ANALYSIS OF TRUCK USE AND
HIGHWAY COST ALLOCATION
IN TEXAS

in cooperation with the
Department of Transportation
Federal Highway Administration

RESEARCH REPORT 332-1
STUDY 2-18-83-332
TRUCK USE AND HIGHWAY COST ALLOCATION
The highway cost allocation problem is one of determining equitable charges for each of the vehicle classes sharing transportation facilities such as highways and bridges. Previous attempts at solving this problem can essentially be reduced to two major approaches: (a) proportional allocation methods, which determine costs in proportion to one or more measures of highway usage; and (b) incremental methods, which allocate costs on the basis of highway design differences necessary to accommodate gradually heavier vehicle classes.

This report develops two new highway cost allocation methodologies that actually extend the basic concepts of the incremental and proportional allocation procedures. The new methods are referred to as the "Modified Incremental Approach" and the "Generalized Method." Both methods fulfill the following conditions: (a) highway costs are completely financed by users (completeness condition); (b) vehicle classes reduce their cost responsibilities by sharing the facilities with other vehicle classes (rationality principle); and (c) vehicle classes are charged at least enough to cover their corresponding marginal costs (marginality principle). An example using Texas pavement data is utilized to illustrate the application of the proposed methods.
ABSTRACT

The highway cost allocation problem is one of determining equitable charges for each of the vehicle classes sharing transportation facilities such as highways and bridges. Previous attempts at solving this problem can essentially be reduced to two major approaches: (a) proportional allocation methods, which determine costs in proportion to one or more measures of highway usage; and (b) incremental methods, which allocate costs on the basis of highway design differences necessary to accommodate gradually heavier vehicle classes.

This report develops two new highway cost allocation methodologies that actually extend the basic concepts of the incremental and proportional allocation procedures. The new methods are referred to as the "Modified Incremental Approach" and the "Generalized Method". Both methods fulfill the following conditions: (a) highway costs are completely financed by users (completeness condition); (b) vehicle classes reduce their cost responsibilities by sharing the facilities with other vehicle classes (rationality principle); and (c) vehicle classes are charged at least enough to cover their corresponding marginal costs (marginality principle). An example using Texas pavement data is utilized to illustrate the application of the proposed methods.
IMPLEMENTATION STATEMENT

The new cost allocation methodologies developed in this project have been computerized and tested using limited rehabilitation data from the Texas pavement data base. Proposed changes to the RENU2 program have been implemented and are currently being validated.
DISCLAIMER

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented within. The contents do not necessarily reflect the official views or policies of the Texas State Department of Highways and Public Transportation or the Federal Highway Administration. This report does not constitute a standard, a specification, or a regulation.
## TABLE OF CONTENTS

Abstract ............................................................................................................ i
Implementation Statement ............................................................................. ii
Disclaimer ......................................................................................................... iii
Table of Contents ............................................................................................ iv
List of Tables ..................................................................................................... vi
List of Figures ................................................................................................... vii

Chapter 1 Introduction .................................................................................... 1

Chapter 2 Literature Survey ........................................................................... 4

Chapter 3 Pavement Rehabilitation: The RENU2 Program ....................... 8
   3.1 Pavement Performance Function ......................................................... 8
   3.2 Pavement Survival Function ............................................................... 15
   3.3 Age Adjustment Function ................................................................... 17
   3.4 Load Shifting Function ....................................................................... 23
   3.5 Cost Estimating Function .................................................................... 23

Chapter 4 Methodology ................................................................................... 27
   4.1 The Modified Incremental Approach ................................................ 27
   4.2 The Generalized Method ..................................................................... 32
   4.3 Environmental Factors ....................................................................... 36

Chapter 5 Application of the Methodology .................................................. 41

Chapter 6 Summary ........................................................................................ 46

References .......................................................................................................... 47

Appendix 1 Modified RENU2 Fortran Code .................................................. 49
Appendix 2 Basic Input Data for the Modified RENU2 Program ............... 120
Appendix 3 Cost Output from the Modified RENU2 Program .................. 126
Appendix 4 Modified Incremental Approach BASIC Code ....................... 157
Appendix 5 Sample Run for the Modified Incremental Approach ........... 164
Appendix 6 Generalized Method Basic Code .............................................. 166
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Serviceability Performance Curve Parameters by Pavement Type</td>
</tr>
<tr>
<td>3.2</td>
<td>Primary Distress Type and Curve Fit Parameters by Pavement Type</td>
</tr>
<tr>
<td>3.3</td>
<td>Design Parameters for PSI Survivor Curves</td>
</tr>
<tr>
<td>3.4</td>
<td>Design Parameters for Distress Survivor Curves</td>
</tr>
<tr>
<td>4.1</td>
<td>Vehicle Combinations</td>
</tr>
<tr>
<td>5.1</td>
<td>Vehicle Class Data</td>
</tr>
<tr>
<td>5.2</td>
<td>Illustrative Pavement System</td>
</tr>
<tr>
<td>5.3</td>
<td>Rehabilitation Cost Estimates</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Basic Approach of the Incremental Method</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Performance Curves</td>
<td>10</td>
</tr>
<tr>
<td>3.2</td>
<td>Typical Survival Curve</td>
<td>17</td>
</tr>
<tr>
<td>3.3</td>
<td>Age Distribution Adjustment</td>
<td>22</td>
</tr>
<tr>
<td>3.4</td>
<td>Rehabilitation Cost as a Function of ESALs</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>Input Cost Estimates</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>Cost Allocation Using the Modified Incremental Approach</td>
<td>30</td>
</tr>
<tr>
<td>4.3</td>
<td>The Core in the Generalized Method</td>
<td>35</td>
</tr>
<tr>
<td>4.4</td>
<td>Effects of the Vehicle Classes on Cost as Functions of e</td>
<td>40</td>
</tr>
</tbody>
</table>
1. **INTRODUCTION**

An important problem currently receiving a great deal of attention from state legislatures is the highway financing problem. In recent years, this problem has become more acute due to the fact that a significant portion of U.S. highway pavements is reaching unacceptable levels of user serviceability.

Two basic questions must be answered by a highway cost allocation procedure: (a) how much is needed to keep a highway (or other transportation facilities) operational during a specified planning horizon? And (b) what fraction of the total cost must be charged to each vehicle class in the traffic stream served by the facility? This paper summarizes the results of recent work aimed at providing adequate answers to both of these questions. In particular, two new cost allocation methods are developed: a modified incremental approach and an optimization method; the optimization method will be referred to as a generalized procedure, since in fact it is based on an extension of the concepts used in the incremental [4,10,11] and proportional allocation [6,7,10] methods.

