)
C
      DIMENSION    AS(130),C1(75),CAP(130),CC(75,4),CC1(75,4),CU(130),
     *             D(100,25),LINK1(130),LINKC(130),LINKK(75,4),LINKK1(75,4)
     *             ,MLD(20),MLU(20),            NINTP(9),NODUL(5),NP1(75),
     *             PQC(130),QCOST(130),TQ(130),TTC(25),TURV(100)
C
      COMMON       INTAP,JCI,NH,NINT,NNODE,NTPRO,NTPRO1,NLINKS,NR,Z,
     *             CU1(130),FET(130),IB(130),INT(10),KK(11),L(130),
     *             LINKIN(130,4),LINTD(80),LINTU(80),MA(75),MAT(75),NAV(50)
     *             ,NOD(75),NS1(100),NSUR(10,4),TURV1(100),NRC(5),ZLINKO
C
      DATA TC,AQ/2501*0.0/,ISLCE/1/,ZLINK/100*0/
C
C   INPUT DATA
C   **********
C
C   CONTROL CARD FOR DIVERSION CONSIDERATION
      READ(5,558) IDIV
C   INTEGER VALUES
      READ 82,NORG,NDEST,NNODE,NSLCE,NLINKS,NMERGE,INTAP,MIDEM,NIDEM
     /,LIDEM,NOMRG,MINC,IO
C   REAL VALUES
      READ 84,SLICE,DAMP,FRNO,FRNN,GROW,R
C   NUMBER OF LINK DESCRIPTION CHANGES FOR EACH TIME SLICE
      READ 80,(NLCHG(I),I=1,NSLCE)
C   NUMBER OF MERGE DESCRIPTION CHANGES FOR EACH TIME SLICE
      READ 80,(NMCHG(I),I=1,NSLCE)
C READING NO. OF INT. TO BE PLOTTED FOR EACH TIME SLICE.
      READ(5,82)(NINTP(I),I=1,NSLCE)
C   ORIGIN NODE NUMBERS
      READ 80,(KO(I),I=1,NORG)
C   DESTINATION NODE NUMBERS
      READ 80,(KD(I),I=1,NDEST)
C   NUMBER OF LINKS OUT OF EACH NODE
      READ 83,(MA(I),I=1,NNODE)
```

```
C  READING LINK INFORMATION --IN ASCENDING ORDER BY LINK NUMBER
      DO 849 IX=1,NLINKS
      READ 87,LL,
     /        IB(IX),L(IX),AA1(IX),AA2(IX),AA3(IX),CU1(IX),
     /CU2(IX),CU3(IX),PGC(IX),FET(IX),MAB(IX),(LINKIN(IX,J),J=1,4)
      IF(LL.NE.IX) GO TO 62
      CAP(IX)=AA1(IX)+AA2(IX)+AA3(IX)
      IF(POC(IX).LT..0001) PGC(IX)=.0001
      IF(CU2(IX).LT.CU1(IX)) CU2(IX)=CU1(IX)
      IF(CU3(IX).LT.CU2(IX)) CU3(IX)=CU2(IX)
      OCU(IX)=0.
  849 CONTINUE
C  DESCRIPTION OF EACH 2 TO 1 LINK MERGE SECTION
      DO 1194 JL=1,NMERGE
      READ 86,  (MERGE(JL,IM),IM=1,7)
 1194 CONTINUE
      GO TO 70
   62 PRINT 63
      STOP
   90 PRINT 81
      STOP
C  PRINT SUMMARY OF THE MAIN INPUT PARAMETERS
   70 PRINT 109
      WRITE(6,199)
      PRINT 179,IDIV, NOMRG ,DAMP,
     /NSLCE,SLICE,NLINKS,NNODE,NORG,NDEST,      FRNO,FRNN,GROW,R
C
C  INITIAL PREPARATION FOR ALL TIME SLICES
C  ************************************
C
C  CALCULATING IDENTIFICATION NUMBER OF FIRST LINK OUT OF EACH NODE
      MAT(1)=1
      DO 99 JA=2,NNODE
      MAT(JA)=MAT(JA-1)+MA(JA-1)
   99 CONTINUE
C  PREPARE A VECTOR REQUIRED FOR AVOIDING ILLIGICAL PATHS THROUGH
C  O-D NODES
      Z=0
      DO 37 I=1,NORG
      DO 7 J=1,NDEST
      IF(KO(I).EQ.KD(J)) GO TC 27
      GO TO 7
   27 Z1=KO(I)
      Z2=MAT(Z1)
      Z3=MA(Z1)
      DC 34  II=1,Z3
      Z=Z+1
   34 NAV(Z)=Z2+II-1
```

```
      7 CONTINUE
     37 CONTINUE
C   PREPARE AUTOMATIC TURN PRCHIBITIONS
      I11=0
      NTPRO1=0
      DO 853 IX=1,NNODE
      DO 94 I=1,5
   94 NODUL(I)=0
      IV=0
      DO 2 K1=1,NDEST
      IF(IX.EQ.KD(K1)) GO TO 853
    2 CONTINUE
      LINO=MAT(IX)
      LINKE=MA(IX)+LINO-1
      DO 852 J=LINO,LINKE
      DO 854 J1=1,4
      M=LINKIN(J,J1)
      IF(M.EQ.0) GO TO 854
      DO 95 I=1,5
      IF(M.EQ.NODUL(I)) GO TO 854
   95 CONTINUE
      IV=IV+1
      IF(IV.GT.5) PRINT 195,IX
      NODUL(IV)=M
  854 CONTINUE
  852 CONTINUE
      DO 860 J2=1,5
      IF(NODUL(J2).EQ.0) GO TC 853
      DO 861 J=LINO,LINKE
      DO 862 J1=1,4
      M=LINKIN(J,J1)
      IF(M.EQ.NODUL(J2)) GO TC 861
  862 CONTINUE
      IF(J .EQ.1) GO TO 105
      IF(J .EQ.NLINKS) GO TO 106
  105 IF(IB(J).EQ.IB(J+1).AND.L(J).EQ.L(J+1)) GO TO 855
      IF(J.EQ.1) GO TO 107
  106 IF(IB(J).EQ.IB(J-1).AND.L(J).EQ.L(J-1)) GO TO 855
  107 NTPRO1=NTPRO1+1
      LINTU(NTPRO1)=NODUL(J2)
      LINTD(NTPRO1)=J
      GO TO 861
  855 I11=I11+1
      MLU(I11)=NODUL(J2)
      MLD(I11)=J
  861 CONTINUE
  860 CONTINUE
  853 CONTINUE
```

```
      NTPRO=NTPRO1
      IF(I11.EQ.0) GO TO 859
      DO 857 J=1,I11
      NTPRO=NTPRO+1
      LINTU(NTPRO  )=MLU(J)
      LINTD(NTPRO  )=MLD(J)
  857 CONTINUE
  859 DO 307 I=1,NNODE
  307 NOD(I)=I
      IF(NTPRO.EQ.0) GO TO 85
      DO 303 I=1,NTPRO
      K4=LINTU(I)
      K5=L(K4)
      IF(NOD(K5).LT.0) GO TO 303
      NOD(K5)=-(K5*100+I)
  303 CONTINUE
C  PRINT TURN PROHIBITIONS
      WRITE(6,570)
      DO 580 I=1,NTPRO
      J1=LINTU(I)
      J2=LINTD(I)
      IF(I.GT.NTPRO1) GO TO 581
      WRITE(6,572)I,IB(J1),L(J1),IB(J2),L(J2)
      GO TO 580
  581 WRITE(6,575)I,IB(J1),L(J1),IB(J2),L(J2)
  580 CONTINUE
      WRITE(6,573)
C  PRINT LINK INFORMATION
      PRINT 180
      PRINT 197
      DO 1974 NA=1,NLINKS
      PRINT 198, NA,IB(NA),L(NA),AA1(NA),AA2(NA),AA3(NA),CU1(NA),CU2(NA)
     1,CU3(NA),(LINKIN(NA,J),J=1,4)
 1974 CONTINUE
C  PRINT ORIGINAL MERGE INFORMATION
      WRITE(6,200)
      DO 250 JL=1,NMERGE
      WRITE(6,201)MERGE(JL,1),MERGE(JL,4),MERGE(JL,6)
      WRITE(6,202)JL,MERGE(JL,3)
      WRITE(6,203)MERGE(JL,2),MERGE(JL,5),MERGE(JL,7)
  250 CONTINUE
C
C  THE BEGINNING OF A TIME SLICE.
C  *****************************
C
      Z4=0
   85 IDEM=0
```

