TECHNICAL REPORT STANDARD TITLE PAGE

		TECHNICAL REPORT STANDARD TITLE PAGE				
1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.				
FHWA/TX-86/45+356-1						
4. Title and Subtitle		5. Report Date				
TRUCK LANE NEEDS METHODOLOG	Y: A HEURISTIC	November 1985				
APPROACH TO SOLVE A FIVE OF NETWORK DESIGN PROBLEM	TION DISCRETE	6. Performing Organization Code				
7. Author(s)	<u> </u>	8. Performing Organization Report No.				
John J. Massimi, Hani S. Ma and C. Michael Walton	hmassani,	Research Report 356-1				
9. Performing Organization Name and Addres		10. Work Unit No.				
Center for Transportation F	logoarah					
The University of Texas at		11. Contract or Grant No.				
Austin, Texas 78712-1075	AUSEIN	Research Study 3-18-83-356				
Austin, lexas /8/12-10/5		13. Type of Report and Period Covered				
12. Sponsoring Agency Name and Address Texas State Department of F Transportation; Transp	lighways and Public portation Planning Divisio	Interim				
P. O. Box 5051	<u> </u>	14. Sponsoring Agency Code				
Austin, Texas 78763						
15. Supplementary Notes Study conducted in cooperation with the U. S. Department of Transportation, Federal Highway Administration Research Study Title: "A Study of Truck Lane Needs"						
16. Abstract						
increasing traffic of large report describes a procedur related link improvements i an integrated network model the Texas highway network. The link improvement s problem with multiple impro- this procedure is the define expansion (lane addition) a of both existing and added of cars and trucks in the t ment problem embedded in the A branch and bound int	e for the selection of an n the highway network. The ling methodology for the election problem is cast vement types per link. On ition of link improvement nd operational scheme (ex- lanes). Another is the co- raffic stream in solving e network design problem. eger programming approach	ne Texas highway system. This optimal subset of truck- nis procedure is a component of study of truck lane needs in as a discrete network design ne of the principal features of				
increasing traffic of large report describes a procedur related link improvements i an integrated network model the Texas highway network. The link improvement s problem with multiple impro- this procedure is the defin expansion (lane addition) a of both existing and added of cars and trucks in the t ment problem embedded in th A branch and bound int problem and the particular	r and heavier trucks on the e for the selection of an n the highway network. The ling methodology for the election problem is cast vement types per link. Of ition of link improvement nd operational scheme (ex- lanes). Another is the co- raffic stream in solving e network design problem. eger programming approach	he Texas highway system. This optimal subset of truck- his procedure is a component of study of truck lane needs in as a discrete network design he of the principal features of in terms of both capacity clusive use by cars or trucks onsideration of the interaction the network equilibrium assign- is adapted and tested for this ack-related link improvement				
increasing traffic of large report describes a procedur related link improvements i an integrated network model the Texas highway network. The link improvement s problem with multiple impro- this procedure is the define expansion (lane addition) a of both existing and added of cars and trucks in the t ment problem embedded in th A branch and bound int problem and the particular measures.	r and heavier trucks on the e for the selection of an n the highway network. The ling methodology for the election problem is cast vement types per link. Or ition of link improvement nd operational scheme (ex- lanes). Another is the co- raffic stream in solving e network design problem. eger programming approach features introduced by tru- 18. Distribution S	he Texas highway system. This optimal subset of truck- his procedure is a component of study of truck lane needs in as a discrete network design he of the principal features of in terms of both capacity clusive use by cars or trucks onsideration of the interaction the network equilibrium assign- is adapted and tested for this ack-related link improvement				
increasing traffic of larger report describes a procedur related link improvements i an integrated network model the Texas highway network. The link improvement so problem with multiple impro- this procedure is the definer expansion (lane addition) a of both existing and added of cars and trucks in the to ment problem embedded in the A branch and bound into problem and the particular measures.	r and heavier trucks on the for the selection of an in the highway network. The ling methodology for the selection problem is cast avenent types per link. On ition of link improvement ind operational scheme (explanes). Another is the coraffic stream in solving e network design problem. eger programming approach features introduced by trues intrues introduced by tr	he Texas highway system. This optimal subset of truck- his procedure is a component of study of truck lane needs in as a discrete network design he of the principal features of in terms of both capacity clusive use by cars or trucks onsideration of the interaction the network equilibrium assign- is adapted and tested for this ack-related link improvement				
<pre>increasing traffic of large report describes a procedur related link improvements i an integrated network model the Texas highway network. The link improvement s problem with multiple impro- this procedure is the define expansion (lane addition) a of both existing and added of cars and trucks in the t ment problem embedded in th A branch and bound int problem and the particular measures. 17. Key Words truck lane, methodology, net an an a</pre>	r and heavier trucks on the e for the selection of an n the highway network. The ling methodology for the election problem is cast vement types per link. Of ition of link improvement nd operational scheme (ex- lanes). Another is the car affic stream in solving e network design problem. eger programming approach features introduced by tru- features introduced by tru- des, , five ational National S	he Texas highway system. This optimal subset of truck- his procedure is a component of study of truck lane needs in as a discrete network design he of the principal features of in terms of both capacity clusive use by cars or trucks onsideration of the interaction the network equilibrium assign- is adapted and tested for this ack-related link improvement				

TRUCK LANE NEEDS METHODOLOGY: A HEURISTIC APPROACH TO SOLVE A FIVE OPTION DISCRETE NETWORK DESIGN PROBLEM

by

John J. Massimi Hani S. Mahmassani and C. Michael Walton

Research Report Number 356-1

A Study of Truck Lane Needs Research Project Number 3-18-83-356

conducted for

Texas State Department of Highways and Public Transportation

> in cooperation with the U.S. Department of Transportation Federal Highway Administration

> > by

CENTER FOR TRANSPORTATION RESEARCH BUREAU OF ENGINEERING RESEARCH THE UNIVERSITY OF TEXAS AT AUSTIN

November 1985

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

PREFACE

This report documents the network modelling methodology developed for the study of truck lane needs in the Texas highway network. A general overview of the overall approach, as well as a description of the model's capabilities and input requirements, can be found in a companion report on the findings of study CTR 3-18-83-356. The present technical report is intended to fully document the research performed, specifically the development of the network design procedure used to select a subset of link improvements to be implemented under given expenditure levels. This report serves as a technical user's guide, as well as a reference for modifying or updating the computer program.

The principal features of the network link selection methodology described in this report are: 1) the capability to select not only which links to improve but also the type of improvement, 2) the definition of link improvement in terms of not only the provision of special truck lanes, but also the type of operation, including restricted access of existing or new lanes to either vehicular class, and 3) the representation of the interaction of cars and trucks in the traffic stream and its effect on travel time.

·

ABSTRACT

Special truck lanes have been proposed as a measure to deal with the increasing traffic of larger and heavier trucks on the Texas highway system. This report describes a procedure for the selection of an optimal subset of truck-related link improvements in the highway network. This procedure is a component of an integrated network modelling methodology for the study of trucklane needs in the Texas highway network.

The link improvement selection problem is cast as a discrete network design problem with multiple improvement types per link. One of the principal features of this procedure is the definition of link improvement in terms of both capacity expansion (lane addition) and operational scheme (exclusive use by cars or trucks of both existing and added lanes). Another is the consideration of the interaction of cars and trucks in the traffic stream in solving the network equilibrium assignment problem embedded in the network design problem.

A branch and bound integer programming approach is adapted and tested for this problem and the particular features introduced by truck-related link improvement measures.

EXECUTIVE SUMMARY

The motivation for this study arises from the need to plan for and accommodate the increasing traffic of larger and heavier trucks in the Texas highway system. In particular, the construction of special lanes for the exclusive use of trucks in selected critical parts of the network is being given consideration as an approach to deal with truck-related highway problems.

This report describes a procedure for the selection of an optimal subset of truck-related link improvements in the highway network, so as to minimize overall system costs (including costs incurred by car and truck users), subject to a specified maximum level of expenditures. This procedure is part of an integrated network modelling methodology for the study of truck lane needs in the Texas highway network.

The principal features of the link improvement selection problem addressed here are: 1) the consideration of link improvement in terms of both capacity expansion and operating scheme (lane access restrictions by either cars or trucks); 2) the definition of five possible improvement options for each link; 3) modelling of the interaction between two user classes, cars and trucks, in their shared use of the common highway links; 4) the need to solve many user equilibrium network assignment problems with asymmetric interactions in the search process; and 5) the incorporation of an elaborate set of feasibility rules, including geometric, pavement and operational criteria for the addition of special truck lanes to a given existing highway link.

A branch and bound integer programming approach has been adapted and tested for the features of this particular discrete network design problem with multiple link improvement options, and tailored to the special requirements of the truck lanes problem. The structure and detailed steps of this methodology are described in this report.

The specific link improvement options are described in this report, and implementation details in terms of network representation of these improvements are explained. Also presented are the feasibility rules for

vii

each improvement type given a link's characteristics. These rules are used for screening purposes, thereby reducing the search space and enhancing the model's computational efficiency.

In the branch and bound algorithm adapted for this problem, the procedures for calculating the upper and lower bounds of the objective function at each node of the branch and bound search process take advantage of the particular features of the improvement options in this problem.

Initial testing and implementation of this link improvement selection methodology has established its usefulness for the study of truck lane needs in the Texas highway network. It is recommended that further research be conducted in order to make more effective use of the powerful capabilities of this methodology. In particular, its computational requirements could be improved by using "shortcuts" that would not significantly affect accuracy. In addition, the methodology is flexible enough to allow the consideration of a broader array of truck-related improvements (outside the scope of the present study), and to explicitly deal with multiple objectives in the link improvement selection problem.

viii

IMPLEMENTATION STATEMENT

The methodology developed in this study can assist the SDHPT in dealing with the question of special lanes or facilities for truck traffic. Its applicability is however not limited to the analysis of exclusive truck lanes. It can handle a variety of highway link improvement options, involving capacity expansion jointly with operational schemes. The latter can include any combination of lane access restrictions to either cars or trucks, for existing as well as new lanes. As such, the network modelling methodology provides a flexible tool to address a variety of measures aimed at dealing with the inevitable increases in truck and car traffic, and the increasingly dissimilar mix of vehicle classes using the highway system. While source updating and fine-tuning of truck-related measures would be necessary, the structure of the methodology offers the requisite adaptability for these tasks.

TABLE OF CONTENTS

PREFACE	. iii
ABSTRACT	. v
EXECUTIVE SUMMARY	• Vi.i.
IMPLEMENTATION STATEMENT	• j.x
LIST OF FIGURES	• xi.i.
CHAPTER I. INTRODUCTION	
Introduction	. 1
Problem Statement	. 2
Scope and Objectives	• 3
Overview	. 4
CHAPTER II. LINK IMPROVEMENTS: DEFINITION AND SCREENING	
Proposed Link Improvements	• 5
Elimination Criteria	. 7
CHAPTER III. DETAILS OF THE ALGORITHM	
The Model	. 15
Overview of Model Components	. 19
Control Parameters for Solution Strategy	. 19
Mechanisms for Search Process	. 21
Lower Bound Determination	. 23
Upper Bound Calculation	
Representation of Network Configurations	
CHAPTER IV. SUMMARY & CONCLUSIONS	
Summary of the Algorithm	. 30
Recommendations for Further Research	-
REFERENCES	. 35
APPENDIX	• 39

LIST OF FIGURES

Figure					Page
1	Link Improvement Options	•	•	•	6
2	Representation of Both Implicit Enumeration Trees	•	•	•	8
3	Example of Single Highway Representation Utilization	•	•	•	16
4	Network Representation of 3 Highways with Proposed Lane Addition $(m=3), \ldots, \ldots, \ldots, \ldots, \ldots, \ldots$	•		•	18

CHAPTER I. INTRODUCTION

Introduction

Traffic increase is an inevitable by-product of the phenomenal growth of the State of Texas. Both general and commercial traffic are increasing, in metropolitan and rural areas alike, impinging upon the ability of the current highway system to meet the future needs of the state's economy and the mobility of its residents in a safe, efficient manner. Furthermore, current trends toward larger and heavier trucks on one hand, and toward more compact fuel-efficient passenger vehicles on the other, raise serious safety considerations due to the sharing of the same roadways by vehicles with such dissimilar and extreme characteristics.