Proportional allocation methods determine cost responsibilities on the basis of a measure that reflects the amount of use of a highway facility by each of the various vehicle classes. Common measures include gross vehicle weight, vehicle-miles of travel, and equivalent single axle loads (ESALs). It must be noted that these methods may yield results that conflict with the perception of fairness by individual vehicle classes; this indeed hinders the acceptability of the results by all the users of the facility and questions the overall applicability of the proportional methods.

Incremental allocation methods identify cost responsibilities on the basis of the cost differences associated with the sequential introduction of
vehicle classes into the traffic stream. Inconsistent results are obtained when vehicle classes are introduced in different sequences, however. This inconsistency constitutes a serious flaw in any cost allocation method that seeks to be equitable.

The two procedures developed in this paper exhibit properties which make them superior to those previously used in the context of highway facility planning. In particular, they fulfill three fundamental requirements:

(a) **Completeness**: the provision of highway facilities must be entirely financed by the various vehicle classes that utilize them.

(b) **Rationality**: the common facility is the most economically attractive alternative for all vehicle classes to meet their transportation needs; that is, any other alternative to satisfy this need, such as using an exclusive facility, would be more expensive for any vehicle class.

(c) **Marginality**: the allocated costs associated with any vehicle class must be sufficient to at least cover its corresponding marginal costs.

The completeness requirement insures that only funds provided by highway users are considered to finance the common highway facility. The rationality requirement is a well established concept in the economics literature [16] which deals with a fundamental characteristic of economic behavior. Any procedure which violates this condition would be strongly objected. The marginality requirement is another widely accepted economical principle [19]. The violation of this principle implies the existence of cross-subsidization among the vehicle classes involved. The rationality and marginality requirements establish an essential element of equity in the cost allocation procedure.
In conclusion, having an equitable cost allocation methodology (which satisfies the rationality and marginality principles) to analyze the many aspects related to the highway financing problem enhances the acceptability of the results among the various vehicle classes which must cover the total cost of the facility. In Chapter 2 this important issue is briefly discussed for both the proportional and the incremental methodologies.

This report presents a summary of the relevant previous work related to highway cost allocation procedures in Chapter 2. A brief description of the RENU2 program is given in Chapter 3. Chapter 4 delineates the basic methodology developed for this study. Such methodology includes two cost allocation procedures—the Modified Incremental Method and the Generalized Method—and a procedure to study the effect of environmental factors on highway costs. Applications of the proposed methodology using Texas pavement data are given in Chapter 5.
2. LITERATURE SURVEY

Currently available solution procedures for the highway cost allocation problem are not economically justifiable. Indeed, perhaps a non-controversial solution methodology to that problem does not exist; nevertheless, cost must be allocated in some rational way. Traditionally, it has been an accepted practice to define cost responsibilities on the basis of some criterion of efficiency which represents the use of the facility by the various vehicle classes.

One of the most widely used methods in highway cost allocation is the so-called incremental approach, which was adopted in the earlier cost allocation studies conducted in the United States. This approach was adequate while new construction was the principal cause of highway cost. However, now that a larger portion of the budget must be assigned to the maintenance and rehabilitation of existing facilities, the incremental approach has been reviewed and questioned, and some important problems, which will be discussed later, have been found.

The incremental method has been used in a number of cost allocation studies such as the first Federal Highway Cost Allocation Study [4], and studies conducted in several states including Virginia, Washington, North Dakota, Montana, Kentucky, and Rhode Island [3, 10, 11, 12, 13, 15].

According to the incremental method, the cost of a highway facility designed for the lightest vehicle class is initially calculated; then vehicle classes are sequentially included in increasing order of axle weight and corresponding highway design or rehabilitation costs are calculated for the resulting traffic streams and a specified design period. At each step of the procedure, the cost difference between one design and the next is allocated to the vehicle class incorporated in that step. Some minor varia-
tions of the basic incremental method have also been considered [6].

Although it meets the completeness, rationality, and marginality requirements aforementioned, there is one important difficulty with the incremental method: it is not consistent. The method produces different results if vehicle classes are introduced in different orderings. This is due to the presence of overlapping requirements among the various vehicle classes. Figure 2.1 illustrates this inconsistency. The shaded areas in Figures 2.1(a), 2.1(b), and 2.1(c) represent costs allocated to vehicle classes 1, 2, and 3 if they are sequentially introduced in that order. However, the shaded areas in Figures 2.1(d), 2.1(e), and 2.1(f) represent the same costs when vehicle class 3 is included first, followed by vehicle classes 1 and 2.

Another accepted approach to the problem under consideration is to allocate costs in proportion to a single numerical criterion which, in the context of transportation systems, represents a measure of use or damage caused by the vehicle classes using a common highway facility. This method is known as the proportional method or the consumption approach [6,7]. The appeal of this method lies on its simplicity and on the fact that, if the appropriate basis is selected, the fairness of its results is less open to dispute.

A major issue with the proportional allocation method, however is that it may yield cost allocations which conflict with the interests of the individual vehicle classes. This difficulty is due to the fact that the method ignores the strategic alternatives (coalitions) available to the vehicle classes in order to meet their transportation needs. Such strategic alternatives include sharing a common facility with all vehicle classes, sharing a facility with some of the other vehicle classes, and having an exclusive facility. In other words, under the proportional allocation me-
Figure 2.1  Basic Approach of the Incremental Method
method, it is possible for a particular vehicle class to pay more by sharing a common facility than it would have to pay by having its own exclusive one.

In a pioneering and enlightening article, Young, Okada, and Hashimoto [18] analyze several cost allocation methods used in water resources management. Among the methods discussed, those that stem from the theory of cooperative games [16,17] are of particular interest. These methods provide means for approaching the cost allocation problem by taking into account all the possible strategic alternatives available to each vehicle class in the provision of highway facilities needed to meet a specified traffic demand. These various strategic possibilities actually establish constraints which define a set of feasible solutions that satisfy the completeness, rationality and marginality requirements. The cost allocations resulting from these methods are more likely to be accepted because they are formulated on the basis of fundamental economic principles.
3. **PAVEMENT REHABILITATION: THE RENU2 PROGRAM**

The RENU2 program [8] estimates the maintenance and rehabilitation costs associated with changes in legal load limits. In this program, the number of equivalent loads (ESAL) is calculated for both current and proposed legal limits from given traffic compositions and axle distributions. Based on the resulting ESALs, life cycles of typical pavements are analyzed and rehabilitation and maintenance costs are estimated. The RENU2 program essentially performs five functions:

(a) Pavement performance function.
(b) Pavement survival function.
(c) Pavement age adjustment function.
(d) Load shifting function.
(e) Cost estimating function.

The rest of this chapter will be dedicated to the discussion of each of these points.