```
      FRN=FRNO
      WRITE(6,185)ISLCE
C CHECK IF ANY INTERSECTION TC BE PLOTTED IN THIS TIME SLICE
      NINT=NINTP(ISLCE)
      N1=0
      IF(NINT.EQ.0) GO TO 123
      DO 127 MI=1,NINT
  127 READ(5,563)INT(M1),(NSUR(M1,I),I=1,4)
C  PREPARE VECTORS FOR TURNING VOLUMES PLOTTING
      DO 120 I=1,NINT
      L1=NSUR(I,1)
      L2=NSUR(I,2)
      L3=NSUR(I,3)
      L4=NSUR(I,4)
      KK(I )=N1+1
      IF(L1.EQ.0) GO TO 121
      N1=N1+8
      NS1(N1-7)=L4*100+L1
      NS1(N1-6)=L2*100+L1
      NS1(N1-5)=L1*100+L4
      NS1(N1-4)=L1*100+L2
  121 IF(L1.NE.0) GO TO 122
      N1=N1+4
  122 NS1(N1-3)=L3*100+L4
      NS1(N1-2)=L3*100+L2
      NS1(N1-1)=L4*100+L3
      NS1(N1)=L2*100+L3
  120 CONTINUE
      KK(1+NINT)=N1+1
C  ALTERING ANY LINK CHARACTERISTICS IN THIS TIME SLICE
  123 N=NLCHG(ISLCE)
      WRITE(6,205)
      IF(N.EQ.0) GO TO 251
      WRITE(6,213)
      DO 8 I=1,N
      READ 103,M,X1,X2,X3,X4,X5,X6
      WRITE(6,198)M,IB(M),L(M),X1,X2,X3,X4,X5,X6
      AA1(M)=X1
      AA2(M)=X2
      AA3(M)=X3
      CAP(M)=AA1(M)+AA2(M)+AA3(M)
      CU1(M)=X4
      CU2(M)=X5
    8 CU3(M)=X6
      GO TO 3
  251 IF(ISLCE.EQ.1) GO TO 253
      WRITE(6,207)
      GO TO 3
  253 WRITE(6,206)
```

```
C  ALTERING ANY MERGE CHARACTERISTCS IN THIS TIME SLICE
   3 N=NMCHG(ISLCE)
     WRITE(6,208)
     IF(N.EQ.0) GO TO 252
     WRITE(6,214)
     NR=0
     DO 4  I=1,N
     READ 104,M,M1,X1,X2
     IF(X1.NE.0.) GO TO 22
     NR=NR+1
     NRC(NR)=MERGE(M,1)
     GO TO 38
  22 IF(X2.NE.0.) GO TO 38
     NR=NR+1
     NRC(NR)=MERGE(M,2)
  38 WRITE(6,201)MERGE( M,1),X1           ,MERGE( M,6)
     WRITE(6,202) M,MERGE( M,3)
     WRITE(6,203)MERGE( M,2),X2           ,MERGE( M,7)
     IF(MERGE(M,3).NE.M1) PRINT 63
     MERGE(M,4)=X1
   4 MERGE(M,5)=X2
     WRITE(6,215)
     GO TO 9
 252 IF(ISLCE.EQ.1) GO TO 254
     WRITE(6,210)
     GO TO 9
 254 WRITE(6,209)
   9 DO 1295 JL= 1,NMERGE
     CAP(MERGE(JL,1))=MERGE(JL,4)
     AA1(MERGE(JL,1))=MERGE(JL,4)
     CAP(MERGE(JL,2))=MERGE(JL,5)
     AA1(MERGE(JL,2))=MERGE(JL,5)
C  EACH MERGE HAS NO WEAVING TO BEGIN.  VARIATION IN ITS DOWNSTREAM LINK≠S
C  CAPACITY IS BORNE BY ITS HIGHEST SUBLINK COMPONENTS.
     MERGE(JL,8)=MERGE(JL,4)+MERGE(JL,5)
     MERGE(JL,9)=MERGE(JL,8)
     MERGE(JL,10)=MERGE(JL,4)
     MERGE(JL,11)=MERGE(JL,5)
     MERGE(JL,12)=0.
     MERGE(JL,13)=0.
     I=MERGE(JL,3)
     CAP(I)=MERGE(JL,8)
     AA3(I)=CAP(I)-AA1(I)-AA2(I)
     IF(AA3(I).GE.0.) GO TO 1295
     AA3(I)=0.
     AA2(I)=CAP(I)-AA1(I)
     IF(AA2(I).GE.0.) GO TO 1295
     AA2(I)=0.
```

```
      AA1(I)=CAP(I)
 1295 CONTINUE
C  SETTING THE DEMAND MATRIX EQUAL TO THE QUEUED DEMAND
      NNODE1=NNODE+25
      DO 23 I =1,NNODE1
      ZLINKO(I)=ZLINK(I)
      DO 23 J=1,NDEST
   23 D(I,J)=AQ(I,J)
C  ADDING ON THE DEMAND MATRIX FOR THE NEW TIME SLICE.
      DO  25 IX=1,NORG
      READ 86,    (SY(J),J=1,NDEST)
      II=KO(IX)
      DO 25 J=1,NDEST
   25 D(II,J)=SY(J) *(1.+GROW)+ D(II,J)
C  RECORDING ACTIVE SOURCE NODES FOR THIS TIME SLICE IN A VECTOR
      DO 13 JJ=1,NNODE1
   13 ITREE(JJ)=0
      MN=NNODE+Z4
      DO 15 I=1,MN
      DEMAND=0.0
      DO 11 J=1,NDEST
   11 DEMAND=DEMAND+D(I,J)
      IF(DEMAND.LT.0.1)GO TO 15
      IF(I.LE.NNODE) GO TO 126
      MM=ZO(I-NNODE)
      ITREE(I)=MM
      GO TO 15
  126 ITREE(I)=I
   15 CONTINUE
      IF(IO.EQ.0) GO TO 255
      WRITE(6,212)
      GO TO 21
  255 WRITE(6,211)
C
C  ROUTINE FOR EACH ITERATION
C  ************************
C
   21 DO 24 I=1,NNODE1
      ZLINK(I)=0
      DO 24 J=1,NDEST
   24 AQ(I,J)=0.
      Z4=0
      IDEM=IDEM+1
      PRINT 186, IDEM
      IF(IO.EQ.0) GO TO 108
      PRINT 182
      PRINT 183
```

```
  108 IPP=0
      PPY=0.
      S=1.
      DO 41 I=1,NMERGE
   41 BBEG(I)=0.
C  BEGINNING WITH FIRST SUBLINK COMPONENTS AND NO ASSIGNED DEMANDS
      DO 71 IX=1,NLINKS
      ICOMP(IX)=1
      CU(IX)=CU1(IX)
      AA(IX)=AA1(IX)
      TQ(IX)=0.
      CSQ(IX)=CAP(IX)
      QCOST(IX)=0.0
   71 AS(IX)=0
      IF(N1.EQ.0) GO TO 10
      DO 125 I=1,N1
  125 TURV1(I)=0.
      ICT=0
C
C  BEGINNING OF THE ROUTINE FOR AN INCREMENT IN THE ASSIGNMENT
C  ****************************************************************
C
   10 IPP=IPP+1
      DO 14 I=1,NNODE1
      DO 14 J=1,NDEST
   14 Q(I,J)=0.
C  CALCULATION OF UNIT COSTS TO USE IN FINDING MIN. COST PATHS
      DO 301 IX=1,NLINKS
      CU(IX)=CU(IX)*(1.-FRN)+CCU(IX)*FRN
      A(IX)=0.
  301 ATQ(IX)=0.
      IF(N1.EQ.0) GO TO 26
      DO 124 I=1,N1
  124 TURV(I)=0.
C  TENTATIVELY ASSIGNING ALL REMAINING DEMAND FROM EACH SOURCE NODE --DOWN
C  TO STATEMENT 229
   26 DO 229 NH=1,MN
      SOURCE=ITREE(NH).
      IF(SOURCE.EQ.0) GO TO 229
C  INITIALISE NODE LABELS
      DO 35 NODE=1,NNODE
      C(NODE)=9999.
      IF(NOD(NODE).GT.0) GO TO 35
      DO 304 J=1,4
      LINKK(NODE,J)=0
  304 CC(NODE,J)=9999.
   35 CONTINUE
```

```
      C(SOURCE)=0.0
      Y2=0.
      CALL MINPATH (SOURCE,C,LINK,NP,LINKC,ICT,CU,Y2,CC,LINKK)
C  ROUTINE FOR EACH DESTINATION NODE--DOWN TO STATEMENT 22
      IF(IDIV.EQ.0) GO TO 1000
      IF(ICT.EQ.0) GO TO 1000
      DO 36 NODE=1,NNODE
      C1(NODE)=9999.
      IF(NOD(NODE).GT.0) GO TO 36
      DO 305 J=1,4
      LINKK1(NODE,J)=0
  305 CC1(NODE,J)=9999.
   36 CONTINUE
      C1(SOURCE)=0.0
      Y2=1.
      CALL MINPATH (SOURCE,C1,LINK1,NP1,LINKC,ICT,CU,Y2,CC1,LINKK1)
 1000 DO 15000 IY=1,NDEST
      IF(D(NH    ,IY)*S.LE.0.1) GO TO 15000
      II=KD(IY)
      IF(II.EQ.SOURCE) GO TO 15000
      IF (C(II).GE. 9998.)  GO TO 100
      TTC(IY)=0.
      CALL FCONPT (II,NP,LINK,TTC,IC,IY,JC,SOURCE,QCOST,
     /AS,CAP,TQ,PQC,    CU,CC,LINKK,C,Z5)
      IF(IDIV.EQ.0) GO TO 15002
      IF(IC .EQ.0) GO TO 15002
      IF(C1(II).GE.9998.) GO TO 15002
      CALL FCONPT (II,NP1,LINK1,TTC,IC,IY,JC1,SOURCE,QC
     /OST,AS,CAP,TQ,PQC,    CU,CC1,LINKK1,C1,ZP1)
C CALCULATION OF PERCENTAGE DIVERTED
      IF(TTC(IY).LT.1.)TTC(IY)=1.
      V=1.+(C1(II)-C(II))/TTC(IY)
      P1=1./V**R
      GO TO 15003
15002 P1=0.
15003 IF(P1.EQ.1.0) GO TO 15006
        IF(JC.EQ.II) GO TO 131
C  STORAGE OF QUEUES AND PREPARATION FOR TIME SLICES -RELATED ILLOGICAL
C  PATHS
      IF(JC.EQ.SOURCE) GO TO 110
      ML=NNODE+1
      NL=NNODE+24
      IF(Z4.EQ.0) GO TO 140
      DO 130 J=ML,NL
      IF(ZLINK(J).NE.Z5) GO TO 130
      Q(J,IY)=Q(J,IY)+D(NH    ,IY)*S*(1.-P1)
      GO TO 131
```