The development and implementation of economically and technically viable solutions require concerted planning efforts supported by careful analysis of anticipated truck traffic on the highway network. The implications of various proposed solutions in terms of required expenditures and resulting service levels to both trucking firms and the general public should be systematically assessed and evaluated. There exist no quick fixes to a problem of this magnitude, and the decisions and strategies developed now will have long-ranging effects on future development. Adequate methodological support for this planning activity is essential for its proper conduct.

With the expected growth in intercity traffic for the rest of this century, highway agencies will have to provide new and improved facilities for a more complex mix of vehicles. At one extreme, very large truck combinations are already affecting the design and capacity considerations of some Texas corridors. Large amounts of truck traffic on IH-35 and US-287, for example, have tended to accelerate facility deterioration rate, to jeopardize some geometric design features, and to create potentially more hazardous driving conditions in roadway operations. Importantly, all these tendencies will worsen over time.

1

Consequently, information needs to be developed that will assist the Texas State Department of Highways and Public Transportation (SDHPT) engineers in preparing construction rehabilitation and maintenance projects to accommodate the unique demands that large, heavy trucks place on highway facilities. Ultimately, this information will lead to establishing criteria and warranting procedures for the development and implementation of truckrelated facilities.

The Center for Transportation Research (CTR) at The University of Texas at Austin has conducted a study to develop a methodology for evaluating the viability of special lanes exclusively for truck use, and to determine highway links that are best suited for such improvement. The present report describes the detailed methodological aspects of a procedure developed in conjunction with this study, for the identification of an optimal subset of link improvements to be implemented under given expenditure levels. A more general presentation can be found in a final report summarizing the results of this study (Mahmassani, et al, 1985).

Problem Statement

Consider the network in the State of Texas which consists of intercity highway links shared by two user classes: cars and trucks. Based on projected future demands for travel between cities, and current trends in truck sizes and weights, it is believed that conditions can be improved through the provision of additional traffic lanes to accommodate truck The problem faced by the operating agency is to select an optimal traffic. subset from a set of proposed link improvements which will result in the minimum total travel cost to users, measured in total travel time. The constraint on link improvement is that capacity expansion of each individual link (single direction of highway) will be limited to at most one additional lane. Associated with each link improvement is a cost of construction. Furthermore, a budget is given which limits total expenditures incurred. Therefore, the optimal improvements will be determined by the allowable cost. Clearly the problem at hand is a discrete network design problem. This report presents the documentation for an integer programming computer code to solve such a problem and identify the desired subset of link improvements.

Scope and Objectives

The link selection problem addressed in this study differs from previous network design formulations in the following aspects:

- a. the problem is not only to select links for lane addition, but also to determine the best operational strategy (lane access restrictions) for that link,
- b. the existence of five possible improvement options, defined in the next chapter, instead of the usual two (build, no build) in the traditional network design problem,
- c. the need to capture the interaction between two user classes, cars and trucks, in their use of existing as well as new possible lanes,
- d. the need to solve a user-equilibrium network traffic assignment problem with asymmetric interactions to evaluate each subset of improvements (see the companion report by Mouskos et al. [1985] for further details on the assignment procedure),
- e. the incorporation of an elaborate set of feasibility rules, including geometric, pavement and operational criteria, for the addition of special truck lanes to a given highway link.

The above features introduce difficulties in the solution procedure in addition to those already present in the standard network design problem. This requires the development of special approaches, tailored to the truck lanes context, to circumvent some of these difficulties and reduce the computational complexity of the problem to a manageable level.

The objective of this research is to develop a computer code that efficiently searches the set of possible link improvement combinations to identify a subset of links and associated improvements, whose implementation cost does not exceed some specified budget, that maximizes some measure of effectiveness. In this case this measure is defined as the total travel cost savings relative to the do-nothing base-case scenario. Note that maximizing travel cost savings is equivalent to minimizing total travel costs.

This report documents the procedure developed for the above purpose. An overview is given in the next section.

Overview

Having presented the motivation for this study and the problem it addresses, the next chapter describes the types of link improvements that can be handled by the model. These improvements are defined not only in terms of lane addition, but also in terms of the scheme followed for operating the new lanes along with the existing ones. Also presented in Chapter II are the feasibility rules for particular improvement types for a link with given characteristics. These rules are used for screening purposes to eliminate unfeasible or undesirable matchings, thereby reducing the search space and enhancing the model's computational efficiency.

Chapter III presents the details of the search algorithm, which is based on the "branch and bound" method for integer programming. In particular, the special procedures for calculating the upper and lower bounds of the objective function at each node of the branch and bound search process are presented. These bounding procedures are specific to the link selection problem under consideration.

Closing remarks are presented in Chapter IV, which summarizes the algorithm along with conclusions and recommendations for future methodological refinement. A detailed flowchart of the computer code is given in the Appendix.

CHAPTER II. LINK IMPROVEMENTS: DEFINITION AND SCREENING

This chapter describes the types of link improvements that can be handled by the model. All these improvements involve the addition of a traffic lane. However, they differ in terms of the operational scheme adopted to determine the access by either trucks or cars to both the new and existing lanes. The implications of the possibility of multiple improvement options for each link on the solution procedure are discussed. This is followed by specification of the feasibility conditions, involving geometric, pavement and operational criteria, for each type of improvement given a highway link's characteristics. These feasibility conditions form the basis for screening and elimination rules that contribute significantly to reducing the search domain for the problem.

Proposed Link Improvements

Keeping in mind that each link is accessible by two distinct classes of vehicles, and that the main objective of this study is to assess the viability of special lanes exclusively for trucks, the following link improvements are suggested:

- Option 1: Expand the link by one lane and allow all traffic on entire link.
- <u>Option2</u>: Expand the link by one lane, but allow only truck traffic on <u>new</u> lane with all traffic allowed on <u>old</u> lanes.
- <u>Option 3</u>: Expand the link by one lane, but all truck traffic must use new lane with all car traffic allowed on old lanes only.
- Option 4: Expand the link by one lane, but allow only car traffic on new lane with all traffic allowed on old lanes.

Fig 1 illustrates an example of implementing each of the four options on a link which presently has three lanes. Note that the network representation followed in this study consists of specifying a directed link for each

5

	_	resen			Prop	osed	
Option 1:	-	-	-	-	-	-	-
	+	-	-	-	-	-	-
	-	-	-	-	-	-	-
Option 2:		I			I	1	
	-	-	-	x	-	-	-
	-	-	-	x	-	-	-
	-	-	-	x	-	-	-
Option 3:			_				
	-	-	-	x	0 0 0	0	o
	-	-	-	x	0	0	ο
	-	-	-	x	0	ο	0
Option 4:		r•		I	1		
	-	-	-	0		-	-
	-	-	-	0	-	-	-
	-	, —	-	0	-	-	-
	•				. 1		

Legend

- Designates lane open to all traffic
- x Designates lane open to truck traffic only
- o Designates lane open to car traffic only.

Fig 1. Link improvement options.

direction of flow on a given highway. Therefore, a particular highway section, with traffic allowed in both directions (regardless of separation), is represented as a pair of directed links, in opposite directions. Therefore, if a link, as defined here, is improved by a given option, then the sister link (i.e., the paired link in the opposite direction) must also be improved by the same option.

For a pure 0-1 integer programming problem the number of possible solution sets (or combinations of links to be improved) is equal to 2^n , where n is equal to the total number of integer variables; in this discrete network design problem, n equals the number of links being considered for capacity expansion. However, the availability of four improvement options per link increases the number of possible solution sets of 5^n (See Fig 2). Therefore, it becomes imperative that some screening rules be specified to eliminate as many solution sets as possible before exercising the branch-and-bound algorithm. These rules reflect feasibility considerations for each type of improvement, and are specified in a special subroutine called ELIMINATE, described hereafter.

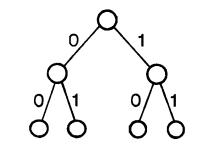
Elimination Criteria

The ELIMINATE subroutine tests for the feasibility of expanding capacity on each link of the highway network. Geometric, pavement and operational characteristics data of each link, obtained from the Highway Performance Monitoring System (HPMS), is manipulated to determine if any given link has the dimensions or features to be expanded. The following is a list of data and their respective definitions required by the ELIMINATE subroutine:

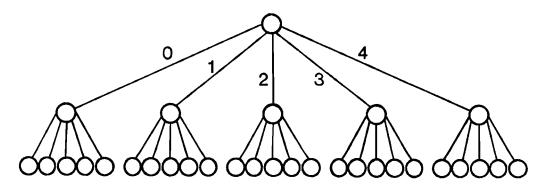
<u>Rural/Urban</u> <u>Designation</u>: Describes land use intensity of the area surroundings of a particular highway link (i) in terms of

Code	Description
1	Rural
2	Small Urban
3	Urbanized

The variable URB (i) denotes the rural/urban designation of highway link (i) in the computer program. Note that highway link (i) represents link (i) and its sister link (the opposite - direction link with which it is paired).



Tree with Two Options



Tree with Five Options

Fig 2. Representation of both implicit enumeration trees.

<u>Functional Class</u>: Describes the designated function of a particular highway link (i)

Code Description Rural 1 Principal arterial - interstate 26789 Principal arterial - other Minor arterial Major collector Minor collector Local Urban 11 Principal arterial - interstate Principal arterial - freeways or expressways Principal arterial - other 12 14 16 17 Minor arterial Collector 19 Local

IFUNC(i) codes the functional class of highway link (i) accordingly. In addition, FUNC, an array of functional classes not acceptable for lane improvement purposes is defined.

<u>Trucks/Commercial Vehicles</u>: Describes the degree of access of a particular highway link (i) to trucks/commercial vehicles.

Code	Description
1	Not a parkway - Trucks/commercial vehicles allowed.
2 3	Parkway - Trucks/commercial vehicles prohibited. Not a parkway - Trucks/commercial vehicles prohibited - all
ц	day. Not a parkway - Trucks/commercial vehicles prohibited during
7	specific periods.

TRCV(i) codes the access status of trucks/commercial vehicles for highway link (i).

<u>Number of Through Lanes</u>: Contains the prevailing number of lanes <u>in</u> <u>both directions</u> carrying through traffic on any given highway link (i) (Excludes truck climbing lanes).

This is represented by the variable LANE (i), which denotes the number of lanes in both directions for the highway (i) (i.e., link (i) and sister link).

<u>Surface/Pavement Type</u>: Contains the code that represents the type of surface on highway link (i).

Rather than list all the surface/pavement codes and descriptions, only the two types pertinent to this report are presented.

Code	Description
60	High Flexible - Mixed bituminous or bituminous penetration road on a rigid or flexible base with combined thickness (surface and base) thickness of 7 inches or more. Includes any bituminous concrete, sheet asphalt, or rock asphalt.
70	High Rigid - Portland cement concrete pavements with or without bituminous surfaces of less than 1 inch.