3.1 **Pavement Performance Function**

The pavement performance function predicts the deterioration trend of a pavement in terms of the loss of PSI (present serviceability index) or the increase in area or severity of a distress (cracking, rutting, flushing, etc.) as the level of traffic loads increases. In this function, the life cycle of a pavement is identified for given traffic conditions. It is assumed that a terminal performance index (either a minimal PSI value or maximal distress area/severity values) is specified and that the life cycle of a pavement is completed when this critical value is reached.

A functional form that has been found to adequately represent the loss of PSI for Texas highways is:
\[ g_t = e^{-\left(\frac{\rho}{W}\right)^2} \]  

(3.1)

where

\[ W: \text{ No. of cumulative ESALs}, \]
\[ \rho: \text{ Scale parameter}, \]  
\[ \beta: \text{ Form parameter} \]

The damage function \( g(W) \) can also be expressed as the ratio of the loss in serviceability after \( W \) 18-kip ESALs to a specified maximum design loss.

Let \( P_0 \) be the initial PSI (at \( W = 0 \)), \( P_t \) be the PSI after \( W_t \) 18-kip ESALs, and \( P_f \) be a lower bound on the PSI. Then the relative loss after \( W_t \) ESALs can be expressed as:

\[ g_t = \frac{(P_0 - P_t)}{(P_0 - P_f)} \]  

(3.2)

From Eq. (3.2) it is possible to express \( P_t \) as a function of \( g_t \), as follows:

\[ P_t = P_0 - (P_0 - P_f)g_t \]  

(3.3)

Eq. (3.3) can be further rewritten after using Eq. (1). The final result is given by:

\[ P_t = P_0 - (P_0 - P_f)e^{-\left(\frac{\rho}{W}\right)^2} \]  

(3.4)

Figure 3.1(a) shows the form of the loss of PSI \( P_t \) as a function of the cumulative number of ESALs \( W \) according to Eq. (3.4). Note that \( P_f \)
Figure 3.1 Performance Curves
represents an asymptotic minimum PSI value. Figures 3.1(b) and 3.1(c) show, respectively, the influence of parameters \( \rho \) and \( \beta \) on the form of the function.

The Texas Transportation Institute has estimated, through statistical procedures, values for \( P_f \), \( \rho \) and \( \beta \) from measured pavement data [9]. Table 3.1 indicates mean, maximum, and minimum values of these parameters for hot mix, black base, and overlaid pavements.

Very frequently pavements may be seriously distressed and in need of major rehabilitation before the serviceability index drops to its terminal value. This is particularly true of pavements with severe alligator and transverse cracks. In cases where the asymptotic serviceability index, \( P_f \), is higher than the terminal serviceability index, \( P_t \), or when the remaining life calculated from the serviceability index equation is excessively long (say 30 to 40 years), the pavement will probably need major rehabilitation due to distress.

Pavement distress can appropriately be represented by estimating two separate components: density and severity. Density may be expressed either as the percent of the total pavement surface area that is covered by the distress, or total crack length per unit area, or crack spacing, or similar measures. Severity may be expressed as either an objective or subjective measure. Examples of objective measures are crack width, crack depth, and relative displacement at joint. Subjective measures may be assessed reliably by comparing the observed distress with photographs of different levels of severity which may be described as none, slight, moderate, or severe and may be given numerical ratings such as 0, 1, 2, and 3, respectively, or be assigned numbers that are proportional to these in a range between 0 and 1. The change of either area or severity of distress can be evaluated using the previously discussed equations.
Table 3.1 Serviceability Performance Curve Parameters by Pavement Type

<table>
<thead>
<tr>
<th>Pavement Type</th>
<th>Black Base</th>
<th>Hot Mix Asphalt concrete</th>
<th>Overlays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Test Sections</td>
<td>51</td>
<td>36</td>
<td>77</td>
</tr>
<tr>
<td>( \rho ) (mean)</td>
<td>2.321</td>
<td>1.960</td>
<td>1.974</td>
</tr>
<tr>
<td>( \rho ) (min)</td>
<td>0.005</td>
<td>0.100</td>
<td>0.013</td>
</tr>
<tr>
<td>( \rho ) (max)</td>
<td>17.239</td>
<td>11.098</td>
<td>9.188</td>
</tr>
<tr>
<td>( \beta ) (mean)</td>
<td>1.337</td>
<td>1.952</td>
<td>1.196</td>
</tr>
<tr>
<td>( \beta ) (min)</td>
<td>0.300</td>
<td>0.095</td>
<td>0.095</td>
</tr>
<tr>
<td>( \beta ) (max)</td>
<td>6.277</td>
<td>7.259</td>
<td>2.893</td>
</tr>
<tr>
<td>( P_o ) (mean)</td>
<td>4.15</td>
<td>3.87</td>
<td>3.92</td>
</tr>
<tr>
<td>( P_o ) (min)</td>
<td>2.79</td>
<td>2.86</td>
<td>2.07</td>
</tr>
<tr>
<td>( P_o ) (max)</td>
<td>4.77</td>
<td>4.78</td>
<td>4.88</td>
</tr>
<tr>
<td>( P_f ) (mean)</td>
<td>1.962</td>
<td>1.661</td>
<td>2.121</td>
</tr>
<tr>
<td>( P_f ) (min)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.004</td>
</tr>
<tr>
<td>( P_f ) (max)</td>
<td>4.295</td>
<td>4.305</td>
<td>4.391</td>
</tr>
</tbody>
</table>
In order to study the behavior of the area covered by a given type of distress, and the corresponding level of severity, two indices will be introduced: (a) the distress area index, and (b) the distress severity index. Each of these indices represents a number between 1 and 0 which decreases as the level of traffic is increased. Note that the present serviceability index (PSI) has a similar behavior, with the exception that it decreases from $P_0$ to $P_f$.

Specifically, the distress area index decreases from a value $A_0$ ($A_0 < 1$) to a value $A_f$ ($0 < A_f < A_0$) as the traffic increases; similarly, the distress severity index decreases from a value of $S_0$ ($S_0 < 1$) to a value $S_f$ ($0 < S_f < S_0$) as the traffic level increases; that is, a recently rehabilitated pavement will have indices close to one, as opposed to pavements in need of rehabilitation which will have indices close to zero.