```
  130 CONTINUE
  140 Z4=Z4+1
      ZO(Z4)=JCI
      ZLINK(NL+1)=Z5
      Q(NNODE+Z4,IY)=Q(NNODE+Z4,IY)+D(NH     ,IY)*S*(1.-P1)
      IF(ZLINKO(NH).EQ.0) GO TO 131
      IF(L(ZLINKO(NH)).EQ.JC) GO TO 15005
      GO TO 131
  110 Q(JC ,IY)=Q(JC ,IY)+D(NH     ,IY)*S*(1.-P1)
      GO TO 15005
  131 X=1.-P1
C TENTATIVELY STORE QUEUED DEMAND ,ASSIGNQUEUES AND FLOWS TO LINKS IN THE
C FIRST MIN. PATH
      CALL ASSIGN (JC,LINK,II,ATQ,A,IY,NP,SOURCE,X,D,S,TURV,PQC,QCOST)
15005 IF(P1.EQ.0.) GO TO 15000
C  TENTATIVELY ASSIGN FLOWS TO LINKS IN THE 2ND MINIMUM PATH
15006 X=P1
      CALL ASSIGN(JC1,LINK1,II,ATQ,A,IY,NP1,SOURCE,X,D,S,TURV,PQC,QCOST)
15000 CONTINUE
      GO TO 229
  100 D1=D(NH     ,IY)*S
      PRINT 102,SOURCE,KD(IY),D1
      STOP
  229 CONTINUE
C
C  FINDING THE CRITICAL LINK WHICH LIMITS THE FRACTION OF THE WHOLE
C  DEMAND MATRIX THAT CAN BE ASSIGNED.--FINDING THIS FRACTION.
C
      PP=0.
      DO 409 IBEGIN=1,NLINKS
      IF(A(IBEGIN).LT..0001) GO TO 409
      P=A(IBEGIN)/AA(IBEGIN)
      IF (P .LT. PP) GO TO 409
      PP=P
      IBAST=IBEGIN
      AXE=A(IBEGIN)
      BXE=AA(IBEGIN)
  409 CONTINUE
      IF(PP.LE.1.0)PP=1.
      IF(PP.GT.1000.) PP=1000.
C  INCREMENT QUEUED DEMANDS
      DO 45 I=1,NNODE1
      DO 45  J=1,NDEST
   45 AQ(I,J)=AQ(I,J)+Q(I,J)/PP
C  INCREMENTING ASSIGNMENTS OF FLOWS AND QUEUES TO LINKS
      DO 48 IA=1,NLINKS
      TQ(IA)=TQ(IA)+ATQ(IA)/PP
```

```
 48 AS(IA)=AS(IA)+A(IA)/PP
    IF( N1.EQ.0) GO TO 414
    DO 129 I=1, N1
129 TURV1(I)=TURV1(I)+TURV(I)/PP
414 IF (IPP.GE.MINC) GO TO 90
C  CALCULATE CUMULATIVE FRACTION OF O-D,S ASSIGNED
    S=S*(PP-1.)/PP
    IF(S.GT..001) GO TO 302
    S=0.
    PP=1.
302 PPY=1.D0-S
    IF(IO.EQ.0) GO TO 101
    PRINT 91,IPP,PPY, IBAST,AXE,BXE
C ***
C  WEAVE ROUTINE
C ***
C
C  MERGE ROUTINE FOR SHARING OF MERGE CAPACITY AMONG ITS APPROACHES
C
101 IF(IDEM.GT.NIDEM) GO TO 12
    DO 10000 MC=1,NMERGE
    IF(PPY.EQ.0.) GO TO 10000
    L1=MERGE(MC,1)
    L2=MERGE(MC,2)
    L3=MERGE(MC,3)
    IF(AS(L1).LT.    CAP(L1).AND.AS(L2).LT.    CAP(L2))GO TO 10000
    IF(AS(L1).GE.    CAP(L1).AND.AS(L2).GE.    CAP(L2))GO TO 10000
    IF(BBEG(MC).EQ.0..AND.PPY.NE.1.) BBEG(MC)=PPY
    IF(AS(L1).GE.    CAP(L1))GO TO 10003
C  L1 SHARES ITS MERGE CAPACITY ENTITLEMENT WITH L2
C  RESERVE SOME CAPACITY BEFORE SHARING
    RES=AS(L1)*(1.-PPY)/PPY
    RES=RES+MERGE(MC,12)*(1.-PPY)/(1.-BBEG(MC))
    F=AS(L1)+RES
    RATIO=MERGE(MC,9)/MERGE(MC,8)
C  IF NOTHING TO SHARE, SKIP IT
    IF(F.GT.MERGE(MC,4)*RATIO) GO TO 10000
C  CALCULATE AMOUNT WHICH COULD BE GIVEN TO L2
    EXCESS=RATIO*(MERGE(MC,7)+F/MERGE(MC,4)*(MERGE(MC,5)-MERGE(MC,7)))
   /-AS(L2)
    IF(EXCESS.LE.0.) GO TO 10000
    CAP(L2)=CAP(L2)+EXCESS*CAMP
    AA1(L2)=CAP(L2)
    CAP(L1)=RATIO*MERGE(MC,4)*(MERGE(MC,7)-CAP(L2))/(MERGE(MC,7)-
   /MERGE(MC,5))
    AA1(L1)=CAP(L1)
    CAP(L3)=CAP(L1)+CAP(L2)
```

```
      AA3(L3)=CAP(L3)-AA1(L3)-AA2(L3)
      GO TO 10000
C  L2 SHARES ITS MERGE CAPACITY ENTITLEMENT WITH L1
C  RESERVE SOME CAPACITY BEFORE SHARING
10003 RES=AS(L2)*(1.-PPY)/PPY
      RES=RES+MERGE(MC,13)*(1.-PPY)/(1.-BBEG(MC))
      F=AS(L2)+RES
      RATIO=MERGE(MC,9)/MERGE(MC,8)
C  IF NOTHING TO SHARE, SKIP IT
      IF(F.GT.MERGE(MC,5)*RATIO) GO TO 10000
C  CALCULATE AMOUNT WHICH COULD BE GIVEN TO L1
      EXCESS=RATIO*(MERGE(MC,6)+F/MERGE(MC,5)*(MERGE(MC,4)-MERGE(MC,6)))
     /-AS(L1)
      IF(EXCESS.LE.2.) GO TO 10000
      CAP(L1)=CAP(L1)+EXCESS*DAMP
      AA1(L1)=CAP(L1)
      CAP(L2)=RATIO*MERGE(MC,5)*(MERGE(MC,6)-CAP(L1))/(MERGE(MC,6)-
     /MERGE(MC,4))
      AA1(L2)=CAP(L2)
      CAP(L3)=CAP(L1)+CAP(L2)
      AA3(L3)=CAP(L3)-AA1(L3)-AA2(L3)
10000 CONTINUE
C  UPDATING LINKS
   12 DO 4100 IX=1,NLINKS
C
C  PREPARING APPROPRIATE LINKS FOR QUEUING.  ESTIMATES OF QUEUE SERVING
C  CAPACITIES AND UNIT COSTS OF QUEUING--DOWN TO STATEMENT 4100
C
      IF(AS(IX).LT. CAP(IX)      .AND.TQ(IX).LT.PQC(IX)       )GO TO 4100
      IF(AS(IX).EQ.0.) GO TO 4100
C  PASSING BACK ESTIMATE OF CAPACITY TO SERVE QUEUE
      I=IB(IX)
      FO=0.
      FI=0.
      JD=MAT(I)-1
      JO=MA(I)
      JI=MAB(IX)
      IF(JI.EQ.0) GO TO 4100
      DO 30 J=1,JO
   30 FO=FO+AS(JD+J)
      DO 31 J=1,JI
   31 FI=FI+AS(LINKIN(IX,J))
      IF(FI.LT.1.) FI=1.
      DO 4110 J=1,JI
      IRR=LINKIN(IX,J)
      CSQ(IRR)=CSQ(IX)*AS(IRR)/FI*FO/AS(IX)
      IF(CSQ(IRR).LT.15.) CSQ(IRR)=15.
```