SURPV(i) denotes the code that represents the surface/pavement type of link (i).

Lane Width: Contains the prevailing traffic lane width (through lanes) for any given highway link (i) to the nearest whole foot.

<u>Right Shoulder Width</u>: Contains the width of the right shoulder for a given link (i) to the nearest whole foot.

RSHOW(i) equals the right shoulder width of link(i). The right shoulder width for the sister link of link(i) is the same.

Left Shoulder Width: Contains the width of the left shoulder for any given link(i) to the nearest whole foot.

LSHOW(i) equals the left shoulder width of link(i). The left shoulder width for the sister link of link(i) is the same.

Shoulder Type: Contains the predominant type of shoulder on any given highway link (i). If shoulder types differ, the right shoulder type is considered to be predominant.

Code

1	Fully surfaced
2	Stabilized
2345	Earth
4	Curb
5	None

SHOT(i) denotes the shoulder type of highway link (i) accordingly.

<u>Median Width</u>: Contains the predominant median width (excluding shoulders) measured between the through roadways of any given link (i) to the nearest whole foot.

MEDW(i) denotes the median width of link (i).

Existing Right-of-Way Width: Contains the sum of the: Median width; right shoulder width x 2; left shoulder x 2; lane width x number of lanes; plus, land adjacent to highway link (i) presently available for roadway use. ROW(i) assumes the right-of-way value for highway link(i).

<u>Is Widening Feasible</u>: Contains the code value which describes the feasibility of widening any given link (i). This variable is concerned with property <u>beyond</u> the existing right-of-way width of highway link (i). In other words, this variable concerns property <u>not</u> presently available for roadway use. Only physical features such as buildings, severe terrain, park lane, etc. are considered.

Code Description

1	No
2	Yes, less than one lane
34	Yes, one lane
4	Yes, two lanes
5	Yes, more than two lanes

WID(i) is the variable containing the code value of widening feasibility for link (i). The sister link value for this variable is the same.

<u>Speed Limit</u>: Contains the maximum speed allowed on a given highway link(i), and is denoted by the variable SPLT(i).

<u>Percent Trucks</u>: Contains the average ratio of trucks to total vehicles using a given link(i), expressed in percent using variable TRUP(i).

Note that this value is recalculated at each evaluation of a network design configuration within the traffic assignment subroutine.

In addition to the above data available from HPMS, the following variables are required by the ELIMINATE subroutine:

SHOW = The sum of RSHOW(i) + LSHOW(i) for any given link i. ISUM = The sum of MEDW(i) + 2 * SHOW for any given link i. ITOT = The difference of ROW(i) - lane(i) * laneW(i) - ISUM. CD(i) = The code value associated with link i which indicates the type of

(D(i) = The code value associated with link 1 which indicates the type of land utilized for lane addition.

<u>Code</u> <u>Description</u>

1	land utilized i	S	ITOT only.	
23	land utilized i	S	not owned presently	[WID(i)].
3	lane utilized i			
ц	land utilized j			
5	land utilized i	S	ISUM only.	

11

no lane addition options are feasible, but modified option 4
 becomes available as described hereafter.
 no options feasible.

Note that when CD(i) = 6, none of the previously mentioned lane addition options are feasible; therefore, all options except number 4 are fathomed. Option 4 is modified as illustrated by the following:

		Initial		
-	- -	 >	0 0 0	(not available)
		Modified		
	- - -	 >	0 - - 0 - - 0 - -	

In other words, trucks are prohibited from using one of the existing lanes, generally the left-most one. It was decided to allow this alternative if any given highway link(i) could not feasibly build a new lane and possessed other qualifying characteristics to be mentioned.

TPC(i,j) = the cost of utilizing code value CD(i) = j on link i. ICOST(i) = the cost of constructing any lane addition option on link i. ICON1 = the minimum width allowed to utilize CD(i) = 1. ICON2 = the minimum width allowed to utilize CD(i) = 3. ICON3 = the minimum width allowed to utilize CD(i) = 4. ICON4 = the minimum width allowed to utilize CD(i) = 5. ICON5 = the minimum width allowed to utilize CD(i) = 6.

The following variables are necessary for the implicit enumeration process followed in the search algorithm:

- S(j) = the index of the separation variable (link) at level j. (ILEVEL).
- V(i) = defines the option currently examined at link i. [V(i)=0 if no option is examined presently FREE link. V(i)=5 if all options have been already examined and discarded].
- Y(i) = defines the current system optimal solution being examined at link i. [Y(i)=0 if no option chosen at link i, Y(i)=1 if some option is chosen at link i.].
- YR(i) = defines the current user-optimal solution being examined at link i.[YR(i)=0 if no option chosen at link i,YR(i)=1 if some option is chosen at link i.].

YB(i) = defines the current best feasible solution YR(i).

The ELIMINATE Subroutine

The ELIMINATE subroutine tests link i for feasibility of lane addition. If link i fails to pass through the ELIMINATE subroutine, then the sister link of link i also is fathomed. Therefore, only one side of the highway needs to be tested in this subroutine.

- (Initial). Set ILEVEL = 0. TPC (i,7) = 0 for all links (i).
 Go to step 2.
- 2. (Initialize). Implicit Enumeration Variables for link i. If all links tested, to step 8. Read from HPMS data file (For link i).
 - TPC (i, j) for all j. where j = 1, ..., 6 (for all code types of construction).
 - IFUNC(i), TRVCL(I), ROW(i), RSHOW(i), LSHOW(i), MEDW(i),

LANE(i),LANW(i),SPLT(i),SURPV(i).

Go to step 3.

3. (Compute).

SHOW - RSHOW(i) + LSHOW(i)
ISUM = MEDW(i) + 2 * SHOW
ITOT = ROW(i) - LANE(i) * LANW(i) - ISUM
Go to step 4.

4. (Compare).

a. IF FUNC(i) ε IFUNC.

Functional class of link is not acceptable to allow for any lane addition or capacity expansion alternatives. Go to step 7. Else

b. If TRCV(I) = 1 then

Trucks/commercial vehicles are prohibited for at least part of a day, and therefore, link i is not a candidate for lane addition. Let CD(i) = 7. Go to step 7. Else

c. If LANE(i) ≤ 6 or LANE(i) is not an even number then

The number of lanes in a single direction must be less than 3 or the number of lanes in each direction must not be equal. Therefore, link is not a candidate for lane addition. Let CD(i) = 7. Go to step 7. Else

d. If SPLT(i) < 50 then

The speed limit of link i is not acceptable to allow for any lane addition or capacity expansion alternatives. CD(i) = 7. Go to step 7. Else

e. If SURPV(i) = 60 or 70 then

```
The surface/pavement type of link i .... CD(i) = 7. Go to step 7. Else go to step 5.
5.
     (Coding)
           If ITOT \geq ICON1 then
     Let CD(i) = 1. Go to step 6. Else
           If WID(i) \geq 4 then
     Let CD(i) = 2. Go to step 6. Else
           If (MEDW(i) \ge ICON2 then
     Let CD(i) = 3. Go to step 6. Else
           If URB(i)=3 and if SHOT(i) \leq 3 and If SHOW(i) \geq ICON3 then
     Let CD(i) = 4. Go to step 6. Else
           If ISUM \geq ICON 4 then
     Let CD(i) = 5. Go to step 6. Else
           If LANE(i) \geq 6 and If TRUP(i) ICON 5 and If URB(i) = 1 then
     Let CD(i) = 6. V(i) = 3. Go to step 6.
     (Set) ICOST(i) = TPC(i,CD(i))
6.
     Go to step 2.
     (ELIMINATE) ILEVEL = ILEVEL + 1.
7.
                  w(i) = -ILEVEL
                  v(i) = 5
                  yr(i) = 0
                  S(ILEVEL) = i
                  ICOST(i) = M
     Go to step 2.
```

8. (STOP) ELIMINATION subroutine is completed.

CHAPTER III. DETAILS OF THE ALGORITHM

This chapter presents the structure and detailed features of the principal components of the algorithm developed to search for the optimal subset of links and associated improvement options in the state highway network. A rigorous formulation of the model is first given, followed by a description of five principal components of the algorithm, each of which constitutes a subroutine in the computer code developed for this study.

The Model

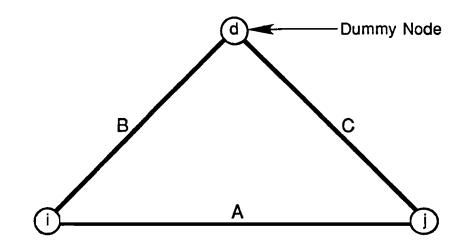
Consider a highway network with N nodes, and assume that nodes 1, 2, ..., N are origins and destinations. Let existing arcs (in single direction only) be numbered 1, 2, ..., m; dummy arcs (defined as mechanisms to keep the existing lanes distinct from the new lanes) be numbered m+1, ..., 2m; and label proposed lane additions to the network 2m+1, ..., 3m, where i is the highway link (including the opposite-direction paired sister link) under consideration, m+1 is the dummy link associated with link i, and 2m+i is the lane addition associated with link i. Figure 3 depicts an example of the highway network representation for the purpose of this study.

Assume two origin-destination matrices exist, the respective entries of which denote the amount of flow (cars and trucks, per day, respectively) that are expected to travel via highway from any given origin node to any given destination node. The scope of this report does not include discussion of solving the traffic assignment problem which exists within a discrete network design problem. However, a solution technique which solves the two-class traffic assignment problem unique to this network design problem has been developed in conjunction with this study. It is described in the companion report by Mouskos et al. (1985).

Let T[.] define the total time function to be minimized. It consists of two components $T_A[.]$ and $T_T[.]$ for cars and trucks, respectively. Define the set

 $Y = [y=(y_1, y_2, \dots, y_m | y_i=0 \text{ or } 1]$

15



- A original highway link (both directions, i.e. links i and 6m+i for cars, links 3m+i and 9m+i for trucks.
- B dummy links (valves). links 2m+i and 7m+i for cars, links 4m+i and 10m+i for trucks.
- C proposed lane additions. links 2m+i and 8m+i for cars, links 5m+i and llm+i for trucks.
- Note: i=1,...,m. where m=total number of existing lanes in single direction only

Fig 3. Example of single highway representation utilization (both directions).

The interpretation of the binary variable y_j is that if $y_j=1$, then the additional lane represented by arc 2m+1 is constructed; otherwise it is not. Let c_i be the cost of construction associated with project 2m+i, i=1,...m. The variable c_j is defined by link i and the code value CD(i) derived in subroutine ELIMINATE, described in Chapter II. The total construction cost associated with any $y_j \in Y$ is $\sum_{i=1}^{m} c_i y_i = C*Y$. If the budget B represents the maximum allowable expenditure for network improvement, then $C*Y \leq B$. In addition, define the binary sets

 $D=[d=(d_1, d_2, \dots, d_m)|d_i=0 \text{ or } 1]$

$$K = [k = (k_1, k_2, \dots, k_m) | k_i = 0 \text{ or } 1]$$

The variable d_i is exclusively defined by the ELIMINATE subroutine. If CD(i)=7, then $d_i=0$; otherwise $d_i=1$. The variable k_i is initially defined by the ELIMINATE subroutine, but its value may change within the implicit enumeration process. If V(i)=5, $k_i=0$, if V(i)=0, then $k_i=0$ or 1; otherwise $k_i = 1$.

The network design problem to be solved is

(J) Min T[.]