The distress area index, $A$, is expressed by a relationship similar to that of Eq. (3.4), namely,

$$A = A_0 - (A_0 - A_f) e^{-\left(\rho / W\right)^\beta}$$

(3.5)

Similarly the distress severity index, $S$, is expressed as

$$S = S_0 - (S_0 - S_f) e^{-\left(\rho / W\right)^\beta}$$

(3.6)

Using the $A$, $S$, and $W$ data from the Texas Transportation Institute data base, the parameters and have been estimated for the most significant distress types affecting black base, hot mix, and overlaid pavements which are, respectively, alligator cracking area, alligator cracking severity, and transverse cracking severity [1]. Table 3.2 summarizes the results obtained [9].
Table 3.2 Primary Distress Type and Curve Fit Parameters by Pavement Type

<table>
<thead>
<tr>
<th>Pavement Type</th>
<th>Black Base</th>
<th>Hot Mix Asphalt concrete</th>
<th>Overlays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Distress</td>
<td>Alligator Cracking Severity</td>
<td>Alligator Cracking Area</td>
<td>Transverse Cracking Severity(*)</td>
</tr>
<tr>
<td>$p$ (mean)</td>
<td>1.19</td>
<td>0.93</td>
<td>85.57</td>
</tr>
<tr>
<td>$p$ (min)</td>
<td>0.14</td>
<td>0.07</td>
<td>24.13</td>
</tr>
<tr>
<td>$p$ (max)</td>
<td>3.01</td>
<td>3.63</td>
<td>194.83</td>
</tr>
<tr>
<td>$\beta$ (mean)</td>
<td>2.54</td>
<td>3.43</td>
<td>1.47</td>
</tr>
<tr>
<td>$\beta$ (min)</td>
<td>0.89</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>$\beta$ (max)</td>
<td>8.78</td>
<td>18.21</td>
<td>5.52</td>
</tr>
</tbody>
</table>

(*) The $p$ and $\beta$ terms for this case are determined in terms of the number of months the pavement has been in service.
3.2 **Pavement Survival Function**

The pavement survival function estimates the percent of miles of pavement that do not need to be rehabilitated when the pavement performance function indicates that the specified critical performance index is reached. From this information it is possible to subsequently estimate the predicted number of miles that will need to be rehabilitated after a given number of load applications.

A survivor curve is a functional relationship that predicts the percentage of mileage in a given pavement category that does not require immediate rehabilitation at a specified time. This specified time can be considered as the time at which the pavement has reached a given traffic load level, or the time since last rehabilitation. Evidently, to decide if a pavement requires or does not require some kind of rehabilitation, it is first necessary to define a measure of pavement performance. This measure of performance has been defined in terms of PSI or distress as shown in the previous sections. The fundamental idea behind the development of a survivor curve is the concept that since the performance relationship is deterministic, it would be meaningful to determine a second relationship that estimates the percent of pavement mileage that actually survives when the performance function reaches a critical value.

Survival times are data that measure the time to failure. These times are subject to random variations, and like any random variables, form a distribution; the two-parameter Weibull distribution [14] is assumed as the survival distribution for predicting the survival or failure rate of pavements. The Weibull distribution is one of the well-known survival distributions; its applicability to various failure situations, such as electron tube failure, the fatigue life of deep-groove ball bearings, etc., has been
The Weibull distribution is characterized by two non-negative parameters $\lambda$ and $\gamma$; its probability density function, $f(w)$, and the cumulative distribution function, $F(w)$, are defined as follows:

$$f(w) = \lambda \gamma (\lambda w)^{\gamma-1} e^{- (\lambda w)\gamma}$$  \hspace{1cm} (3.7)$$

$$F(w) = 1 - e^{- (\lambda w)\gamma}$$  \hspace{1cm} (3.8)$$

In the specific application of the Weibull distribution to the study of pavement survivability, $w$ represents the traffic load at which the pavement reaches a critical performance level. The parameters $\lambda$ and $\gamma$ are referred to as a "scale parameter" and a "shape parameter", respectively.

The survival function, denoted by $s(w)$, is defined as the probability that an individual mile of pavement of a given type survives a traffic load larger than $w$. From the definition of the cumulative distribution function $F(w)$, it can be concluded that $s(w) = 1 - F(w)$. That is,

$$s(w) = e^{- (\lambda w)\gamma}$$  \hspace{1cm} (3.9)$$

Figure 3.2 illustrates the typical shape of the survival function. As explained here, $s(w)$ is the survival rate of a given type of pavement structure under $w$ traffic loads.
Survival curves have been obtained for Texas flexible pavements using different critical levels for PSI and the most relevant types of distress [9]. Table 3.3 shows $\lambda$ and $\gamma$ values for PSI survival curves corresponding to hot mix, black base and overlaid pavements. Table 3.4 gives $\lambda$ and $\gamma$ values associated with distress survival curves for the same types of pavements.

### 3.3 Pavement Age Adjustment Function

The pavement age adjustment function updates the age distribution of the pavement mileage when it is rehabilitated. Typical rehabilitation actions include regular thin overlays when the pavement fails because of several distress types or medium to thick overlays when it fails as a result of PSI loss.

Figure 3.3 summarizes the age adjustment procedure for a representative pavement section and a given year of a specified analysis period. The procedure can be described as follows:
Table 3.3 Design Parameters for PSI survivor curves

<table>
<thead>
<tr>
<th></th>
<th>Black Base</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_c )</td>
<td>( \lambda )</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>1.0</td>
<td>0.276</td>
<td>2.111</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>0.417</td>
<td>1.549</td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>0.607</td>
<td>1.497</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Hot Mix</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_c )</td>
<td>( \lambda )</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>1.0</td>
<td>0.423</td>
<td>1.363</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>0.687</td>
<td>1.365</td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>0.787</td>
<td>1.012</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Overlays</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_c )</td>
<td>( \lambda )</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>1.0</td>
<td>0.327</td>
<td>1.524</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>0.555</td>
<td>1.163</td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>0.818</td>
<td>1.088</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.4 Design Parameters for Distress Survivor Curves

<table>
<thead>
<tr>
<th></th>
<th>gc = 0.25</th>
<th></th>
<th>gc = 0.50</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>λ</td>
<td>γ</td>
<td>λ</td>
<td>γ</td>
</tr>
<tr>
<td><strong>Rutting</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.010</td>
<td>2.801</td>
<td>0.006</td>
<td>2.133</td>
</tr>
<tr>
<td>Severity</td>
<td>0.010</td>
<td>3.143</td>
<td>0.006</td>
<td>2.782</td>
</tr>
<tr>
<td><strong>Alligator Cracking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.006</td>
<td>3.065</td>
<td>0.003</td>
<td>2.129</td>
</tr>
<tr>
<td>Severity</td>
<td>0.007</td>
<td>4.380</td>
<td>0.004</td>
<td>2.681</td>
</tr>
<tr>
<td><strong>Longitudinal Cracking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.006</td>
<td>2.815</td>
<td>0.003</td>
<td>2.068</td>
</tr>
<tr>
<td>Severity</td>
<td>0.008</td>
<td>3.285</td>
<td>0.005</td>
<td>2.279</td>
</tr>
<tr>
<td><strong>Transverse Cracking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.006</td>
<td>2.760</td>
<td>0.003</td>
<td>1.878</td>
</tr>
<tr>
<td>Severity</td>
<td>0.008</td>
<td>3.382</td>
<td>0.004</td>
<td>2.443</td>
</tr>
</tbody>
</table>
### Table 3.4 Design Parameters for Distress Survivor Curves (Cont'd)