```
      IF(CSQ(IRR).GT.CAP(IRR))CSQ(IRR)=CAP(IRR)
 4110 QCOST(IRR)=TQ(IRR)/CSQ(IRR)*3600.*SLICE+1.
 4100 CONTINUE
C
C  FINDING APPROPRIATE SUBLINK, UNIT FLOW COST AND CORRESPONDING CAPACITY FOR
C  EACH LINK
C
      ICT=0
      DO 4118 IX=1,NLINKS
      IF(AS(IX).LT. CAP(IX)        .AND.TQ(IX).LT.PQC(IX)       )GO TO 4111
      ICOMP(IX)=4
      CUS=CU3(IX)
      AAS=0.
      GO TO 4114
 4111 IF (AS(IX).GE.AA1(IX)       ) GO TO 4113
      ICOMP(IX)=1
      AAS=AA1(IX)-AS(IX)
      CUS=CU1(IX)
      GO TO 4114
 4113 IF(AS(IX).GE.AA1(IX)+AA2(IX)       ) GO TO 4115
      ICOMP(IX)=2
      AAS=AA1(IX)+AA2(IX)-AS(IX)
      CUS=CU2(IX)
      GO TO 4114
 4115 ICOMP(IX)=3
      AAS=CAP(IX)-AS(IX)
      CUS=CU3(IX)
C  FINDING UNIT COST OF QUEUEING AND APPROPRIATE CAPACITY FOR EACH LINK
 4114 IF(AAS.EQ.0.)GO TO 40
      IF(QCOST(IX).LT.1.) GO TO 4120
      QCOST(IX)=TQ(IX)/CSQ(IX)*3600.*SLICE+1.
C LINKS CONTAINING QUEUES IN THE MIN. PATH
   40 ICT=ICT+1
      LINKC(ICT)=IX
      IF(AAS.EQ.0.) GO TO 4120
      IF(TQ(IX).GT.PQC(IX)) GO TO 4122
      CUS=CUS*(1.-TQ(IX)/PQC(IX))
C  FINDING TOTAL COST AND APPROPRIATE CAPACITY FOR EACH LINK
 4122 IF(CSQ(IX)*.02.LT.AAS) AAS=CSQ(IX)*.02
      IF(PQC(IX).LE..0001) GO TO 4120
      IF(PQC(IX)-TQ(IX).LT.AAS) AAS=PQC(IX)-TQ(IX)
 4120 CU(IX)=QCOST(IX)+CUS
      AA(IX)=AAS
 4118 CONTINUE
C
C  END OF AN INCREMENT
C  ************************************************************
C
```

```
C   IF MORE DEMAND TO ASSIGN IN THIS ITERATION, TRY ANOTHER INCREMENT
       IF(PP.GT.1.0) GO TO 10
       PRINT 184
       PRINT 189
C   STORE OLD UNIT COSTS FOR NEXT ITERATION
       DO 28 I=1,NLINKS
    28 OCU(I)=CU(I)
       FRN=FRNN
C
C   SETTING MERGE CAPACITY ENTITLEMENTS FOR NEXT ITERATION
C
       JOE=0
       DO 16 MC=1,NMERGE
       L1=MERGE(MC,1)
       L2=MERGE(MC,2)
       L3=LINKIN(L1,1)
       L4=LINKIN(L2,1)
       RATIO=MERGE(MC,9)/MERGE(MC,8)
       E1=MERGE(MC,10)*RATIO
       E2=MERGE(MC,11)*RATIO
C   ESTIMATING LEFTOVER QUEUE FOR EACH APPROACH
       TL1=TQ(L3)
       TL2=TQ(L4)
       DO 17 I=1,4
       J=LINKIN(L3,I)
       IF(J.NE.0) TL1=TL1+TQ(J)
       J=LINKIN(L4,I)
    17 IF(J.NE.0) TL2=TL2+TQ(J)
       MERGE(MC,12)=TL1
       MERGE(MC,13)=TL2
C   CALCULATE TEMPORARY ENTITLEMENTS IF APPROPRIATE
       IF(NOMRG.GE.1) GO TO 20
C   ESTIMATING TOTAL DEMAND FOR EACH APPROACH
       TL1=TL1+AS(L1)
       TL2=TL2+AS(L2)
C   ESTIMATING APPROPRIATE ENTITLEMENTS ON BASIS OF MASTER ENTITLEMENTS
C   AND ESTIMATED DEMANDS
       IF(TL2.LT.E2) GO TO 18
       IF(TL1.GE.E1) GO TO 33
       EXC=E1-TL1
       REQ=TL2-E2
       DIFF=EXC-REQ
       IF(DIFF.GT.0.) GO TO 19
       CAP(L2)=E2+EXC
       CAP(L1)=TL1
       GO TO 32
    18 EXC=E2-TL2
```

```
      REQ=TL1-E1
      DIFF=EXC-REQ
      IF(DIFF.GT.0.) GO TO 19
      CAP(L1)=E1+EXC
      CAP(L2)=TL2
      GO TO 32
   19 CAP(L2)=TL2+DIFF*.5
      CAP(L1)=TL1+DIFF*.5
      GO TO 32
   33 CAP(L1)=E1
      CAP(L2)=E2
   32 AA1(L1)=CAP(L1)
      AA1(L2)=CAP(L2)
      MERGE(MC,4)=CAP(L1)
      MERGE(MC,5)=CAP(L2)
      WASTE=MERGE(MC,9)-AS(L1)-AS(L2)
      BACKUP=TQ(L3)+TQ(L4)
      IF(WASTE.GT.5..AND.BACKUP.GT.1.) JOE=1
   20 PRINT 190,MC,AS(L1),MERGE(MC,12),AS(L2),MERGE(MC,13)
   16 CONTINUE
C  END OF ITERATION
C  ********************************************************************
C
C  DETERMINING WHETHER TO PERFORM ANOTHER ITERATION
      IF(IDEM.GE.MIDEM) GO TO 51
      IF(IDEM.LT.LIDEM.OR.JOE.EQ.1) GO TO 21
      I=IDEM/2
      II=IDEM-2*I
      IF(II.GT.0) GO TO 21
C  END OF TIME SLICE
C  PRINT OUT FLOWS, QUEUES AND TOTAL TRAVEL TIME
   51 PRINT 181,ISLCE
      DO 520I=1,NNODE
      IBEGIN=MAT(I)-1
      K=MA(I)
      IF(K.EQ.0) GO TO 520
      DO 54  J=1,K
      IBEGIN=IBEGIN+1
      QCU(IBEGIN)=CU(IBEGIN)
      IF(ICOMP(IBEGIN).EQ.1)TC=TC+CU1(IBEGIN)*AS(IBEGIN)
      IF (ICOMP(IBEGIN).EQ.2)TC=TC+CU2(IBEGIN)*AS(IBEGIN)
      IF(ICOMP(IBEGIN).EQ.3) TC=TC+CU3(IBEGIN)*AS(IBEGIN)
   54 CONTINUE
      PRINT 187
      IX=MAT(I)
      PRINT 188,I,(L(J),AS(J),CU(J),J=IX,IBEGIN)
  520 CONTINUE
```

```
  529 CONTINUE
      NN=NNODE+Z4
      DO 52 I=1,NN
      DO 52 J=1,NDEST
      TC=TC+AQ(I,J)*SLICE*3600.
   52 CONTINUE
      PRINT 989,TC
C
C PRINT OUT QUEUES STORED ON LINKS
C
      IQ=0
      WRITE(6,561) ISLCE
      IF(ICT.EQ.0) GO TO 77777
      DO 88888 I=1,ICT
      II=LINKC(I)
      IF( TQ(II).LT..1) GO TO 88888
      WRITE(6,562) IB(II),L(II),TQ(II)
      IQ=IQ+1
88888 CONTINUE
      IF(IQ.EQ.0) GO TO 77777
      GO TO 1
77777 WRITE(6,567)
C
C PRINT OUT CAPACITATED LINKS
C
    1 IU=0
      WRITE(6,564)ISLCE
      DO 6666 I=1,NLINKS
      IF(AS(I).EQ.0.)GO TO 6666
      IF(AS(I).LT.CAP(I)) GO TO 6666
      WRITE(6,562) IB(I),L(I),AS(I)
      IU=IU+1
 6666 CONTINUE
      IF(IU.EQ.0)WRITE(6,566)
C
C  PRINT TURNING VOLUMES
C
      IF(NINT.EQ.0) GO TO 995
      WRITE(6,999)ISLCE
      DO 990 I=1,NINT
      L1=NSUR(I,1)
      L2=NSUR(I,2)
      L3=NSUR(I,3)
      L4=NSUR(I,4)
      LINO=KK(I)
      LINP=KK(I+1)-1
      NN=LINO+3
      NN1=LINO+4
```

```
      IF(L1.EQ.0) GO TO 991
      WRITE(6,992)INT(I),L1,(TURV1(J),J=LINO,NN      )
      WRITE(6,994)L4,INT(I),L2,(TURV1(M),M=NN1,LINP),L3
      GO TO 990
  991 WRITE(6,996)INT(I)
      WRITE(6,994)L4,INT(I),L2,(TURV1(J),J=LINO,LINP),L3
  990 CONTINUE
  995 IF(ISLCE.GE.NSLCE) GO TC 98
      ISLCE=ISLCE+1
      GO TO 85
C  END OF FINAL TIME SLICE
C  ***********************************************************************
C
C  A FINAL CHECK TO SEE IF THERE ARE ANY LEFTOVER QUEUES.
   98 WRITE(6,568)
      IU=0
      J=NNODE+Z4
      DO 96 I=1,J
      DO 96 M=1,NDEST
      IF(AQ(I,M).LE. 0.001) GC TO 96
      IF(ZLINK(I).EQ.0) GO TO 150
      PRINT 151,L(ZLINK(I)),KD(M),AQ(I,M),IB(ZLINK(I)),L(ZLINK(I))
      GO TO 152
  150 PRINT 88,I,KD(M),AQ(I,M)
  152 IU=IU+1
   96 CONTINUE
      IF(IU.EQ.0) WRITE(6,569)
      TC=TC/3600.
      PRINT 89,TC
C
C  FORMAT STATEMENTS
C  *****************
C
   63 FORMAT(≠1≠,≠LINK INPUT OUT OF SEQUENCE≠)
   80 FORMAT(36I2)
   81 FORMAT(25H1INCREMENT LIMIT EXCEEDED)
   82 FORMAT(15I3)
   83 FORMAT(72I1)
   84 FORMAT(6F5.2)
   86 FORMAT(13F6.0)
   87 FORMAT(3I3,8F6.2,I1,4I3)
   88 FORMAT(10X,*LEFTOVER QUEUE FROM NCDE*,I3,* TO NODE*,I3,*=*,F8.3,
     *5X,*DEMAND QUEUED IN ORIGIN*)
   89 FORMAT(//,* TOTAL CCST (VEHICLE-HCURS) FOR ALL TIME SLICES = *,
     /E10.3)
   91 FORMAT(I6,2X, E20.9,I11,3X,2E19.9)
   92 FORMAT(I6,2X,2E20.9,1X,I3,* TO *,I3,3X,2E19.9)
  102 FORMAT(1H1,*      NO PATH FROM ORIGIN      *,I3,*     TO DESTINATION
```