Subject to

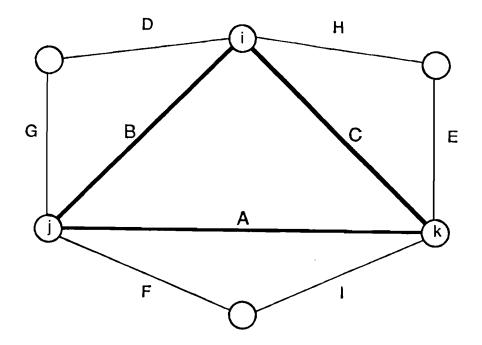
yEY
dED
kEK

$$d_i - y_i \ge 0$$
 for $i=1,...,m$
 $d_i - k_i \ge 0$ for $i=1,...,m$
 $y_i - k_i = 0$ for $i=1,...,m$

C*Y <u><</u> B

where m equals the total number of existing links in one direction. (See Fig 4).

The branch-and-bound algorithm, which is a widely used procedure for solving integer programming problems, was utilized to develop a solution technique for solving the above problem (J). The subroutines unique to this network design problem are first described, followed by a summary of the algorithmic steps.



A,B,C -- represent existing car and truck links in both highway directions.

Links represented are:

Link i -- existing link open to car traffic in one direction. Link 3m⁺i -- existing link open to truck traffic in one direction. Link 6m⁺i -- existing link open to car traffic in opposite direction. Link -9m⁺i -- existing link open to truck traffic in opposite direction.

D,E,F -- represent dummy car and truck links in both highway directions.

Links represented are:

Link m⁺i -- dummy link associated with link i. Link 4m⁺i -- dummy link associated with link 3m⁺i. Link 7m⁺i -- dummy link associated with link 6m⁺i. Link 10m⁺i -- dummy link associated with link 9m⁺i.

G,H,I -- represent proposed lane additions for cars and trucks in both highway directions.

Links represented are:

Link 2m+i -- proposed lane addition associated with link i. Link 5m+i -- proposed lane addition associated with link 3m+i. Link 8m+i -- proposed lane addition associated with link 6m+i. Link 11m+i -- proposed lane addition associated with link 9m+i.

Fig 4. Network representation of 3 highways with proposed lane additions (m=3).

Overview of Model Components

Within the branch-and-bound search process, it is necessary to evaluate the objective function at each stage of the process, and obtain both upper and lower bounds for this function. In the network design problem, calculating the objective function requires the solution of a traffic assignment problem. Depending on whether one is seeking a lower bound or an upper bound, the system optimal assignment rule (on a "fully improved" network, described hereafter) or the user equilibrium rule are adopted, respectively, for traffic assignment. This is handled in subroutines RELAX and ROUND, respectively.

In addition, at each stage of the search process, it is necessary to specify a mechanism for proceeding in the search. In other words, a search direction must be specified. Subroutine MAX handles this aspect.

Clearly, there are many possible options to the engineer or analyst using the model, from pre-empting certain improvement types, to the specification of the search mechanisms, and many other details of the algorithm. This user interface is handled by subroutine INITIAL.

Another major interface is that with the traffic assignment models; the link improvements corresponding to a particular solution (subset of links) must be translated in terms of inputs that the traffic assignment models can recognize. This interface is accomplished through the link performance (or cost) functions associated with each link in the representation described earlier. In particular, when a lane addition is not included in the solution currently being tested, its corresponding link (numbered [2m+i] for an addition to original highway section i) is "deactivated" through the specification of a very large travel cost value. This mechanism is described in detail in the description of subroutine CHANGES.

The remaining sections of this chapter describe these subroutines, in the following order: INITIAL, MAX, RELAX, ROUND and CHANGES.

Control Parameters for Solution Strategy

Subroutine INITIAL allows the user to implement a solution strategy for the problem to be solved. The CPU time required to solve a discrete network design problem will generally depend upon the strategy to be utilized. The following variables are defined to control the solution strategy:

BP, which is a control parameter that equals the number of improvement options not to be available as part of the solution set.

IOPT(k) equals 0 if link improvement option k is available; otherwise IOPT(k)=k.

IDP is a control parameter that determines the option to be utilized during lower bound calculations.

ITP determines the option to be utilized during upper bound calculations.

Steps of Subroutine INITIAL:

1. (Input). Values for the following variables.

IMAX. Used in subroutine MAX. User selects the criterion for determining the separation variable.

- Code Description
- 1 Utilizes AADT values 2 Utilizes VCRT values 3 Utilizes TRUD values
- PER: Used in subroutine ROUND. User may select a percent of deviation from the optimal solution to the problem which would be acceptable. PER allows the algorithm to stop if an acceptable feasible solution is found. PER may reduce the CPU time of the algorithm.

Go to step 2.

- 2. (CHOOSE). Options to be available in the solution process for the particular problem. (Obviously, at least one option must be chosen.) Let BP=0. Go to a.
 - a. If Option 1 is not to be utilized, then
 - SET BP=BP+1. IOPT(1)=1. Go to b.
 - b. If Option 2 is not to be utilized, then
 - SET BP=BP+1. IOPT(2)=2. Go to c.
 - c. If Option 3 is not to be utilized, then
 - SET BP=BP+1. IOPT(3)=3. Go to d.

20

d. If Option 4 is <u>not</u> to be utilized, then SET BP=BP+1. IOPT(4)=4. Go to step 3.

3. (DECIDE). Options to be utilized for bounding procedures.
 a. If BP=0 or IOPT=0, then
 SET ITP=1. IDP=1.

If option 1 is available, then option 1 will be utilized for both lower and upper bound procedures. Go to step 5. Else b. If BP=3, then only one option (k) is available. Therefore, SET ITP=k. IDP=k. Go to step 5. Else go to step 4.

(PICK). Option (k) to be utilized for upper bound calculations.
 SET ITP=k. IDP=1.

If more than one option is available excluding option 1, then the user may pick the option to be utilized for upper bound calculations. However, option 1 <u>will</u> be utilized for lower bound calculations.

Go to step 5.

5. (STOP). The INITIAL subroutine is complete. Start the implicit enumeration process.

Mechanisms for Search Process

Subroutine MAX is responsible for two functions. Using data calculated by the traffic assignment program, a heuristic approach was developed to: First, choose from the set W^O (all i where W(i) = 0) the <u>next</u> link i to be examined by the implicit enumeration algorithm; and, second, create an array which contains an order of preference for adding lanes to the free links (links not yet chosen) of the network. The order is most preferable to least preferable.

In addition to the variables defined in subroutine ELIMINATE, there are additional variables required by subroutine MAX:

<u>Average Annual Daily Traffic</u>: Contains the AADT value of link i as calculated within the traffic assignment program for any given network configuration.

<u>AADT(i)</u> - Equals the average annual daily traffic for link i during traffic assignment for a given network configuration.

 \underline{IMAX} - contains the code equivalent for the heuristic chosen by the user of the program.

	<u>Co</u>	ode	Des	script	ion							
	1 2 3		VCRT	value	s will	be	utilized utilized utilized					
JMAX	_	Equals	s the	next	link i	to	be examined	by	the	implicit	algorithm.	

Order: The array that contains the preference list for the free links of the network.

ORDER(i) - Equals the position of preference for adding a lane to link(i). If order(i)=99, then link i is not a free link.

Steps of the MAX Subroutine:

- (READ). From input Data: The value if IMAX. From the traffic assignment of the network configuration defined by algorithm at the present time: The arrays AADT, VCRT, and TRUP. Go to step 2.
- 2. (DETERMINE). The maximum value for the given link.

a. If IMAX = 1, then Let AADT(i) = MAX [AADT(i), AADT (6m+i)] For all i, where i=1,...,m. Go to step 3. Else b. If IMAX=2, then Let VCRT(i) = MAX [VCRT(i), VCRT(6m+i)]
for all i, where i=1,...,m.
Go to step 3. Else c. If IMAX = 3, then Let TRUP(i) = MAX [TRUP(i), TRUP (6m+i)]For all i, where i-1,...,m. Go to step 3. 3. (Find). The free link with the maximum value of the chosen array. a. If IMAX = 1, THEN JMAX = i where i = MAX [AADT] for i=1,...,m and W(i)=0. b. If IMAX = 2, then JMAX = i, where i=MAX[VCRT] for i=1,...,m and W(i)=0. If IMAX = 3, then с. JMAX = i, where i=MAX [TRUP] for i-1,...,m and W(i)=0. Go to step 4.

4. (ORDER). Excluding the link JMAX, all links i where W(i)=0: arrange the remaining free links [W(i)=0] in the array order, where ORDER(i)=1, if link i contains the <u>maximum</u> value of the chosen array [be it AADT, VCRT, TRUP] for the remaining free variables, etc. ORDER(i) = M if link i is no longer a free variable. Note: M is some large number.

Lower Bound Determination

The generation of a lower bound on the values of the successors of any given node in the tree structure of the search process is one of the primary concerns in the branch and bound technique. Each node of the branch and bound tree corresponds to a solution, or subset of link improvement. This solution is "complete" if all links are set at a value denoting some improvement type, or no improvement; a "partial" solution is one where some links remain to be set, and are therefore called free links (or free decision variables). The principal difficulty in computing lower bounds in the network design problem arises from the well-known "Braess' paradox", which says that the total travel cost associated with the user equilibrium flows on a network may increase when one or more links (or link improvements) are added to a network (see Sheffi [1984] for a discussion of this phenomenon). A method has been devised to circumvent the occurrence of Braess' paradox in this problem.

Subroutine RELAX computes a lower bound for an incomplete (partial) solution. The mechanics of this subroutine are comparable to the lower bound technique suggested by Leblanc (1973). Recall that when solving a minimization problem, a lower bound is determined by some relaxation of the model. Like Leblanc, the lower bounds are calculated by relaxing the budget constraint. The distinction of this subroutine is generated by the fact that five (not two) alternatives exist for each link.

With respect to Braess' paradox, Leblanc proposed to compute lower bounds by allowing all <u>free</u> links to build new lanes (regardless of cost), and then, solve the resultant network configuration by utilizing the system optimal traffic assignment. Indeed, Subroutine RELAX computes lower bounds by allowing all free links to construct new lanes, and ultimately, solve the resultant network configuration. However, problem (J) allows basically four alternatives for each new lane constructed. Furthermore, subroutine INITIAL allows the user to disregard certain options, if desired. Therefore, a justifiable course of action had to be devised specifically for problem (J). The decisions to be presented herein are based on the premise that the least constraining problem must contain the best solution, since constraints limit the size of the feasible region for a given problem. This least constraint situation is achieved in this study using improvement option 1, where both cars and trucks are allowed to use the additional lane.

The output of subroutine RELAX then is the quantity VALUE1, which equals the total travel time associated with the solution to the relaxed system optimal program, thereby providing a lower bound for the current branch and bound node.

Steps of the RELAX Subroutine:

- (SET). All free links to one ZB equals the best feasible solution so far determined. If W(i)=0, then Y(i)=1. Go to step 2. Else Y(I)=YR(I). Go to step 2.
- 2. (DETERMINE). The option to be implemented on the free links.

a. If IDP=1, then

All free links will utilize option 1: All traffic allowed on new lane.

Go to step 3. Else

b. If IDP=2, then

All free links will utilize option 2. -- <u>only</u> trucks allowed on new lane.

Go to step 3. Else

c. If IDP=3, then

All free links will utilize option 3. All trucks and only trucks must use the new lane. $\overset{All}{-}$

Go to step 3. Else

d. If IDP=4, then

All free links will utilize option 4. -- <u>only</u> cars allowed on new lane. Go to step 3.

3. (SOLVE). Traffic assignment for given network configuration using system optimal technique.

Note: Unique network configuration is created by subroutine CHANGES, which is called by the system optimal subroutine. Retrieve VALUE1 from system optimal subroutine.