<table>
<thead>
<tr>
<th></th>
<th>Hot Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$g_c = 0.25$</td>
</tr>
<tr>
<td></td>
<td>$\lambda$</td>
</tr>
<tr>
<td><strong>Rutting</strong></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.007</td>
</tr>
<tr>
<td>Severity</td>
<td>0.007</td>
</tr>
<tr>
<td><strong>Alligator</strong></td>
<td></td>
</tr>
<tr>
<td>Cracking</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.006</td>
</tr>
<tr>
<td>Severity</td>
<td>0.007</td>
</tr>
<tr>
<td><strong>Longitudinal</strong></td>
<td></td>
</tr>
<tr>
<td>Cracking</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.006</td>
</tr>
<tr>
<td>Severity</td>
<td>0.007</td>
</tr>
<tr>
<td><strong>Transverse</strong></td>
<td></td>
</tr>
<tr>
<td>Cracking</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.006</td>
</tr>
<tr>
<td>Severity</td>
<td>0.008</td>
</tr>
</tbody>
</table>
Table 3.4. Design Parameters for Distress Survivor Curves (Cont'd)

<table>
<thead>
<tr>
<th></th>
<th>Overlays</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>g&lt;sub&gt;c&lt;/sub&gt; = 0.25</td>
<td>g&lt;sub&gt;c&lt;/sub&gt; = 0.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>λ</td>
<td>γ</td>
<td>λ</td>
<td>γ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rutting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.009</td>
<td>1.604</td>
<td>0.004</td>
<td>1.219</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td>0.010</td>
<td>1.819</td>
<td>0.005</td>
<td>1.804</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alligator Cracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.007</td>
<td>2.575</td>
<td>0.003</td>
<td>2.080</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td>0.009</td>
<td>2.280</td>
<td>0.005</td>
<td>2.056</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Longitudinal Cracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.008</td>
<td>1.519</td>
<td>0.004</td>
<td>1.197</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td>0.011</td>
<td>1.919</td>
<td>0.006</td>
<td>1.693</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transverse Cracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>0.008</td>
<td>1.792</td>
<td>0.004</td>
<td>1.397</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td>0.011</td>
<td>1.916</td>
<td>0.006</td>
<td>1.797</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.3 Age Distribution Adjustment
(a) Calculate the number of accumulated ESALs for each pavement age category as illustrated in Figure 3.3(a) from traffic data and forecasts.

(b) Obtain the expected fraction $P$ of pavements to fail in the current year for each age category by using the number of accumulated ESALs calculated in the previous step and the appropriate survival curve as shown in Figure 3.3(b). The number of lane-miles that fail and hence are due for rehabilitation in the current year are indicated by the shaded portions of the rectangles representing the number of lane-miles in each pavement age category in Figure 3.3(c). The remaining portions of these rectangles represent the surviving pavement.

(c) The new age distribution is obtained by creating a new age category composed of the total number of lane-miles just rehabilitated and updating the ages of the lane-miles that survived, as indicated in Figure 3.3(d).

3.4 **Load Shifting Function**

The load shifting function is a procedure that modifies a given axle load distribution to reflect a change in the current legal load limits. This module can be used to establish the most likely truck traffic distribution that will occur on a highway system after changing the legal axle load limits. This function will not be used in this study.

3.5 **Cost Estimating Function**

The cost estimating function computes maintenance and rehabilitation costs for both the present and proposed legal load limits in order to determine the impact of the change in such limits. The program computes maintenance costs for both routine and preventive maintenance. All costs
can be calculated for each year of a specified planning horizon and can be broken down according to several types of highway systems (i.e., Interstate, US, State, FM).

Rehabilitation activities considered in RENU2 consist of overlays with asphaltic concrete. When the pavement fails by reaching a critical value of PSI, the thickness of the overlay is calculated so that the pavement remains serviceable throughout the rest of the analysis period. If the pavement fails due to a critical value of distress index (area or serviceability), thin overlays are applied periodically. The time period between overlays is specified by the user in this case. The rehabilitation cost is a function of:

(a) The geometry of the road (lane width, shoulder width, etc.).
(b) The critical performance levels set for PSI and distress, which determine the timing of pavement rehabilitations.
(c) The overlay thicknesses.

Routine maintenance costs are estimated using the EAROMAR procedure [5] and cost information provided by the user. These equations were actually developed to predict maintenance work loads for multilane freeways in terms of: (a) patching, (b) crack sealing, and (c) base and surface repairs. The general form of the EAROMAR model can be formulated as follows:

\[
C_t = \frac{(1100C_1+1000C_2+5C_3)}{(1+e^{-(t-10)/1.16})} \quad (3.10)
\]

where

\( C_t \) = annual maintenance cost in year t per lane-mile,
\( C_1 \) = $/sq yd of bituminous skin patching,
\( C_2 \) = $/linear foot of crack sealing,
\[ C_3 = \$/cu\,yd\,of\,bituminous\,base\,and\,surface\,repair. \]

Cost estimates by the Texas Highway Department for \( C_1, C_2, \) and \( C_3 \) are $3.47, $0.25 and $450, respectively.

Preventive maintenance is implemented in terms of seal coats. These seal coats waterproof and improve the texture of the pavement. The user of the RENU2 program specifies the time between seal coats and the corresponding cost per lane mile. The preventive maintenance option is applicable to all flexible pavements and different values can be considered for each representative pavement section.

Cost estimates obtained from RENU2 are used to construct a cost function similar to that portrayed in Figure 3.4. This function represents costs incurred during a specified analysis period, and is dependent upon the number of cumulative ESALs acting on the pavement. The cost estimate for each vehicle class combination is obtained from Figure 3.4 by using the number of ESALs associated with that combination.
Figure 3.4 Rehabilitation Cost as a Function of ESALs
4. METHODOLOGY

4.1 The Modified Incremental Approach

A modified version of the incremental approach is proposed as a suitable methodology to allocate construction, reconstruction or rehabilitation costs. The proposed modification to the incremental approach attempts to overcome the lack of consistency mentioned in Chapter 2; however, an indirect result of this modification is that the computational complexity of the new procedure is increased.

In the Modified Incremental Approach cost estimates are prepared for every vehicle class, as well as for every combination of two or more vehicle classes. As an illustration, if a highway is designed to accommodate three types of vehicle classes 1, 2, and 3, the final cost allocation for each class is determined only after considering hypothetical designs for the following vehicle class combinations and computing the corresponding design costs: (a) class 1, (b) class 2, (c) class 3, (d) classes 1 and 2, (e) classes 1 and 3, (f) classes 2 and 3, and (g) classes 1, 2, and 3.