```
  /      *,I3,*        LEFT DEMAND =*,F6.2)
103 FORMAT(I3,6F6.2)
104 FORMAT(2I3,2F6.2)
109 FORMAT(1H1///,36X,23H ***********************/,36X,2H *,20X,1H*/25X,
   /32H ************     PROGRAM CORCON    ,13H *************/,36X,2H *,
   /20X,1H*/36X,23H ***********************///14X,*  THIS PROGRAM IS US
   1ED TO INVESTIGATE TRAFFIC ASSIGNMENT AND CONTROL*/14X,*IN A FREEWA
   2Y CORRIDOR. IT IS INTENDED TO BE USED IN EVALUATING RAMP*/14X,*CON
   3TROL STRATEGIES AND PREDICTING THEIR EFFECTS ON SURFACE STREETS.*/
   4/14X,*  TRAFFIC ASSIGNMENT IS BASED ON INDIVIDUAL TRAVEL COST MINI
   5ZATION.*/,14X,*TRAFFIC DIVERSION IS ALLOWED FROM QUEUEING MINIMUM
   6PATHS WITH THREE*/,14X,*STRATEGIES - TOTAL DIVERSION, NO DIVERSION
   7,OR PARTIAL DIVERSION.*//14X,*  A NEW MINIMUM PATH ALGORITHM WAS D
   8EVELOPED TO ACCOUNT FOR TURN */14X, *PROHIBITIONS WHICH ARE USED T
   9O AVOID ILLOGICAL PATHS OR AS A CONTROL*/14X,      *MEANS. A METHOD
   1 FOR AUTOMATICALLY IDENTIFYING THESE TURN PROHIBITIONS*/14X,*WAS D
   2EVELOPED.*//14X,*  THE PROGRAM ALLOWS A SIMPLE LINK -NODE REPRESEN
   3TATION OF THE NETWORK*/14X,*AND,THEREFORE,DECREASES THE CODING EFF
   4ORT REQUIRED BY THE USER.*//14X,         *  ADDITIONALLY,THE PROGR
   5AM CAN CALCULATE THE TURNING VOLUMES AT ANY*/14X,*SPECIFIED INTERS
   6ECTIONS WITHOUT USING TURNING LINKS.*)
151 FORMAT(10X,*LEFTOVER QUEUE FROM NODE*,I3,* TO NODE*,I3,*=*,F8.3,
   *5X,*DEMAND QUEUED ON LINK*,I3,*-*,I3)
179 FORMAT( //, 2X,* SUMMARY OF THE MAIN INPUT PARAMETERS*,/, 3X,36(*-
   1*),//,14X,   *CONTROL PARAMETER (IF 1 DIVERSION IS CONSIDERED ) =*,
   2I3/14X,*CONTROL PARAMETER (IF 0 ENTITLEMENT UPDATING*/,14X,*ROUTIN
   3E IS CONSIDERED)*,14(*. *),   *=*,I3/14X,*FRACTION OF POSSIBLE SHAR
   4ED CAPACITY( IF MORE*/14X,*THAN 0. MERGE CAP. SHARING ROUTINE IS C
   5ONSIDERED) = *,F5.2/14X,*NUMBER OF TIME SLICES*,14(* .*),X,*=*,I3/
   614X,*DURATION OF TIME SLICE*,14(*. *)   ,*=*,F5.2/,14X,*NUMBER OF L
   7INKS*,17(* .*),X,*=*,I3/14X,*NUMBER OF NODES*,17(* .*),X,*=*,I3/14
   8X,*NUMBER OF ORIGINS*,16(* .*),X,*=*,I3/14X,*NUMBER OF DESTINATION
   9S*,14(*. *),*=*,I3/14X,*AMOUNT BY WHICH UNIT COST IN PREVIOUS TIME
   1SLICE*/14X,*IS WEIGHED INTO PRESENT SLICE*,10(* .*),X,*=*,F5.2/14X
   2,*AMOUNT BY WHICH UNIT COST IN PREVIOUSITERATION*/,14X,*IS WEIGHED
   3 INTO PRESENT ITERATION*,8(* .*),X,*=*,F5.2/14X,*FRACTION BY WHICH
   4 ORIGINAL O-D MATRICES ARE*/14X,*INCREASED TO GIVE REQUIRED DEMAND
   5 LEVEL*,5(* .*),X,*=*,F5.2/14X,*R CONSTANT IN THE DIVERSION MODEL*
   6    ,8(* .*), X,*=*,F5.2)
180 FORMAT(1H1,,X,*ORIGINAL LINK INFORMATION */,3X,25(*-*)///)
181 FORMAT(40H1LINK-FLOW CHARACTERISTICS IN TIME SLICE, I2/,X,39(*-*),
   *//)
182 FORMAT(*    INCREMENT    CUMULATIVE                        ATTEMPTED
  /           ALLOWABLE*)
183 FORMAT(*      NUMBER     ASSIGNMENT      CRITICAL LINK     ASSIGNMENT
  /           ASSIGNMENT*/)
184 FORMAT(///,12X,*MERGE        FREEWAY          ENTRANCE-RAMP*)
```

```
185 FORMAT(1H1//,36X,23H *********************** /,36X,2H *,20X,1H*/36X,
   /            18H *   TIME SLICE   ,I1, 4H    */                  ,36X,2H *,
   /20X,1H*/36X,23H ***********************)
186 FORMAT(///,23X,*ITERATION*,I2/,23X,9(*-*)//)
187 FORMAT(* NODE   *,5(*   TO    FLOW   TR.TIME*))
188 FORMAT (I4,3X,5(I5,F7.1,F9.2))
189 FORMAT(22X,*FLOW    QUEUE     FLOW    QUEUE*/)
190 FORMAT(7X,I8,F11.1,F8.1,F9.1,F8.1)
191 FORMAT (11H QUEUE FROM, I3,3H TO, I3,1H=, F6.1)
195 FORMAT(1H1,5X,* NO. OF ENTERING LINKS TO NODE *,I2,*  IS GREATER T
   /HAN 6*,*   ...INCREASE DIMENSION OF VECTOR NODUL(   )*)
197 FORMAT    (5X,*LINK UPSTREAM DOWNSTREAM  SUBLINK CAPACITIES   SUBL
   1INK UNIT COSTS   UPSTREAM LINKS ONTO*/5X,* NO.    NODE    NONODE
   2  1    2     3              1    2     3      WHICH QUEUES MAY SPILL
   3*/)
198 FORMAT(5X,I3,2X,I5,4X,I6,5X,F6.1,1X,F6.1,1X,F6.1,2X,F6.2,1X,F6.2,
   11X,F6.2,X,I3,3(I5))
199 FORMAT(    /,18X,*FOR A DETAILED DESCRIPTION OF THIS PROGRAM SEE-*/
   1/,22X,*EASA,S.M., A MODEL FOR INVESTIGATING TRAFFIC ASSIGNMENT*/,
   233X,44(*-*)/,33X,*AND CONTROL IN A FREEWAY CORRIDOR,*/33X,33(  *
   3-*),/,33X,*M.ENG.,MCMASTER UNIVERSITY,HAMILTON,*//,33X,*ONTARIO,MA
   4RCH,1976.*)
200 FORMAT(///,2X,*ORIGINAL MERGE INFORMATION*/,2X,26(*-*),///
   /,10X,*  MERGE *,5X,*MERGING *,5X,*DOWNST. MERGING *,5X,* ENTITLEME
   /NT *,5X,* ULTIMATE*/,12X,*NUMBER *,4X,  *LINKS NO.*,4X,*LINK NO. *
   /,13X,*CAPACITY *,7X,*  CAPACITY*/,11X,//)
201 FORMAT(22X,F7.2,31X,F7.2,10X,F7.2)
202 FORMAT(13X,I3,22X,F7.2)
203 FORMAT(22X,F7.2,31X,F7.2,10X,F7.2,/,11X,/)
205 FORMAT(///,*    NEW LINK INFORMATION FOR THIS TIME SLICE */,3X,
   /40(*-*))
206 FORMAT(//,12X,*NONE*//,10X,*..ORIGINAL LINK INFORMATION IS USED *)
207 FORMAT(//,12X,*NONE*//,10X,*..LINK INFORMATION OF PREVIOUS TIME SL
   /ICE IS USED*)
208 FORMAT(///,*    NEW MERGE INFORMATION FOR THIS TIME SLICE */,3X,
   /41(*-*))
209 FORMAT(//,12X,*NONE*//,10X,*..ORIGINAL MERGE INFORMATION IS USED*)
210 FORMAT(//,12X,*NONE*//,10X,*..MERGE INFORMATION OF PREVIOUS TIME S
   /LICE IS USED*)
211 FORMAT(///,3X,*FLOWS AND QUEUES FOR EACH FREEWAY AND RAMP MERGING
   /APPROACHES*    /3X,61(*-*)//12X,*FLOWS ARE IN VEH./TIME SLICE*/12X,
   /*QUEUES ARE THE NO. OF VEHICLES ON AND*/12X,*UPSTREAM OF EACH APPR
   /OACH*)
212 FORMAT(///,3X,*DETAILED RESULTS OF INCREMENTAL ASSIGNMENT*/,3X,
   /42(*-*)//,12X, *AFTER EACH ITERATION FLOWS AND QUEUES  OF  MERGING
   /*/,12X,*APPROACHES ARE GIVEN. FLOWS ARE IN VEH./TIME SLICE*/,12X,*
   /AND QUEUES ARE THE NO. OF VEHICLES ON AND UPSTREAM*/,12X,*OF EACH
   /MERGING APPROACH*//)
```