4. (COMPARE). VALUE1 \geq ZB, then Backtrack. Else return to calling subroutine.

Upper Bound Calculation

Subroutine ROUND computes the user-optimal assignment for all feasible solutions (complete or incomplete). If a solution is incomplete, this subroutine will create the best feasible network configuration which is a subset to the partial solution given in order to estimate an upper bound.

It yields the quantity VALUEUP, which equals the total travel time associated with the flow pattern that satisfies the user equilibrium conditions in the network.

Steps of Subroutine ROUND:

(Set). As many free links to one as allowed by the budget. The "bang for the buck" approach is utilized herein with ranking determined by the value of order (i). Preference given to lower 1. order (i) values. Let I = Min [order [i]], If order [i]=99, go to step 2. Remove i from list kkl=kkl+1. SX(kk1)=1If BCOST + ICOST(i) < BUDGET, then Y(i)=1 WX(i)=kkl VX(i)=ITP ELSE Y(i)=0 WX(i) = -kklVX(j)=5 Go to step 3. (Let). For all fixed variables. Y(i)=YR(i) 2. $\overline{WX(1)} = W(\overline{1})$ VX(1)=V(1) SX(/W(i)/) = /W(i)/Go to step 3. 3. (Complete). A solution that contains significant values for the arrays: Y, SX, WX, VX. If arrays are incomplete, then Go to step 1. Else go to step 4.

4. (Solve). The traffic assignment for the unique network configuration defined by the aforementioned arrays. The user-optimal traffic assignment technique is utilized.

Note: network configuration is created by subroutine CHANGES - which is called by the user-optimal subroutine.

Retrieve VALUEUP from user-optimal subroutine.

5. (Compare). VALUEUP to ZB.

If VALUEUP \geq ZB, then

Find next separation variable (call subroutine (MAX).

Else go to step 6.

6. (Replace). ZB with VALUEUP.

Set FF=0. ZB=VALUEUP.

YB=Y **∆B=∆**X ₩B=₩X SB=SX BBUDEBCOST

Go to step 7.

7. (Check). If optimal (sufficient) solution exists. (If [VALUEUP-(21*(per/100))] < Z1, then Set FF=1 OPTIMAL SOLUTION FOUND (call subroutine OPTIMAL) Else if VALUEUP = VALUE1, then Backtrack (call subroutine BACKTRA) Else find next separation variable (call subroutine MAX).

Representation of Network Configurations

Subroutine CHANGES acts as the interpreter between the branch and bound program and the traffic assignment program. Subroutine CHANGES is called only by the traffic assignment programs: System optimal and user-optimal. This subroutine makes the necessary changes to the original network in order to produce a unique network configuration. Subsequently, the calling subroutine (the traffic assignment program) will compute the total travel time (either system optimal or user-optimal) used by the completely specified network configuration.

The proper communication between the integer program and the traffic assignment program involves the following variables:

C(k) - the traffic capacity of link k. RL(k) - the cost (travel time) to the traffic for using link k. T(k) - the type of traffic allowed on link k. where k=1... 12m, equals total number of existing links in single direction only.

These quantities specify the parameter values for the travel time function (link performance function) utilized for link k. A detailed discussion of these functions and their parameters can be found in the companion report by Mouskos et al. (1985).

An initial value for each of these quantities is set by the original network configuration. If a change is made to link k (i.e., a lane addition option has been chosen), then adjustments to these quantities are made, thereby providing the principal mechanism for implementing and representing changes to the network.

The specification of these quantities is as follows:

$C(\underline{i})$ = estimated capacity of existing link i.
$C(\underline{m+i}) = C(i)/LANE(i)$
$C(\underline{2m+i}) = C(i)/LANE(i)$
RL(<u>i</u>) = some estimated value for traversing link i, i=1,,m
<pre>RL(<u>i</u>) = some estimated value for traversing link i, i=1,,m RL(<u>m+i</u>) = RL(<u>m+i</u>) =</pre>
RL(<u>2m+i</u>)= some estimated value for traversing link 2m+i (new lane) if proposed lane addition is constructed, i=1,,m
<pre>T(k) = 0 if traffic on link k is restricted to one class of vehicles (cars or trucks) 1 if traffic on link k is open to both vehicle classes</pre>

where

<u>i</u> denotes existing links [i,3m+i, 6m+i,9m+i]

m+i - dummy links [m+i, 4m+i, 7m+i, 10m+i]

2m+i - proposed lane additions (2m+i, 5m+i, 8m+i, 11m+i]

In addition, variable IBND is defined to indicate which link improvement option is to be implemented if a given link is still free, such that

IBND = { ITP if called by user-optimal subroutine. IDP if called by system-optimal subroutine.

Steps of Subroutine CHANGES:

1.	(Determine).	If existing link is being changed $[i=1,\ldots,m]$ If $y(I)$, then go to step 2. Else	

Go to next I. If all links checked, then go to step 8.

2. (Decide). Appropriate option chosen.

If CD(I)=6, then go to step 3. Else

```
If [W(I)=0, and IBND=1] or V(I)=1, then go to step 4.
                                                               Else
     If [W(I)=0 and IBND=2] or V[I]=2, then go to step 5. Else
     If [W(I)=0 and IBND=3] or V[I]=3, then go to step 6.
                                                             Else
     If [WL(I)=0 and IBND=4] or v[I]=4, then go to step 7.
3.
     (No Lane Addition). Trucks prohibited on left lane.
     Set RL(m+i)=0. RL (7m+i)=0
     Let C(i) = C(i) - some constant (K)
     C(3m+i) = C(i) - K
     C(6m+i) = C(i) - K
     C(9m+i) = C(i) - K
     Go to step 1.
4.
     (Option 1). All traffic allowed on new lane.
     Set RL (m+i) = 0
     Go to step 1.
5.
     (Option 2). All traffic allowed on new lane.
     Set RL (4m+i) = 0 RL (10m+i)=0.
     Go to step 1.
6.
     (Option 3). All trucks and only trucks allowed on new lane.
     Set T(3m+i)=0. RL(3m+i)=M.
        T(9m+i)=0. RL(9m+i)=M.
RL(4m+i)=0. RL(10m+i)=0.
C(i)=C(i) + C(2m+i)
         C(6m+i)=C(i) + C(2m+i)
     Go to step 1.
     (Option 4). Only cars allowed on new lane.
7.
             RL(m+i)=0. RL(7m+i)=0.
     Set
     Go to step 1.
8.
     (Complete). Unique network configuration is defined.
     Return to appropriate traffic assignment program.
```

CHAPTER IV. SUMMARY & CONCLUSIONS

An important element of the network modelling methodology to identify truck lane needs in the Texas highway network is the algorithm to solve a five-option discrete network design problem. This algorithm, described in this report, was developed by extending the methodologies of Leblanc (1973) and Jensen (1980, 1983). Leblanc showed how the two-option discrete network design problem could be solved in a manner that avoids contradictions due to Braess' Paradox. Relaxing the initial problem in two stages guaranteed Leblanc that, indeed, lower bound solutions were being properly identified. The two stages utilized by Leblanc were:

- 1) Relax the budget constraint -- all free variables (links) will assume a value of 1 [i.e., $u_i=1$ if W(i)=0], meaning that all free links will be improved.
- 2) Solve the resultant network configuration with the system optimal traffic assignment model.

Unfortunately, the first stage of Leblanc's lower bound calculation strategy becomes too general for the five-option problem. Stage 1 assumes a unique improvement to be constructed on link i if $u_{i=1}$. However, in the five-option problem addressed here, there actually are four distinctly different improvements:

- 1) add a new lane for all traffic use. [V(i)=1].
- 2) add a new lane for truck traffic use only. [V(i)=2].
- 3) add a new lane for all trucks and only trucks. [V(i)=3].
- 4) add a new lane for car traffic use only. [V(i)=4].

Therefore, to calculate lower bounds for the five option problem, the first stage of Leblanc's methodology needed to be modified appropriately. This modification was based on the observation that option 1 (all traffic allowed on new lane) yielded the least constrained problem. It is actually possible to obtain the patterns that would exist under the other improvement options as a solution to the system optimal assignment problem (using option 1 for free links), if indeed such a solution is optimal. Option 1 was therefore the choice adopted for all free variables when solving the relaxed problem. However, if only one option (k) were to be available in the final

29

feasible solution, then option (k) would be the choice of the free variables. The approach then consists of:

- Relax the budget constraint; all free variables (links) will assume a value of 1 [i.e., ui=1 if W(i)=0].
- 2) If more than one option is available, then option 1 will be the choice for all free variables. If only one option (k) is available, then option (k) will be the choice of all free variables.
- 3) Solve the resultant network configuration with the system optimal traffic assignment model.

Note that if more than one option is available, but option 1 is not one of available choices, option 1 will still be used for lower bound calculations because it could not be adequately proven which of the remaining options is least constraining.

Jensen developed a branch-and-bound implicit enumeration solution technique complete with unique notation for computer implementation (two options) of Operation Research type problems. However, Jensen noted that only certain subroutines would need to be added or changed in order to solve the specific problem identified by the user of such a technique. The specific subroutines defined for the five option discrete network design problem were: ELIMINATE, INITIAL, MAX, RELAX, ROUND, and CHANGES. These subroutines have already been described in detail. A brief description of the functions of the common subroutines are deferred to Appendix A.

The integer program presented in this report was coded in FORTRAN on a CDC 6400 computer.

Summary of the Algorithm

USERASG - user-optimal traffic assignment subroutine. SYSTMAS - system optimal traffic assignment subroutine.

- Call subroutine INITIAL.
 Go to step 2.
- 2. Call subroutine USERAG Solves the traffic assignment problem for the original network configuration. The solution will serve as the initial best feasible solution (ZB).

Let ZB = VALUEUP. Go to step 3.

3. Call subroutine ELIMINATE.

30

Set CC=0. Go to step 4. Call subroutine RELAX. 4. IF FF=1, then go to step 15. If CC=0, the let Z1=VALUE1 If Z1 > ZB, then STOP. No network configuration yields a better solution that the original network. Else Set CC=1. If VALUE1 \geq ZB, then go to step 13. Else go to step 5. 5. Call subroutine SETUP. If ST=0, then go to step 15. Else go to step 6. 6. Call subroutine MAX. Go to step 7. Call subroutine SELECT. 7. Go to step 8. 8. Call subroutine SEPARAT. If FF=1, then go to step 15. If JM-0, then go to step 13. Else go to step 9. 9. Call subroutine SET. If FF=1, then go to step 15. If S(KL)=M, then go to step 11. Else S(KL))=JM. Go to step 10. 10. Call subroutine FEASIBL. If FF=1, then go to step 15. If BUD > BUDGET, the FF=BIG1. Go to step 13. Else set FF=0. Go to step 11. 11. Call subroutine RELAX. If FF=1, then go to step 15. If VALUE1 \geq ZB, then go to step 13. Else go to step 12. 12. Call subroutine ROUND. If FF=1, then go to step 15.

If VALUEUP \geq ZB, then go to step 6.

32

```
Else set FF=0. ZB=VALUEUP.
If [VALUEUP -Z1*(per/100)] \leq Z1, then FF=1. Go to step 15.
Else if VALUEUP = VALUE1, then go to step 14.
Else go to step 6.
```

- 13. Call subroutine BACKTRA. [If at any time part b applies go to step 15.]
 a. If FF=1, then to to step 15.
 b. If KL=ILEVEL, then let FF=1. Go to step 15.
 - c. If FF=BIG1, then to to step 10.
 - d. Go to step 14.
- 14. Call subroutine RESET.If FF=1, then go to step 15.Else go to step 8.
- 15. (End). Call subroutine (OPTIMAL) Enumeration is complete. A detailed flowchart review of this algorithm is listed in Appendix A.