Using the cost estimates obtained for the above class combinations and a few fundamental operations, the total cost (corresponding to the combination including classes 1, 2, and 3) is partitioned into as many cost components as vehicle combinations; moreover, each cost component can be considered as the estimate of the cost effect of a vehicle class combination. In order to simplify the description of the method, the following notation is used:

\[ C_1 = \text{cost of a highway designed for vehicle class 1 alone}, \]
\[ C_2 = \text{cost of a highway designed for vehicle class 2 alone}, \]
\[ C_3 = \text{cost of a highway designed for vehicle class 3 alone}, \]
\( C_{12} \) = cost of a highway designed for vehicle classes 1 and 2,
\( C_{13} \) = cost of a highway designed for vehicle classes 1 and 3,
\( C_{23} \) = cost of a highway designed for vehicle classes 2 and 3,
\( C_{123} \) = Total cost of a highway designed (for vehicle classes 1, 2, and 3);

The shaded areas in Figure 4.1 illustrate the notation described above. In this Figure, each individual vehicle class is represented by a circle. When two or more vehicle classes are simultaneously considered, the corresponding circles exhibit a certain degree of overlapping. This overlapping represents the portion of the total cost that is due to a combined effect of two or more vehicle classes.

As can be illustrated in Figure 4.2(a), the portion of the total cost that can be attributed to only individual classes 1, 2, and 3 is given by Equations (4.1), (4.2), and (4.3), respectively:

\[
\begin{align*}
P_1 &= C_{123} - C_{23} \\
P_2 &= C_{123} - C_{13} \\
P_3 &= C_{123} - C_{12}
\end{align*}
\]

Similarly, the portions of the total cost attributed to the interaction of any two vehicle classes, (1 and 2, 1 and 3, and 2 and 3) can be calculated using Equations (4.1), (4.2), and (4.3) and the initial cost estimates \( (C_1, C_2, C_3, \text{ and } C_{123}) \), as follows:

\[
\begin{align*}
P_{12} &= C_{123} - C_3 - P_1 - P_2 \\
P_{13} &= C_{123} - C_2 - P_1 - P_3 \\
P_{23} &= C_{123} - C_1 - P_2 - P_3
\end{align*}
\]
Figure 4.1 Input Cost Estimates
Figure 4.2 Cost Allocation Using the Modified Incremental Approach
Finally, the results from Equation (4.4), (4.5), and (4.6) are used to obtain \( P_{123} \), the total portion of the cost attributed to the interaction of all vehicle classes, as shown below:

\[
P_{123} = C_{123} - P_1 - P_2 - P_3 - P_{12} - P_{13} - P_{23} \tag{4.7}
\]

Figure 4.2(a) depicts the partitioning of the total cost \( C_{123} \) into the portions defined in Equations (4.1) through (4.7). As can be seen in this Figure, the allocated cost for vehicle class 1, for example, is equal to \( P_1 \) plus appropriate fractions of the portions \( P_{12}, P_{13}, \) and \( P_{123} \). These fractions can be defined in terms of relative facility usage, as measured by vehicle miles of travel (VMT). If \( V_1, V_2, \) and \( V_3 \) represent the number of VMTs associated with classes 1, 2, and 3, respectively, the final allocated cost \( R_1 \), is given by Equation (4.8):

\[
R_1 = P_1 + \frac{V_1}{V_1 + V_2} P_{12} + \frac{V_1}{V_1 + V_3} P_{13} + \frac{V_1}{V_1 + V_2 + V_3} P_{123} \tag{4.8}
\]

Similar results can be obtained for the cost allocations corresponding to classes 2 and 3:

\[
R_2 = P_2 + \frac{V_2}{V_1 + V_2} P_{12} + \frac{V_2}{V_2 + V_3} P_{23} + \frac{V_2}{V_1 + V_2 + V_3} P_{123} \tag{4.9}
\]

\[
R_3 = P_3 + \frac{V_3}{V_1 + V_3} P_{13} + \frac{V_3}{V_2 + V_3} P_{23} + \frac{V_3}{V_1 + V_2 + V_3} P_{123} \tag{4.10}
\]

Figure 4.2(b) represents the final cost allocations given in Equations (4.8), (4.9), and (4.10). In this Figure, it can be observed that the
Modified Incremental Method meets the completeness condition since the sum of the areas representing $R_1$, $R_2$, and $R_3$ is equal to the area representing the total cost $C_{123}$ of Figure 4.1. The shaded area shown in Figure 4.2(c) represents the marginal cost of vehicle class 1. As can be seen by comparing this Figure with Figure 4.2(a) this marginal cost is exactly equal to $P_1$. Also, comparing Figures 4.2(b) and 4.2(c) it is clear that $P_1 \leq R_1$. Therefore, the cost allocated to vehicle class 1 is at least equal to its marginal cost. This shows that the marginality requirement is satisfied. Similarly, the fact that $R_1 \leq C_1$ indicates that the cost allocation corresponding to class 1 in a joint design is less than it would be in a design intended only for class 1. This means that the rationality requirement is satisfied.

The Modified Incremental Approach does not have the inconsistency limitation of the standard incremental method, since it considers all possible combinations of vehicle classes and does not require that vehicle classes be included in any sequence. The development presented in this Section can be generalized for any number of vehicle classes.

4.2 The Generalized Method

This procedure is based on concepts from the theory of cooperative games [16,17]. A linear programming model which includes a set of meaningful economic constraints is formulated and solved to determine the appropriate cost allocation among the vehicle classes that share a transportation facility. Although the procedure developed in this Section is valid for any number of vehicle classes, it will be illustrated with three vehicle classes 1, 2, and 3. The same notation given in Section 4.1 will be used in this illustration.
The Generalized Method expresses the completeness, rationality, and marginality principles in terms of a mathematical model. The completeness requirement, which establishes that the vehicle classes must entirely finance a highway facility, is stated below:

\[ R_1 + R_2 + R_3 = C_{123} \]  \hspace{1cm} (4.11)

The rationality principle, which imposes the condition that the common facility must be the best alternative for all individual vehicle classes 1, 2, and 3 and for all subgroups of vehicle classes 1 and 2, 1 and 3, and 2 and 3, is represented as follows:

\[ R_1 \leq C_1 \] \hspace{1cm} (4.12)
\[ R_2 \leq C_2 \] \hspace{1cm} (4.13)
\[ R_3 \leq C_3 \] \hspace{1cm} (4.14)
\[ R_1 + R_2 \leq C_{12} \] \hspace{1cm} (4.15)
\[ R_1 + R_3 \leq C_{13} \] \hspace{1cm} (4.16)
\[ R_2 + R_3 \leq C_{23} \] \hspace{1cm} (4.17)

The marginality principle establishes that the cost allocations for vehicle classes 1, 2, and 3, and the sum of allocations for subgroups 1 and 2, 1 and 3, and 2 and 3, must at least equal the corresponding marginal costs; this requirement is expressed by the following relationships:
As indicated by Young et al. [18], if Constraint (4.11) holds, then Constraints (4.12)-(4.17) are equivalent to Constraints (4.18)-(4.23). This means that Constraints (4.18)-(4.23) are redundant and need not be considered in the analysis.