```
213 FORMAT(//,5X ,*LINK   UPSTREAM   DOWNSTREAM   SUBLINK CAPACITIES
   1 SUBLINK UNIT COSTS *                                 ,/,5X,* NO.     NODE
   2        NODE          1     2     3           1     2     3 */)
214 FORMAT(//11X,* MERGE *,5X,*MERGING *,7X,*DOWNST. MERGING *,4X,
   /* ENTITLEMENT *,4X,* ULTIMATE*/,12X,*NUMBER *,4X,*LINKS NO.*,6X,
   /*LINK NO.*,13X,*CAPACITY *,7X,* CAPACITY*//)
215 FORMAT(///,10X, *..ENTITLEMENT CAPACITY FOR RAMP IS ITS*/,12X,
   /*METERING RATE CONSIDERED*//,10X,*..NOTE THAT ANY METERING RATES
   /CONSIDERED*/,12X,*IN THE PREVIOUS TIME SLICE ARE APPLIED*/,12X,*IN
   / THIS TIME SLICE UNLESS THEY ARE CHANGED*/,12X,*IN THIS TIME SLICE
   /*)
556 FORMAT(20I4)
558 FORMAT(I1)
560 FORMAT(13F6.2)
561 FORMAT(1H1,5(/), 2X,*LINKS HAVING QUEUES AFTER TIME SLICE  *   ,I2
   /,/, 2X,36(*-*)5(/),14X,*LINK*,/,14X,*FROM*,10X,*QUEUE*,/,10X,*NODE
   /*,5X,*TO*,5X,*(VEHICLES)*/)
562 FORMAT(10X,I3,5X,I3,7X,F7.2)
563 FORMAT(5I3)
564 FORMAT(5(/), 2X,*LINKS REACHING CAPACITY AFTER TIME SLICE   *,I2,/
   /, 2X,40(*-*),5(/),14X,*LINK*,/,14X,*FROM*,/,10X,*NODE*,5X,*TO*,5X,
   /*FLOW(=CAP.)*/)
566 FORMAT(3(/),10X,*THERE ARE NO CAPACITATED LINKS*)
567 FORMAT(3(/),10X,*THERE ARE NO QUEUEING LINKS *)
568 FORMAT(5(/), 2X,*LEFTOVER QUEUES AFTER FINAL TIME SLICE*/,2X,
   /32(*-*)//)
569 FORMAT(10X,*NO LEFTOVER QUEUES AFTER THIS TIME SLICE*)
570 FORMAT(1H1,//, 2X,*NETWORK TURN PROHIBITIONS*/,2X,25(*-*),///,25X,
   1*UPSTREAM LINK OF*,5X,*DOWNSTREAM LINK OF*,
   2      /,5X,*TURN PROH. NO.*,6X,*THE TURN*,13X,*THE TURN*///)
572 FORMAT(12X,I2,2H *,14X,I2,*-*,I2,16X,I2,*-*,I2)
573 FORMAT(//,9X,                                           2H *,
   1          *  TURN PROHIBITIONS AT INTERCHANGES*/,13X,*AND INTERSECTIO
   /NS (INCLUDING U-TURNS)*//,8X,3H **,          *  TURN PROHIBITIONS
   /AT MERGING SECTIONS*)
575 FORMAT(12X,I2,3H **,13X,I2,*-*,I2,16X,I2,*-*,I2)
989 FORMAT(29H CUMULATIVE TIME IN SECONDS = ,F12.1)
992 FORMAT(//        ,8X,*TURNING VOLUMES FOR INTERSECTION AT NODE ..*,2
   /X,I3,//   34X,5H*****/,34X,1H*,3X,1H*,/,24X,1H*,I3,1H*,/,34X,      1
   1H*,3X,1H*/34X,5H*****,4(/,36X,1H*),/,28X,*...*,5X,1H*,5X,*...*,/,2
   21X,F6.0,2X,*..*,5X,1H*,5X,*..*,2X,F6.0/,28X,*  .*,5X,1H*,5X,*.  *,*
   3.*,/,27X,*.*,8X,1H*,8X,*.*,/,26X,*.*,9X,1H*,9X,*.*,/,25X,*.*,5X,*.
   4*,4X,1H*,4X,*.*,5X,*.*,/,24X,*.*,5X,*.*,5X, 1H*,5X,*.*,5X,*.*,/,23
   5X,*.*,5X,*.*,6X,1H*,6X,*.*,5X,*.*,/,28X,*.*,7X,1H*,7X,*.*,/,27X,*.
   6*,8X,1H*8X,*.*,/,24X,*.*,X,*.*,9X,1H*,9X,*.*,X,*.*,/,17X,F6.0,X,*.
   7.*,10X,1H*,10X,*..*,X,F6.0,/,24X,*...*,9X,1H*,9X,*...*)
994 FORMAT(8X,5H*****,21X,5H*****,21X,5H*****/8X,1H*,3X,1H*,21X,1H*,3X
   1,1H*,21X,1H*,3X,1H*/8X,1H*,I3,1H*,21(1H*),1H*,I3,,1H*,21(1H*),1H*,
```

```
     2I3,1H*/8X,1H*,3X,1H*,21X,1H*,3X,1H*,21X,1H*,3X,1H*/8X,5H*****,21X,
     35H*****,21X,5H*****/24X,*...*,9X,1H*,9X,*...*/17X,F6.0,X,*..*,10X,
     41H*,10X,*..*,X,F6.0,/,24X,*.*,X,*.*
     5,9X,1H*,9X,*.*,X,*.*,            /,27X,*.*,8X,1H*,8X,*.*,/,28X,*.*,7X,
     61H*,7X,*.*,/,23X,*.*,5X,*.*,6X,1H*,6X,*.*,5X,*.*,/,24X,*.*,5X,*.*,
     75X,1H*,5X,*.*,5X,*.*,/,25X,*.*,5X,*.*,4X,1H*,4X,*.*,5X,*.*,/,26X,*
     8.*,9X,1H*,9X,*.*,/,27X,*.*,8X,1H*,8X,*.*
     9  ,/,28X,*.  .    * ,X,1H*,X,*      . .* ,/,21X,F6.0,2X,*..*,5X,1H*,5
     1X,*..*,2X,F6.0,/,28X,*...*,5X,1H*,5X,*...*,4(/,36X,1H*)/34X,9H****
     2*,/,34X,1H*,3X,1H*,/,34X,1H*,I3,1H*,/,34X,1H*,3X,1H*/34X,5H*****)
 996 FORMAT(///,4(/),8X,*TURNING VOLUMES FOR INTERSECTION AT NODE ..*,2
     /X,I3,5(/))
 999 FORMAT(1H1/ ,21X,*TURNING VOLUMES FOR SPECIFIED *,/,21X,30(*-*),/,
     /21X,*FOR  INTERSECTIONS-TIME SLICE*,I3,/,21X,29(*-*))
      STOP
      END
```

```
      SUBROUTINE MINPATH (SOURCE,C,LINK,NP,LINKC,ICT,CU,Y2,CC,LINKK)
C     ************************************************************
C
C   THIS SUBROUTINE PERFORMS THE FORWARD PROCEDURE
C   OF THE MINIMUM PATH ALGORITHM WITH TURN PROHIBITIONS.
C   THIS PROCEDURE IS PERFORMED FROM ORIGIN UNDER
C   CONSIDERATION TO EACH NODE IN THE NETWORK
C
      INTEGER   ENTRY,ONE,SOURCE,TABLE(2,200),TWO,Z,ZLINKO(100)
      DIMENSION C(75),CC(75,4),CU(130),LINK(75),LINKC(130),LINKK(75,4),
     *          LUP(4),NP(75)
      COMMON    INTAP,JCI,NH,NINT,NNODE,NTPRO,NTPRO1,NLINKS,NR,Z,
     *          CU1(130),FET(130),IB(130),INT(10),KK(11),L(130),
     *          LINKIN(130,4),LINTD(80),LINTU(80),MA(75),MAT(75),NAV(50)
     *          ,NOD(75),NS1(100),NSUR(10,4),TURV1(100),NRC(5),ZLINKO
C
  100 FORMAT(1H1,* LAST(IN TREE) IS *,I4,*-INCREASE DIMENSION OF TABLE*)
  575 FORMAT(1H1,*NO. OF LINKS ENTERING NODE NUMBER *,I3,* IS MORE THAN
     / 4 */,*  AT THIS NODE THERE IS TURN PROHIBITION...NO. OF ENTERING
     ** LINKS NOT TO EXCEED 4 *)
      N=0
      ONE=1
      TWO=2
      IF(ZLINKO(NH).EQ.0) GO TO 4
      LAST=1
      TABLE(TWO,1)=ZLINKO(NH)
      GO TO 7
    4 LINO=MAT(SOURCE)
      LINKE=LINO+MA(SOURCE)-1
      LAST=LINKE-LINO+1
      DO 40 ENTRY=1,LAST
      IF(Y2.EQ.0.) GO TO 1003
C OMITTING ANY QUEUEING LINKS OUT OF THE SOURCE
      DO 2001 I=1,ICT
      IF(LINKC(I).EQ.LINO) GO TO 40
 2001 CONTINUE
 1003 N=N+1
      TABLE(TWO,ENTRY)=LINO
   40 LINO=LINO+1
C RETURN IF THERE ARE NOT ANY NON-QUEUEING LINKS OUT OF THE SOURCE
      IF(N.EQ.0)GO TO 55
C  REVERSING TABLE POINTERS
    7 ONE=3-ONE
      TWO=3-TWO
C  TRY LOWERING COST WITH EACH LINK IN TABLE ONE
      IV=0
      IF(LAST.GT.195)WRITE(6,100) LAST
      DO 50 ENTRY=1,LAST
```