Recommendations for Further Research

The methodology developed in this study can be very useful in assisting highway agencies deal with the question of special lanes or facilities for truck traffic. However, its applicability is not limited to the analysis of exclusive truck lanes. As indicated, it can handle a variety of highway link improvement options, involving capacity expansion jointly with operational schemes. The latter can include any combination of lane access restrictions to either cars or trucks, for existing as well as potential new lanes. As such, the network methodology provides a flexible tool to address a variety of measures aimed at dealing with the inevitable increases in truck and car traffic, and the increasingly dissimilar mix of vehicle classes using the highway system. While some updating and fine-tuning of the methodology for this broader array of truck-related measures would be necessary, the structure of the methodology offers the requisite adaptability for these tasks. furthermore, the methodology can be extended to consider a finer categorization of vehicle types, as an alternative to the current two-class division.

In order for these tools to be effective in supporting the planning and engineering process, and to fulfill their potential usefulness, there are a number of questions that can benefit from further investigation. As is usually the case with methods designed to deal with large-scale complex transportation network problems, various aspects of the systems under consideration are understood to differing degrees. Of particular concern for problems involving trucks is the nature of the interaction in the traffic stream among various types of vehicles. This is still a poorly understood phenomenon in the transportation engineering field, and is a high priority research item in this area. While outside the scope of this particular study, a pilot effort along these lines was initiated in conjunction with this work. In particular, a promising data set from FHWA was acquired and subjected to a preliminary assessment. The conclusion is that it can form the basis of a more focused study on this problem.

The methodology described in this report has been successfully implemented and tested on the Texas highway network. However, the computational requirements of this algorithm can be extensive. For this reason it is recommended that "shortcuts" be explored to reduce these requirements without significantly sacrificing accuracy. Such heuristic rules have been explored in this study; however, there remains much to be done in this area, particularly along the lines of the numerical testing of possible rules that are thought to have a potentially significant impact on computational cost. Such rules were identified in this report in conjunction with the algorithmic details.

A particularly interesting extension of this work would be to formulate and solve the link improvement selection problem as a multi-objective problem. Two principal objectives here are the total travel cost incurred by trucks versus that incurred by passenger cars. In the current model, those are given equal weight in the total cost objective function. It would be informative to vary these weights, or to examine the two objectives separately, assessing the degree to which they might be in conflict. This would allow the consideration of questions of equity and distributional impacts, particularly in the context of cost allocation studies.

33

REFERENCES

- 1. Jensen, P. A. <u>Microsolve/Operations Research</u>: <u>Introduction to</u> Operations Research, Holden-Day (1983).
- Jensen, P. A., and Barnes. <u>Network Flow Programming</u>, John Wiley & Sons (1980).
- 3. Leblanc, L. J. "Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems," Ph.D. Dissertation, Dept. of Industrial Engineering and Management Sciences, Northwestern University (1973).
- 4. Mahmassani, H. S., Walton, C. M., Mouskos, K., Massimi, J. J., and Levinton, I. "A Methodology for the Assessment of Truck Lane Needs in a Highway Network", CTR Report 356-3F, Center for Transportation Research, The University of Texas at Austin (1985).
- 5. Mouskos, M., Mahmassani, H. S., and Walton, C. M. "Network Assignment Methods for the Analysis of Truck-Related Highway Improvements", CTR Report 356-2, Center for Transportation Research, The University of Texas at Austin (1985).
- 6. Sheffi, Y. <u>Urban Transportation Networks</u>: <u>Equilibrium Analysis with</u> <u>Mathematical Programming Methods</u>, Prentice-Hall (1984).

APPENDIX

Introduction

Herein a detailed flowchart of all the subroutines required by the integer program is introduced. The purpose of each subroutine as well as the definition of all variables used are presented. The form of the flowcharts follows the standard format of Jensen (1978).

ELI	IMIN
-----	------

INITIAL

MCAR=0; ILEVEL=0 DATA(IFUNC(I),I=17),ICON1,ICON2,ICON3,ICON4,ICON5 D0 1, I=1, 999 (END=)

// READ FUNC(I),TRCV(I),LANE(I),MEDW(I), // RSHOW(I),LSHOW(I),ROW(I),LANW(I),WID(I), // SHOT(I),SPLT(I),SURPV(I),URB(I) // MCAR=MCAR + 1

DO 11, N=1, MCAR

 	S(I) ICOS	=V(I)=YR(I)=W(I)=Y(I)=CODE(I)=YB(I)= T(I)=0
11		DO 11,M=1,6
11	11	READ TYPCOST (N,M)

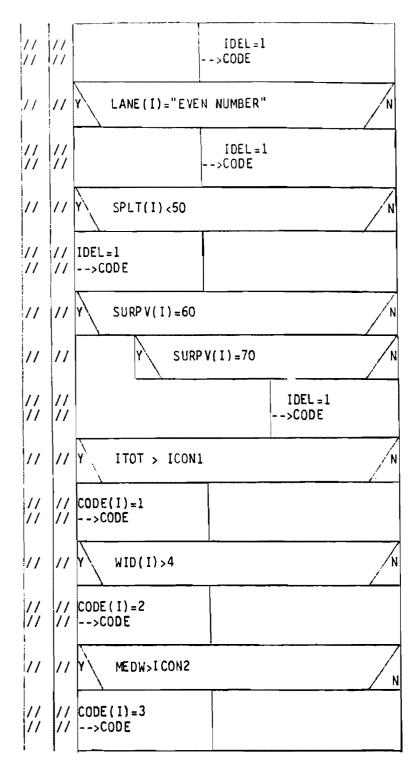
TEST

	DO 3, I=	1, MCAR	¢
// //	SHOW=RSH ISUM=MED ITOT=ROW	DW-ISUM=ITOT=0 DW(I) + LSHOW(I) DW(I) + Z * LANW(I) - ISUM D(I) LANE(I) * LANW(I) -ISUM L, J=1,7	
//	11	FUNC(I)=IFUNC(J)	N
 	// IDEL //>0	=1 CODE	
//	// Y	TRCV(1)=1	N

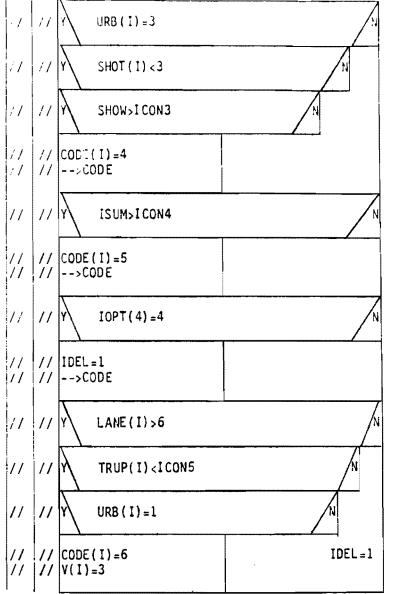
Subroutine ELIMIN

Purpose: To eliminate as many link candidates as possible from consideration of lane addition. Furthermore, to define the construction type and cost associated for each link not eliminated. MCAR = Number of links open to cars (in one direction) ILEVEL= Current level on tree (Top is equal to zero) IFUNC(I)=Functional classes of links not allowed for expansion. ICON1 = Minimum width required for construction type code (I)=1 of land owned but not utilized for highway use. ICON2 = Minimum width of median for construction type -code(I)=3ICON3 = Minimum width or shoulder for construction type code (I)=4.ICON4 = Minimum width of sum of median and

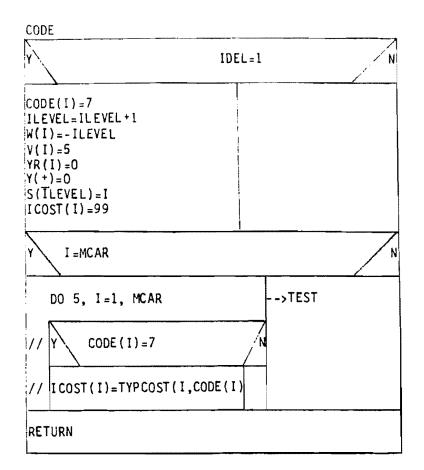
- shoulders for construction type code (I)=5
- ICON5 = Maximum percent of trucks allowed for further consideration of construction type code (I)=6
- FUNC(I) = Functional class of link I.
- TRCV(I)= Trucks/commercial
 vehicles allowed on
 link I.



```
LANE(I) = Number of lanes of
         link I (both
         directions included)
MEDW(1) = Median width of
         link I.
RSHOW(I)=Right shoulder
         width of link I.
LSHOW(I)=Left shoulder width
         of link I.
ROW(I) = Total right-of-way
         width of link I
         (both directions
         included)
LANW(I) = Lane width of link
         Ι.
WID(I) = Widening
         feasibility of link
         Ι.
SHOT(I) = Shoulder type of
         link I.
SPLT(I)= Maximum speed limit
         of link I.
SURPV(I)=Surface/pavement
         type of link I.
URB(I) = Rural/urban
         designation of link
         Ι.
TYPCOST(N.M) = Cost of
        building lane
        addition type M on
        link N. (Lane
         additional type
         refers to the land
         to be utilized for
         the purposed of
         construction)
ICOST(I)=Cost of lane
         addition on link I.
S(.)=
         The array showing
         the separation
         variables of the
         tree. S(i) is the
         index of the
         separation variable
         at level i.
```



- YR(.)= The array giving
 feasible solution to
 the user optimal
 problem
- Y(.)= The array giving a solution to the system optimal problem.
- W(.) =The array which indicates the status of link I. If W(i)=0, then link I is free (link I has not been committed to building a lane addition as yet. If w(i)=k, then link I is committed to building a lane addition at tree level. If w(i) = -k, then link I is committed to not build a lane addition at level A.
- YB(.)= The array giving the best solution.
 - Code(i)=The array which indicates the type of lane addition which will be utilized if indeed link I is recommended to have a lane addition. but, if code(I)=7, then no lane addition will be allowed on link I.



- IDEL = 0-1 variable which indicates if a given link is eligible to contruct a lane addition. IDEL = 1 means no.
 - HOW = Some of median width and show (in both directions)
- ITOT = Total right of way
 minus ISUM minus the
 product of the no.
 of lanes and the
 width of the lanes
 for a given link.

INITIAL

Reads appropriate data for the given problem and initializes variables to zero TEST Determines if a given link is eligible for lane addition by testing the characteristics of a link against minimum (maximum) requirements specified by design engineers. Furthermore, if a link is determined to be eligible for lane addition, the type of lane addition to be utilized as well as the cost of construction will be set.

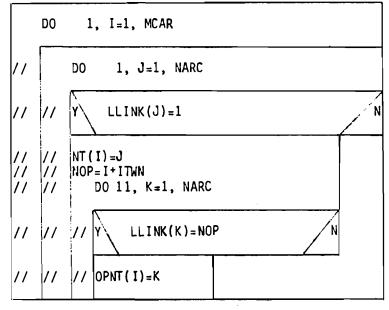
CODE

Sets the appropriate values determined by test routine to a given link.