Constraints (4.11)-(4.17) define the set of feasible solutions for the cost allocation problem. This set is called the "core" [17] of the problem and is represented in Figure 4.3(a). In this Figure, the core is the shaded segment on the plane representing Constraint (4.11). The boundaries or sides of the core are indicated by Constraints (4.12)-(4.17).

The core may contain several solutions of which only one must be selected. One way to accomplish this is to systematically reduce the set of feasible solutions until it contains exactly one solution. The core reduction procedure is illustrated in Figure 4.3(b). The core is reduced by "moving" its sides (constraints) in the directions of the corresponding arrows while keeping them parallel to the original positions. Mathematically, the size of the core is reduced if an amount $t$ is subtracted from each right-hand side of Constraints (4.12)-(4.17). Since only one point is desired, the amount $t$ should be as large as possible without violating any of the Constraints. In conclusion, the core reduction procedure can be formulated in terms of the following linear programming model:

\[
\begin{align*}
R_1 &\geq C_{123} - C_{23} \\
R_2 &\geq C_{123} - C_{13} \\
R_3 &\geq C_{123} - C_{12} \\
R_1 + R_2 &\geq C_{123} - C_3 \\
R_1 + R_3 &\geq C_{123} - C_2 \\
R_2 + R_3 &\geq C_{123} - C_1
\end{align*}
\]
Figure 4.3 The Core in the Generalized Method
maximize \ t

subject to

\begin{align*}
R_1 & \leq C_1 - t \quad (4.24) \\
R_2 & \leq C_2 - t \quad (4.25) \\
R_3 & \leq C_3 - t \quad (4.26) \\
R_1 + R_2 & \leq C_{12} - t \quad (4.27) \\
R_1 + R_3 & \leq C_{13} - t \quad (4.28) \\
R_2 + R_3 & \leq C_{23} - t \quad (4.29) \\
R_1 + R_2 + R_3 & = C_{123} \quad (4.30) \\
R_1, R_2, R_3, t & \geq 0 \quad (4.31)
\end{align*}

4.3 Environmental Factors

An attractive feature of the generalized method is that it lends itself to a meaningful analysis of environmental costs. Environmental costs are those caused by factors other than traffic loads and, therefore, cannot be directly attributed to the individual vehicle classes.

The procedure described in this section can be easily extended to more than three vehicle classes. Only for convenience in the presentation it is assumed that only three classes are involved. The total number of vehicle combinations in this case is equal to 8. Each of these eight combinations can be represented in terms of a sequence of "+" and "-" signs, as indicated in Table 4.1. In this Table a negative sign indicates that a vehicle is not included in a combination, and a positive sign indicates that it is included. As an illustration, Combination 2 corresponds to a design for class 1 only with cost \( C_1 \), while Combination 4 corresponds to a design for classes 1 and 2, with cost \( C_{12} \). In particular, Combination 8 corresponds to a design for vehicle classes 1, 2, and 3; this is the design whose cost \( C_{123} \) is to be allocated to the three vehicle classes. Combination 1 corresponds
to a scenario with no vehicle classes. Since the cost $C_0$ associated with this scenario is not traffic-load related, it is assumed that it estimates the cost effect due to environmental factors.

It is always possible to express $C_0$ as a fraction of the total cost; that is,

$$C_0 = eC_{123} \quad (4.33)$$

where $e$ is an unknown number between 0 and 1. The methodology given in this Section can be used to find a maximal value for $e$ for given $C_1, C_2, \ldots, C_{123}$.

The proposed method is based on the concept of effects associated with a two-level factorial experiment [2]. This concept is illustrated here using Table 4.1. As can be seen in this Table, 4 combinations include vehicle class 1 and 4 combinations do not include it. The average cost associated with the combination not including class 1 is given by

$$E_1^- = (C_0 + C_2 + C_3 + C_{23})/4 \quad (4.34)$$

Similarly, the average cost associated with the vehicle combinations including class 1 is equal to

$$E_1^+ = (C_1 + C_{12} + C_{13} + C_{123})/4 \quad (4.35)$$

The statistical effect of class 1 is defined as $E_1^+ - E_1^-$ since this difference measures the average increase in cost due to vehicle class 1. Letting $E_1$ be equal to $E_1^+ - E_1^-$, and using Equations (4.34) and (4.35), $E_1$ can be written as
Table 4.1 Vehicle Combinations

<table>
<thead>
<tr>
<th>Combination No.</th>
<th>Vehicle Class 1</th>
<th>Vehicle Class 2</th>
<th>Vehicle Class 3</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$c_0$</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>$c_1$</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$c_2$</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>$c_{12}$</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$c_3$</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>$c_{13}$</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>$c_{23}$</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$c_{123}$</td>
</tr>
</tbody>
</table>
\[ E_1 = \frac{(C_1-C_2+C_12-C_3+C_13-C_23+C_123)}{4} - eC_{123}/4 \]  (4.36)

Setting \( A_1 = \frac{(C_1-C_2+C_12-C_3+C_13-C_23+C_123)}{4} \) and \( B = C_{123}/4 \), it is possible to rewrite Equation (4.36) as

\[ E_1 = A_1 - Be \]  (4.37)

The relationship given in Equation (4.37) is linear and indicates that the effect due to vehicle class 1 decreases as the impact of the environmental factors is increased. This behavior is illustrated in Figure 4.4(a).

A similar procedure is followed to find the relationships for vehicle classes 2 and 3. Figure 4.4(b) shows three hypothetical linear relationships for the three vehicle classes under consideration. Since \( E_1, E_2, \) and \( E_3 \) must be positive, the range for \( e \) is between zero and the minimal \( A_i/B \) value. In the case of the illustration given in Figure 4.4(b) this value is \( A_2/B \). In general, \( 0 < e < e' \) where

\[ e' = \min \{A_1/B, A_2/B, A_3/B\} \]  (4.38)

Summarizing, the cost effect due to the environmental factors can at most be a fraction \( e' \) of the total cost. Values of \( e \) exceeding \( e' \) are not valid since they would yield a negative value for the effect associated with at least one vehicle class.
Figure 4.4 Effects of the Vehicle Classes on Cost as Functions of $e$
5. APPLICATION OF THE METHODOLOGY

An application of the Modified Incremental Approach and the Generalized Method using a small sample from Texas pavement data is presented in this Section. Although realistic, these data are by no means comprehensive and are utilized only for illustrative purposes.