```
      LINO=TABLE(ONE,ENTRY)
      INODE=IB(LINO)
      JNODE=L(LINO)
C  IF LINO IS A DOWNSTREAM LINK,USE THE MINIMUM COSTS OF ALL UPSTREAM
C  LINKS.  OF THE TURN PROHIBITION NODE(EXCEPT THE CORRESPONDING UPST.
C  LINKS  OF THE TURN),OTHERWISE USE THE MINIMUM COST OF ALL UPST. LINKS
      IF(NOD(INODE).GT.0) GO TO 312
      IF(INODE.EQ.SOURCE) GO TO 312
      N4=0
      K1=NOD(INODE)/100
      K2=-NOD(INODE)+K1*100
      K3=K2+5
      IF(K3.GT.NTPRO)K3=NTPRO
      DO 313 J=K2,K3
      K4=LINTU(J)
      IF(INODE.NE.L(K4)) GO TO 1
      IF(LINTD(J).EQ.LINO) GO TO 314
      GO TO 313
  314 N4=N4+1
      LUP(N4)=LINTU(J)
  313 CONTINUE
    1 IF(N4.EQ.0) GO TO 317
      DO 315 J=1,4
      M5=LINKK(INODE,J)
      DO 316 J1=1,N4
      IF(LUP(J1).EQ.M5) GO TO 315
  316 CONTINUE
      IF(CC(INODE,J).EQ.9999.) GO TO 50
      C(INODE)=CC(INODE,J)
      GO TO 312
  315 CONTINUE
      GO TO 312
  317 C(INODE)=CC(INODE,1)
  312 IF(INODE.LT.INTAP.AND.JNODE.LT.INTAP) GO TO 5
      IF(INODE.GE.INTAP) GO TO 65
      HOLDTD=C(INODE)
      GO TO 6
    5 HOLDTD=C(INODE)+CU(LINO)
      GO TO 6
C  INTERSECTION MOVEMENT LINK
   65 IT=LINK(INODE)
      HOLDTD=     C(INODE)+CU(IT) +CU1(LINO)
    6 IF(NOD(JNODE).GT.0) GO TO 310
C  ARRANGEMENTS OF THE COSTS OF THE TURN PROHIBITION NODE*JNODE*IN
C  ASCENDING ORDER
      DO 320 I=1,
      M1=LINKK(JNODE,I)
```

```
      IF(LINO.EQ.M1) GO TO 321
  320 CONTINUE
      GO TO 322
  321 CM1=CC(JNODE,I)
      IF(HOLDTD.GE.CM1) GO TO 50
      GO TO (305,306,307,308),I
  322 IF(CC(JNODE,1).EQ.9999.) GO TO 305
      IF(CC(JNODE,2).EQ.9999.) GO TO 306
      IF(CC(JNODE,3).EQ.9999.) GO TO 307
      IF(CC(JNODE,4).EQ.9999.) GO TO 308
      WRITE(6,575) JNODE
      STOP
  305 CC(JNODE,1)=HOLDTD
      LINKK(JNODE,1)=LINO
      GO TO 304
  306 IF(HOLDTD.LT.CC(JNODE,1)) GO TO 323
      CC(JNODE,2)=HOLDTD
      LINKK(JNODE,2)=LINO
      GO TO 304
  323 T=CC(JNODE,1)
      LT=LINKK(JNODE,1)
      CC(JNODE,1)=HOLDTD
      LINKK(JNODE,1)=LINO
      CC(JNODE,2)=T
      LINKK(JNODE,2)=LT
      GO TO 304
  307 IF(HOLDTD.LT.CC(JNODE,2))GO TO 324
      CC(JNODE,3)=HOLDTD
      LINKK(JNODE,3)=LINO
      GO TO 304
  324 IF(HOLDTD.LT.CC(JNODE,1))GO TO 325
      T=CC(JNODE,2)
      LT=LINKK(JNODE,2)
      CC(JNODE,2)=HOLDTD
      LINKK(JNODE,2)=LINO
      CC(JNODE,3)=T
      LINKK(JNODE,3)=LT
      GO TO 304
  325 T=CC(JNODE,2)
      LT=LINKK(JNODE,2)
      CC(JNODE,2)=CC(JNODE,1)
      LINKK(JNODE,2)=LINKK(JNODE,1)
      CC(JNODE,3)=T
      LINKK(JNODE,3)=LT
      CC(JNODE,1)=HOLDTD
      LINKK(JNODE,1)=LINO
      GO TO 304
  308 IF(HOLDTD.LT.CC(JNODE,1)) GO TO 326
      IF(HOLDTD.LT.CC(JNODE,2)) GO TO 327
```

```
      IF(HOLDTD.LT.CC(JNODE,3)) GO TO 328
      CC(JNODE,4)=HOLDTD
      LINKK(JNODE,4)=LINO
      GO TO 304
  326 T=CC(JNODE,1)
      LT=LINKK(JNODE,1)
      CC(JNODE,1)=HOLDTD
      LINKK(JNODE,1)=LINO
      CC(JNODE,4)=CC(JNODE,3)
      LINKK(JNODE,4)=LINKK(JNODE,3)
      CC(JNODE,3)=CC(JNODE,2)
      LINKK(JNODE,3)=LINKK(JNODE,2)
      CC(JNODE,2)=T
      LINKK(JNODE,2)=LT
      GO TO 304
  327 T=CC(JNODE,2)
      LT=LINKK(JNODE,2)
      CC(JNODE,2)=HOLDTD
      LINKK(JNODE,2)=LINO
      CC(JNODE,4)=CC(JNODE,3)
      LINKK(JNODE,4)=LINKK(JNODE,3)
      CC(JNODE,3)=T
      LINKK(JNODE,3)=LT
      GO TO 304
  328 T=CC(JNODE,3)
      LT=LINKK(JNODE,3)
      CC(JNODE,3)=HOLDTD
      LINKK(JNODE,3)=LINO
      CC(JNODE,4)=T
      LINKK(JNODE,4)=LT
  304 IF(HOLDTD.GE.C(JNODE))GO TO 309
      GO TO 311
  310 IF(HOLDTD.GE.C(JNODE)) GO TO 50
  311 C(JNODE)=HOLDTD
C  LINK INTO NODE (JNODE) IN THE TREE
        LINK(JNODE)=LINO
C  NODE INTO NODE (JNODE) IN THE TREE
      NP(JNODE)=INODE
C  ADD LINKS OUT OF JNODE TO TABLE  TWO
  309 IF(MA(JNODE).EQ.0) GO TO 50
      LINKB=MAT(JNODE)
      LINKE=MA(JNODE)+LINKB-1
      DO 49 LINO=LINKB,LINKE
      IF(INODE.EQ.L(LINO)) GO TO 49
C  CHECK IF THERE ARE ANY _O-D NODES
      IF(Z.EQ.0) GO TO 1500
      DO 8 I=1,Z
      IF(NAV(I).EQ.LINO) GO TO 49
```

```
   8 CONTINUE
C CHECK IF THERE ARE ANY CONGESTED LINKS
 1500 IF(Y2.EQ.0) GO TO 1002
      DO 2000 I=1,ICT
      IF(LINKC( I ).EQ.LINO) GO TO 49
 2000 CONTINUE
 1002 IF(NR.EQ.0) GO TO 2
      DO 3 I=1,NR
      IF(NRC(I).EQ.LINO) GO TC 49
    3 CONTINUE
    2 IV=IV+1
      TABLE(TWO,IV)=LINC
   49 CONTINUE
   50 CONTINUE
C   CHECK IF TREE IS COMPLETED
      IF(IV.EQ.0) GO TO 55
      LAST=IV
      GO TO 7
   55 RETURN
      END
```