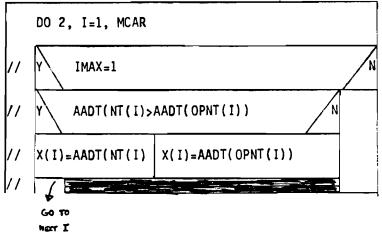
INITIAL

MIN=	=0;AMIN=0;ITWN=MCAR*6;IC=0;LOC=1;JMAX=0;KTY=0 D0 16, I=1, MCAR
// // //	N(I)=0 X(I)=0 NT(I)=0 OPNT(I)=0 ORDER(I)=0

TWINS

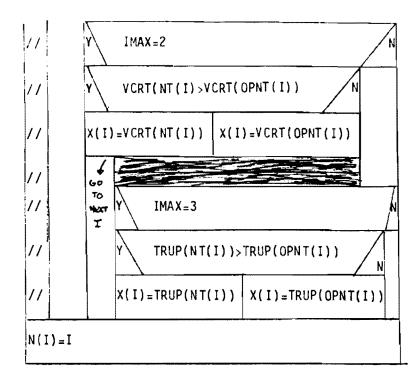


OPTION



SUBROUTINE MAX

Purpose: To determine the next link to examine for the purpose of possibilty of lane addition. MIN = Temporary storage of a given link number. AMIN = Temporary storage of criteron value for given link m.M. ITWN = MCAR 6Counter for order to IC = remaining free links LOC = Variable which indicates if next link to be examined has been chosen yet LOC = 1 No LOC =0 Yes JMAX = Variable which indicates the next link to be examined. N(I) = Array of MCARlinks. I = 1 to MCAR X (I) = Array of criterion values for MCAR links. NT (I)=Identification of MCAR link I in set of NARC links OPNT (I)=Identification of MCAR link (opposition direction of I) in set ORDER(I)=Identification of link I order (if free). If order +) = 99 link I already examined. NARC = Total number of links defined in traffic assignment subroutines. LLINK (J)=Identification of MARC link I J corresponding to MCAR Link I NOP = Identification of link I opposite direction Link I (MCAR)



ORDERS

	N=X(I) I=N(I) DO 4, J=1, MCAR	
11	Y X(J)>AMIN	/
 	DUM=X(J) NDUM=N(J)	
 	X(J)=AMIN N(J)=MIN	
// //	AMIN≖DUM MIN ≖NDUM	

```
IMAX = Criterion supplied by
        user to determine next
        link 1,2,3.
AADT(I)=AADT value of link I.
VCRT(I) =Volume to capacity
        ration of link I.
TRUP(I) =Truck percent of
        link I.
DUM =
NDUM =
```

INITIAL

Defines value of ITWN. Initializes LOC = 1 and all other significant variables to zero.

TWINS

Finds link I of MCAR set in link array of narc set. also finds opposite direction of link I in llink array of Narc set.

OPTION

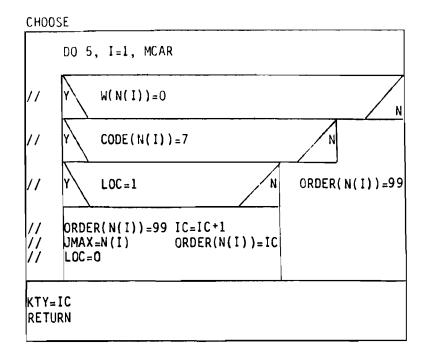
Depending on the value of IMAX sets the value of x(I) to larger value Link I and Link I (opposite direction).

ORDERS

Rearranges the order to links (MCAR in N(.) array from maximum value x(I) to minimum value of x(I). [For links (max) I].

CHOOSE

Chooses the first link in N(.) array (whose order (.) array value does not equal 99) to be the next link to be examined. Sets such M(I) value equal to JMAX. Creates an order (.) array for remaining free links.

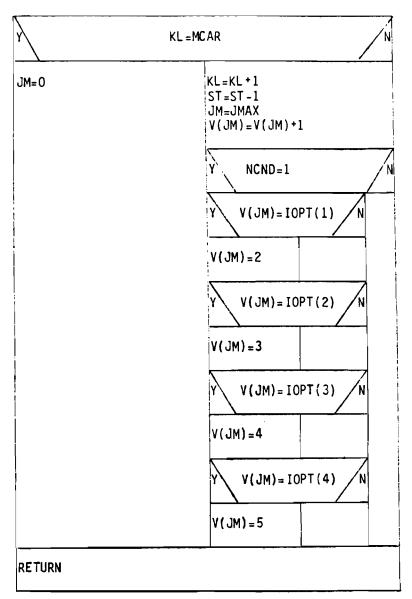


COUNT=ST=FF=BUD= KL=ILEVEL ST=MCAR-KL	:JM=BBUD=0	
Y	ST=0	N
FF=1 OPTIMAL		
RETURN		

SUBROUTINE SETUP

- PURPOSE=To inialize the variables significant toward implicit enumeration of the tree.
- COUNT= Variable that counts iterations of branch/bound.
- ST = Variable which indicates how many levels on tree there are to fathom.
- FF = Feasibility indicator
 FF = 0, Feasible FF =
 1, optimal FF = BIGI,
 Infeasible.
- BUD = Variable which indicates how much money has been spent on construction to date.
- JM = Variable which represents the present link being examined for possible construction.
- BBUD = Dummy variable which substitutes for bud variable in subroutine round.
- KL= The initial level at which the branch and bound procedure begins.



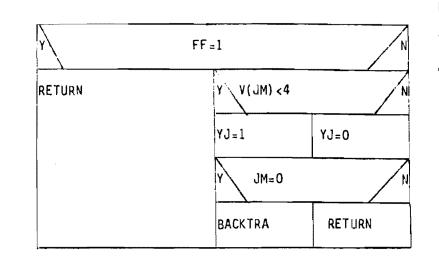


SUBROUTINE SELECT

Purpose: To set variable JM to the link JMAX as determined by subroutine MAX and to set V (JM) (the alternative array) to the next alternative of the additions. If any lane addition alternative is rendered not applicable for the solution, this subroutine will bypass that particular choice and go on to the next alternative.

NCND = Flag variable which indicates if any lane addition alternatives are to eliminate from the solution process. NCND = 1, Yes. NCND=0, No.

IOPT () = Flag variable which indicates if lane addition alternative I is to be eliminated from the solution process. (I = 1 ...4) IOPT(I) = I or 0 If IOPT (I) = I then yes



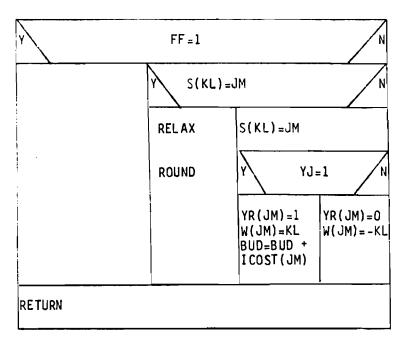
SEPARAT

SUBROUTINE SEPARAT

Purpose: To determine if the branching of the variable JM will be one of the four lane addition alternatives (YJ=1) or no lane addition at all (YJ=0). If JM=0 all possibilities exhausted: back tract.

YJ = Variable whichsignifies if any lane addition alternative is being examined. YJ = 1, Yes. YJ=0, No.

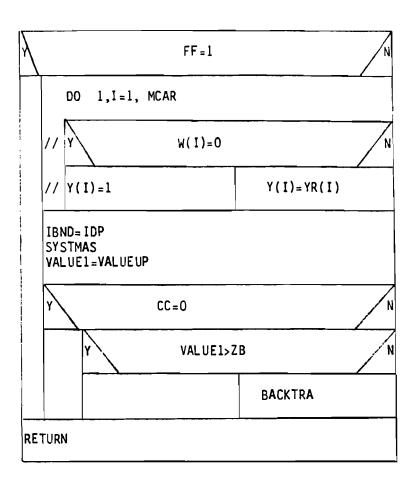




SUBROUTINE SET

Purpose: To set variable JM at level KL of tree. Set YR and W arrays to indicate whenever the variable is set to construct an alternative (YJ=1) or not to construct an alternative (YJ=0). If YJ is set to 1, adjust the value of BUD to include cost of construction for link JM.

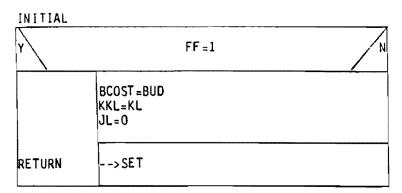




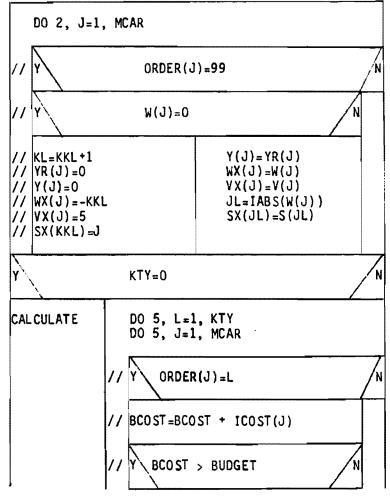
SUBROUTINE RELAX

Purpose: To determine lower bounds on the objective function (total travel time of traffic) by relaxing the budget constraint. All free variables (links) are allowed to construct lane additions regardless of the cost. Furthermore, subroutine relax utilizes the system optimal traffic assignment which when solved yields a solution no worse than a user optimal solution udner the same exact specifications of design.

- IBMD = Temporary variable
 used to replace IDP
 and ITP in
 subroutines
- IDP = Variable which represents the lane addition type used for lower bound calculations
- Value 1 =Objective function value determined by system optimal traffic assignment
- Value UP = objective function value determined by system optimal traffic assignment
- CC = flag for first use of subroutine relax. cc=0 yes cc=1 no



SET



SUBROUTINE ROUND

Purpose: To set a feasible solution, calculate the total travel time (value up) of the feasible solution. and then to compare this valueup to ZB (the best solution so far). If valueup < ZB, then ZB (old)</pre> is replaced by valueup (ZB (new)). Furthermore, if valued < Z1 (the absolute lowest bound to the problem) then stop, valueup is optimal for this problem. Otherwise, if valueup = valuel (the system optimal solution to the same set of 0-1 variables) then backtract, else find the next link to examine. KKL = Dummy variable that temporarily replaces KL in subroutine round.

JL = Dummy variable used to temporarily store absolute value of W(J). WX = Temporary storage array for W)(.) array. VX = Temporary storage array for v(.) array. SX = Temporary storage array for s(.) array. YB = An array that stores the best solution of the YR () array. VB = An array that stores the best solution of the V () array. WB = An array that stores the best solution of the W () array. SB = An array that stores the best solution of the S () array. ITP = Variable which represents the lane addition type used for upper bound

calculations.

<pre>// BCOST=BCOST-ICOST(J) YR(J)=1 // YR(J)=0 Y(J)=1 // Y(J)=0 KKL=KKL+1 // KKL=KKL+1 VX(J)=ITP // VX(J)=5 WX(J)=KKL // WX(J)=-KKL SX(KKL)=J // SX(KKL)=J</pre>

CALCULATE

IBND=ITP USERASG				
r	VALUEUP >	ZB	N	
MAX FF = 0 SELECT ZB=VALUEUP SEPARAT D0 3, I = 1, SET // YB (I) =YR (I FEASIBL // VB (I) =VX (I RELAX // WB (I) =WX (I ROUND // SB (I) =SX (I // BBUD =BCO ST OPT IMAL Y VALUE UP -Z FF = 1 Y		I=1, MCAR YR(I) VX(I) WX(I) SX(I) COST UP-Z1*PER/10	(I) (I) (I) ST -Z1*PER/100 <z1 n<="" td=""></z1>	
	OPT IMAL	BACKTRA	MAX SELECT SEPARAT SET FEASIBL RELAX ROUND	
RETURN				

INITIAL

Sets cost equal to the present value of bud, sets KKL equal to the present value of K., and initializes JL equal to zero.

SET

Checks if link J has been set (to 0 or 1) by testing the order array. If link J has been set (ordered (J)=99) then the propr values of YR, W, V and S for link J are temporarily placed into their corresponding positions of the Y, WX VX and SX respectively. Otherwise, if link J has not been set, the order array determines the temporary destiny of the free links. All free links have order array value between 1 and KTY. Starting with the link whose order array value is 1, we temporarily allow that link to build an additional lane (ITP choice of lane) provided the cost of build that particular link does not put the entire project over budget. (This procedure is continued until the subroutine has looked at the link whose order array value is KTY.) At this point, all links should be at least temporarily set to zero or one

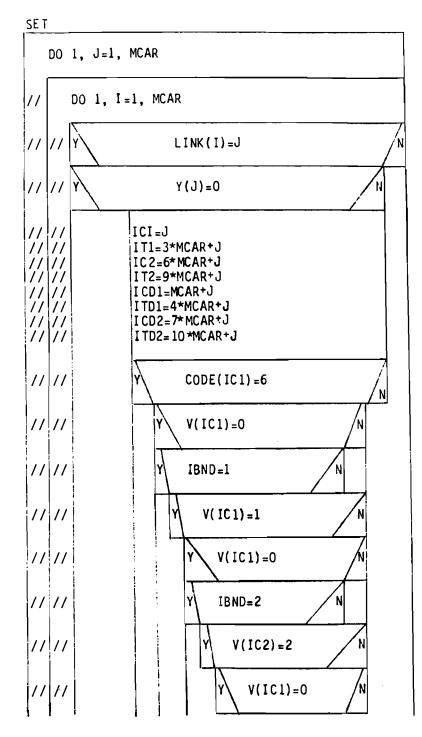
CALCULATE

Calls the user optimal traffic assignment to calculate valueup. If valueup > ZB, then continue the search of optimal solution down that branch of tree. Otherwise, if valueup < ZB, then valueup becomes the best solution so far (ZB = valueup), and therefore the values temporarily stored in the YR, VX, WX, SX and Boost arrays are placed into the YB, VB, WB, SB and BBUD arrays respectively for storage of best solution to date. Furthermore, if valueup is within the given percent (per) of Z1 (the absolute lowest bound), then stop valueup is determined to be the optimal solution to the program. Else, if valueup = value 1 (the system optimal solution at that point in tree) backtract because a better solution cannot be found if the program continues down this branch of the tree. OPT IMAL

COUNT = COUNT + 1				
r	FF = 1		N	
PRINT STOP	'BEST SOLUTION'	RETURN		

SUBROUTINE OPTIMAL

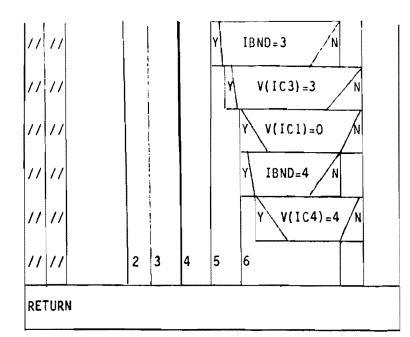
Purpose: To stop the algorithm if an optimal solution has been found or if the entire tree has been implicitly enumerated. FF = 1 signifies one of the two possible alternatives has been reached. CHANGES

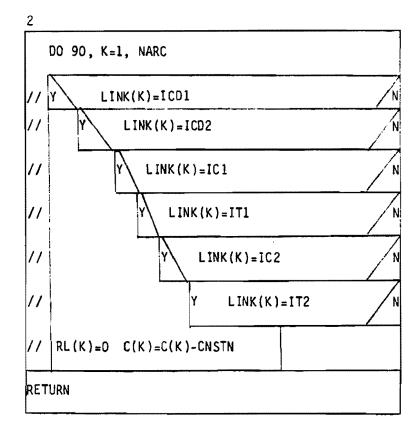


SUBROUTINE CHANGES

Purpose: To transmit the changes (i.e. - the link addition characteristics) made by subroutines round or relax to the corresponding traffic assignment subroutines (systmas or userasg respectively) in order to represent accurately the network design of the given vertex. IC1 = Car link equivalent of link J IT1 = Truck link equivalent of link J. ICZ = Opposite direction car link equivalent of link J. ITZ - Opposite direction truct link equivalent to link J. ICD1 - Dummy car link associated with link J lane addition. ICDZ = Dummy opposite direction car link associated with link J. ITD1 = Dummy truck link associates with link J. ITDZ - Dummy opposite direction truck link associated with link J. SET

Sets the values of IC1, IT1, ICZ, ITZ, ICD1, ICDZ, ICD1, ITDZ associated with link J for purpose of making the proper changes of link J in order to accuraltely represent the new network design.





2

Changes of link J are such: Link J will not have any additional lanes, but trucks will be prohibited from using left-hand lane.

3

Changes of link J are such: Number of lanes of link J are increased by one (in both directions) all traffic is allowed on the new lane.

4

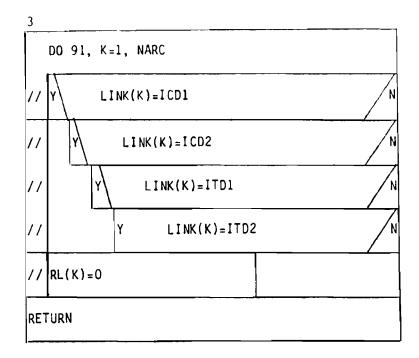
Changes of link J are such: Number of lanes link J are increased by one (in both directions) only trucks are allowed to use the new lane. (Note: trucks are allowed to use <u>all</u> lanes.)

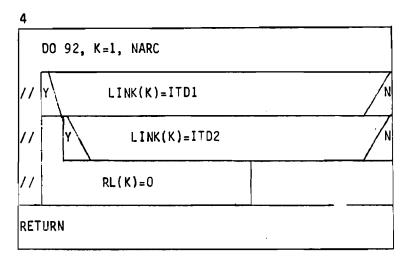
5

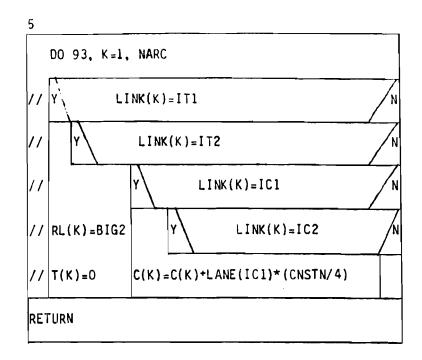
Changes of link J are such: Number of lanes of link J are increased by one (in both directions) all trucks are restricted to use the new lane <u>only</u>. All cars are restricted from using the new lane.

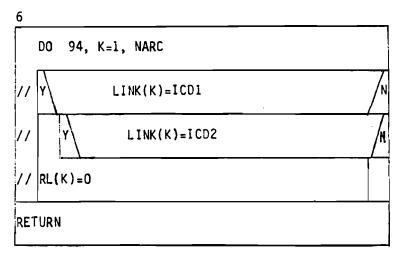
6

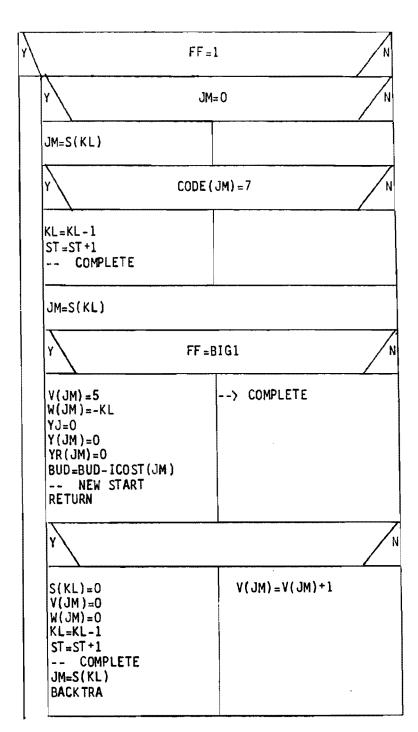
Changes of link J are such: Number of lanes of link J are increased by one (in both directions) only cars are permitted to use the new lane. (Note: Cars are permitted to use all lanes.)











Subroutine BACKTRA

Purpose: To change direction of implicit enumeration process once a vertex has been fathomed. Fathoming occurs when: the system optimal solution(VALUE1) is greater than or equal to the current best feasible solution (ZB); the user-optimal solution (VALUEUP) equals the system optimal solution (VALUE1); the expense required (BUD) to construct all lane additions defined by the solution set (Y) exceeds the budget (BUDGET); or no separation variable (JM=0) exists.

Change

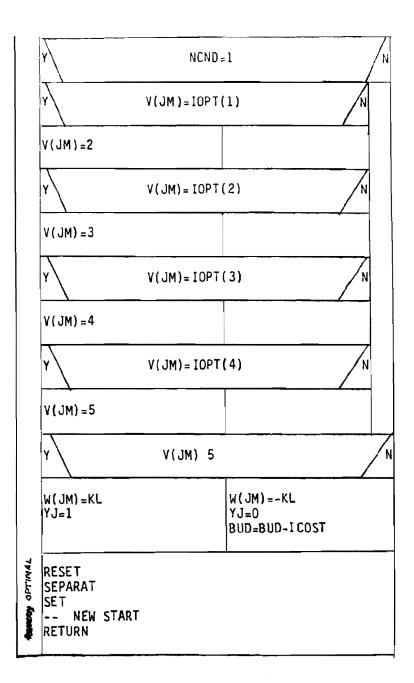
Creates a new vertex by changing the value of the most recently defined variable (JM) that has not exhausted all options [V(i)<5].

Complete

Signifies that enumeration is complete. Optimal solution has been found.

New Start

Subroutine has successfully fathomed a vertex. A new solution is defined.

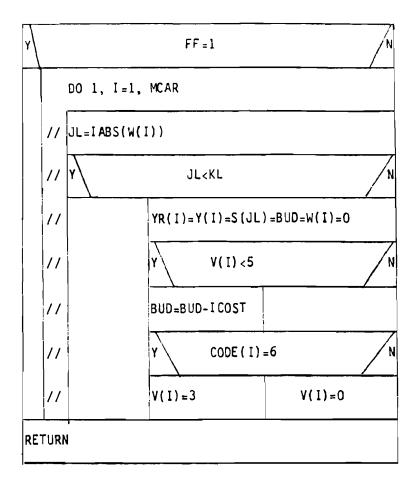


Y	KL=ILEVEL	N
FF=1 OPTIMAL STOP	RETURN	

NEW START

	ļ
FEASIBLE	ł
RELAX	
ROUND	-
RETURN	1
RETURN	Į



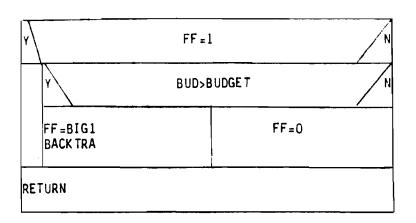


SUBROUTINE RESET

Purpose: To free variables set at tree levels higher than KL. This is caused by a backtrack operation to level KL.

For each variable, test if it was set at a tree level higher than KL. If not, go to next variable. If so, then set YR, X,S,W to O and set V to appropriate value (O or 3) for the level at which the variable was set. Adjust the value of BUD if V(I) < 5.

FEASIBL



SUBROUTINE FEASIBLE

Purpose: To determine if the sum of the cost of construction of lane additions recommended (bud) is less than total budget allowed. If Bud < budget continue to next subroutine. Otherwise, backtract.