```
      SUBROUTINE FCONPT (II,NP,LINK,TTC,IC,IY,JC,SOURCE,QCOST,AS,CAP,TQ,
C     ****************************************************************
     /PQC,CU,CC,LINKK,C,Z5)
C     ********************
C
C  THIS SUBROUTINE PERFORMS THE BACKWARD PROCEDURE
C  OF THE MINIMUM PATH ALGORITHM  WITH TURN PROHIBITIONS.
C  IT IS PERFORMED FROM EACH DESTINATION TO THE ORIGIN
C  UNDER CNSIDERATION TC FINALLY DETERMINE THE MINIMUM
C  PATHS.DURING THIS PROCEDURE THE FIRST CONGESTED
C  POINT IN THE PATH IS DETERMINED.
C
      INTEGER    SOURCE,Z5,ZLINKO(100)
      DIMENSION AS(130),C(75),CAP(130),CC(75,4),CU(130),QCOST(130),
     *          LINK(75),LINKK(75,4),NP(75),PQC(130),TQ(130),TTC(25)
      COMMON    INTAP,JCI,NH,NINT,NNODE,NTPRO,NTPRO1,NLINKS,NR,Z,
     *          CU1(130),FET(130),IB(130),INT(10),KK(11),L(130),
     *          LINKIN(130,4),LINTD(80),LINTU(80),MA(75),MAT(75),NAV(50)
     *          ,NOD(75),NS1(100),NSUR(10,4),TURV1(100),NRC(5),ZLINKO
C
      IC=0
      JC=II
      I=II
   27 J=NP(I)
      IF(JC.NE.I) GO TO 32
      JCI=J
      Z5=LINK(I)
      IF(IFETCH.EQ.ZLINKO(NH)) GO TO 30
   32 IFETCH=LINK(I)
C TEST IF LINK WILL PATH VEHICLES THROUGH
      IF(AS(IFETCH).EQ.CAP(IFETCH)) GO TO 1
      IF(QCOST(IFETCH).LT.1.) GO TO 20000
      JC=I
      JCI=J
      Z5=LINK(I)
    1 IC=IC+1
C CAL. TOTAL QUEUEING TIME IN THE PATH FROM SOURCE TO DEST. IY
      TTC(IY)=TTC(IY)+QCOST(IFETCH)
20000 IF(AS(IFETCH).LT.CAP(IFETCH).AND.TQ(IFETCH).LT.PQC(IFETCH))
     /GO TO 20004
      JC=J
20004 IF(J.EQ.SOURCE) GO TO 30
      IF(NOD(J).GT.0) GO TO 10
      K=LINK(J)
      IF(IFETCH.EQ.1) GO TO 70
      IF(IFETCH.EQ.NLINKS) GO TO 71
   70 IF(IB(IFETCH).EQ.IB(IFETCH+1).AND.L(IFETCH).EQ.L(IFETCH+1))GO TO40
      IF(IFETCH.EQ.1) GO TO 72
```

```
71 IF(IB(IFETCH).EQ.IB(IFETCH-1).AND.L(IFETCH).EQ.L(IFETCH-1))GO TO40
72 GO TO 50
40 IF(LINKIN(IFETCH,1).EQ.K) GO TO 10
   LINK(J)=LINKK(J,2)
   NP(J)=IB(LINK(J))
   IF(JC.NE.J) GO TO 10
   JCI=NP(J)
   Z5=LINK(J)
   GO TO 10
50 DO 60 J1=1,4
   IF(CC(J,J1).EQ.9999.) GO TO 60
   K1=NOD(J)/100
   K2=-NOD(J)+K1*100
   K3=K2+5
   IF(K3.GT.NTPRO)K3=NTPRO
   DO 20 M=K2,K3
   K4=LINTU(M)
   IF(J.NE.L(K4)) GO TO 61
   IF(LINKK(J,J1).EQ.LINTO(M).AND.IFETCH.EQ.LINTD(M)) GO TO 60
20 CONTINUE
61 LINK(J)=LINKK(J,J1)
   NP(J)=IB(LINK(J))
   IF(JC.NE.J) GO TO 10
   JCI=NP(I)
   Z5=LINK(I)
   GO TO 10
60 CONTINUE
10 I=J
   IF(I.NE.SOURCE) GO TO 27
30 RETURN
   END
```

```
      SUBROUTINE ASSIGN (JC,LINK,II,ATQ,A,IY,NP,SOURCE,X,D,S,TURV,PCC,QC
C     ******************************************************************
     /OST)
C     ****
C
C
C  THIS SUBROUTINE ASSIGNS THE REMAINING DEMAND TO LINKS
C  UPSTREAM OF THE FIRST CONGESTED POINT IN THE MINIMUM
C  PATH.IT ALSO KEEPS RECORD OF THE TURNING VOLUMES
C  FOR SPECIFIED INTERSECTIONS.
C
      INTEGER    SOURCE,ZLINKO(100)
      DIMENSION A(130),ATQ(130),D(100,25),LINK(75),NP(75),PCC(130),
     *          QCOST(130),TURV(100)
      COMMON    IN!AP,JCI,NH,NINT,NNODE,NTPRO,NTPRO1,NLINKS,NR,Z,
     *          CU1(130),FET(130),IB(130),INT(10),KK(11),L(130),
     *          LINKIN(130,4),LINTD(80),LINTU(80),MA(75),MA1(75),NAV(50)
     *          ,NOD(75),NS1(100),NSUR(10,4),TURV1(100),NRC(5),ZLINKO
C
C  TENTATIVELY STORING THE QUEUED DEMAND PHYSICALLY ON THE LINK
      I=JC
      IX=LINK(JC)
C IF LINK IS DUMMY LINK,STORE THE QUEUES IN THE PRIVIOUS LINK
      IF(JC.EQ.II) GO TO 10
      IF(PCC(IX).LE..0001) GO TO 20
      ATQ(IX)=ATQ(IX)+D(NH     ,IY)*S*X
      GO TO 10
   20 I2=NP(I)
      I3=LINK(I2)
      QCOST(I3)=1.0
      ATQ(I3)=ATQ(I3)+D(NH     ,IY)*S*X
C  TENTATIVELY ADDING FLOWS TO LINKS
   10 FEQ=1.0
   29 A(IX)=A(IX)+D(NH     ,IY)*S*X*FEQ
      FEQ=FET(IX)
      K1=I
      I=NP(I)
      K2=I
      IF(I.EQ.SOURCE) GC TO 22
C  CALCULATION OF TURNING VOLUMES
      IF(NINT.EQ.0) GO TO 30
      DO 81 J1=1,NINT
      IF(INT(J1).NE.I) GO TO 81
      IX=LINK(I)
      I1=NP(I)
      LINO=KK(J1)
      LINP=KK(J1+1)-1
      DO 84 MM=LINO,LINP
      M1=NS1(MM)/100
```

165

```
      M2=NS1(MM)-M1*100
      IF(M1.EQ.I1.AND.M2.EQ.K1) GO TO 85
      GO TO 84
   85 TURV(MM)=TURV(MM)+D(NH     ,IY)*S*X.
   84 CONTINUE
   81 CONTINUE
   30 IX=LINK(I)
      IF(ZLINKO(NH).EQ.IX) GO TC 22
      GO TO 29
   22 RETURN
      END
```

REFERENCES

1. Aidoo, J.F., Gaedhart, R.W., and May, A.D., Bay area Freeway Operations Study, Final Report: Analytic Techniques for Evaluating Freeway Improvements, University of California, Berkeley, California, 1971.

2. Brown, G.J.H., and Scott, R.S., Micro-Assignment: A New Tool for Small-Area Planning, Highway Research Record 322, 1970, pp. 149-161.

3. Bureau of Public Roads, U.S. Department of Commerce, Traffic Assignment Manual, 1964. Cited in Reference No. 9.

4. Caldwell, T., On Finding Minimum Routes in a Network With Turn Penalties, Communs Ass. Comput. Mach.4, 1961, pp. 107-108.

5. Control Data Corporation, Data Centres Division, User's Manual, Transportation Planning System for the Control Data 3600 Computer, 1965. Cited in Reference No. 9.

6. Dreyfus, S.E., An Appraisal of Some Shortest-Path Algorithms, Operations Research, Vol. 17, 1969, pp. 395-412.

7. Highway Capacity Manual, HRB spec. Report 87, 1965.

8. Homburger, W.S., Traffic Estimation-Computer Program for Educational purpose, 4th ed., ITTE Course Notes, University of California, Berkeley, California, 1972.

9. Kirby, R.F., and Potts, R.B., The Minimum Route Problem for Networks With Turn Penalties and Prohibitions, Transpn.

Res. Vol. 3, 1969, pp. 397-408.

10. Martin, B.V., Memmott, F.W., III and Bone, A.J., Principles and Techniques of Predicting Future Demand for Urban Area Transportation, MIT Report No. 3, 1971.

11. May, A.D., Freeway Surveillance and Control Systems in the United States, Traffic Engineering and Control, 1970.

12. McDermott, J.M. Operational Effects of Automatic Ramp Control on Network Traffic, Highway Research Record 202, 1967, pp. 1-26.

13. Pignataro, L.J. et al., Weaving Area Operations Study: Analysis and Recommendations, Highway Research Record 398, 1972, pp. 15-30.

14. Potts, R.B., and Oliver, R.M., Flows in Transportation Networks, Mathematics in Science and Engineering, Vol. 90, New York and London, 1972.

15. Stover, V.G., and Blumentritt, C.W., User's Manual for the Texas Large Systems Traffic Assignment Programs, Texas Transportation Institute, Texas A & M University, College Station, Texas, 1966.

16. Wang, J.J., and May, A.D., Freeway Operations Study-Phas III, Interim Report No. 2: Analytical Techniques for Ramp Control, University of California, Berkeley, Calif. 1972.

17. Wattleworth, J.A., and Shuldiner, P.W., Analytical Methods in Transportation; Left Turn Penalties in Traffic Assignment Models, J. Engng. Mech. Div. Am. Soc. Civ. Engrs. 89, 1963, pp. 97-126.

18. Yagar, S., Analysis of the Peak Period Travel in a Freeway-Arterial Corridor, Ph.D. Thesis, University of California,

Berkeley, California, 1970.

19. Yagar, S., Dynamic Traffic Assignment by Individual Path Minimization and Queueing, Transp. Res., Vol. 5, 1971, pp. 179-196.

20. Yagar, S., Dynamic Assignment of Time-Varying Demands to Time-Dependent Networks, Transportation Development Agency, Canada Ministry of Transport, 1974.