It is intended to allocate the estimated rehabilitation costs incurred in an analysis period of 18 years among four vehicle classes for a highway system consisting of two kinds of pavements. Table 5.1 describes the vehicle classes considered in this example, accumulated ESALs throughout the analysis period for each vehicle class, and percentages of VMTs corresponding to each vehicle class. Table 5.2 displays highway classification, pavement type, and pavement mileage for each of the two kinds of pavement.

A modification of the RENU program [8] was performed in order to obtain rehabilitation costs for the various vehicle combinations. Using these figures and ESAL data, rehabilitation costs were estimated for all vehicle combinations using the cost function discussed in Chapter 3. Table 5.3 gives the rehabilitation cost estimates associated with each vehicle class combination.

The results obtained from the Modified Incremental Approach and the Generalized Method are given below:

(a) **Modified Incremental Approach**

<table>
<thead>
<tr>
<th>Vehicle Class</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>$ 947,000</td>
</tr>
<tr>
<td>Class 2</td>
<td>$ 33,000</td>
</tr>
<tr>
<td>Class 3</td>
<td>$1,047,000</td>
</tr>
<tr>
<td>Class 4</td>
<td>$ 213,000</td>
</tr>
</tbody>
</table>
Table 5.1 Vehicle Class Data

<table>
<thead>
<tr>
<th>Vehicle Class</th>
<th>Truck Type</th>
<th>ESALs (millions)</th>
<th>VMT (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2D</td>
<td>3.590</td>
<td>96.43</td>
</tr>
<tr>
<td>2</td>
<td>3A</td>
<td>0.647</td>
<td>1.18</td>
</tr>
<tr>
<td>3</td>
<td>3-S2</td>
<td>15.317</td>
<td>2.06</td>
</tr>
<tr>
<td>4</td>
<td>2-S1-S2</td>
<td>5.172</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Table 5.2  Illustrative Pavement System.

<table>
<thead>
<tr>
<th>Pavement</th>
<th>Highway Classification</th>
<th>Pavement Type</th>
<th>Mileage (lane-miles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interstate</td>
<td>Flexible Overlaid</td>
<td>57</td>
</tr>
<tr>
<td>2</td>
<td>U. S.</td>
<td>Hot Mix</td>
<td>135</td>
</tr>
</tbody>
</table>

43
Table 5.3 Rehabilitation Cost Estimates

<table>
<thead>
<tr>
<th>Combination</th>
<th>Cost (millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.06</td>
</tr>
<tr>
<td>2</td>
<td>0.76</td>
</tr>
<tr>
<td>1,2</td>
<td>1.11</td>
</tr>
<tr>
<td>3</td>
<td>1.87</td>
</tr>
<tr>
<td>1,3</td>
<td>2.04</td>
</tr>
<tr>
<td>2,3</td>
<td>1.90</td>
</tr>
<tr>
<td>1,2,3</td>
<td>2.06</td>
</tr>
<tr>
<td>4</td>
<td>1.18</td>
</tr>
<tr>
<td>1,4</td>
<td>1.46</td>
</tr>
<tr>
<td>2,4</td>
<td>1.24</td>
</tr>
<tr>
<td>1,2,4</td>
<td>1.51</td>
</tr>
<tr>
<td>3,4</td>
<td>2.105</td>
</tr>
<tr>
<td>1,3,4</td>
<td>2.22</td>
</tr>
<tr>
<td>2,3,4</td>
<td>2.13</td>
</tr>
<tr>
<td>1,2,3,4</td>
<td>2.24</td>
</tr>
</tbody>
</table>
(b) **Generalized Method**

<table>
<thead>
<tr>
<th>Vehicle Class</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Class 1</td>
<td>$410,000</td>
</tr>
<tr>
<td>Vehicle Class 2</td>
<td>$320,000</td>
</tr>
<tr>
<td>Vehicle Class 3</td>
<td>$1,030,000</td>
</tr>
<tr>
<td>Vehicle Class 4</td>
<td>$480,000</td>
</tr>
</tbody>
</table>

It can be verified that both methods yield results which are consistent with the completeness, rationality and marginality principles. A considerable difference in the results can be observed between the two methods. This difference is explained by the influence of the measure of highway usage (VMT) on the allocation of common costs in the Modified Incremental Approach. In this example, a significant portion of the total cost $C_{1234}$ is attributed to the interaction of all vehicle classes. A large percentage of this portion is allocated to vehicle class 1 due to the high percent of VMTs associated with it. On the other hand, the Generalized Method distributes the cost among the vehicle classes without considering VMTs; in case that the degree of pavement damage, and not highway utilization, is the dominant criterion in the decision making process, the generalized results are appropriate. The maximum percentage of the total cost that can be attributed to the environment $e'$ is equal to 45%, as indicated by Equation (4.38).
6. SUMMARY

This report summarizes the work performed in relation to study 2-18-83-332 "Analysis of Truck Use and Highway Cost Allocation in Texas" during F. Y. 83-84. This Fiscal Year, emphasis was placed on the development of a sound conceptual methodology for the allocation of costs related to the provision and upkeeping of highway facilities.

In particular, two cost allocation methods were developed: the Modified Incremental Approach and the Generalized Method. These methods exhibit significant conceptual advantages over those previously used in the context of highway cost allocation. In particular, they fulfill the following conditions: (a) highway costs are completely financed by users; (b) vehicle classes reduce their cost responsibilities by sharing the facilities with other vehicle classes; and (c) vehicle are charged at least enough to cover their corresponding marginal costs.

In addition, a procedure was developed to assess a range of values for the percentage of the total rehabilitation and maintenance costs that can be attributed to the effect of the environment, that is, independent of traffic.

The proposed methodology was illustrated using two representative sections from the Texas pavement data base.
7. REFERENCES


APPENDIX 1

MODIFIED RENU2 FORTRAN CODE
PROGRAM TO DETERMINE EFFECT OF LEGAL LOAD LIMITS ON LONG-RANGE PAVEMENT COSTS.

THIS VERSION CREATED AUG 7-1981
ISN 0034 WRITE (6,600) TITLES,NVC
ISN 0035 600 FORMAT (1X,5A4,15)
ISN 0036 NCOMB=2**NVC-1
ISN 0037 DO 1000 ICOMB=1,NCOMB
ISN 0038 DO 1 I=1,8
ISN 0039 DO 1 J=1,30
ISN 0040 PIH(J,I)=0.
ISN 0041 QIH(J,I)=0.
ISN 0042 1 CONTINUE
ISN 0043 CALL INIT(1)
ISN 0044 CALL COMGEN
ISN 0045 REWIND 10
ISN 0046 100 CALL INPUT (IGO)
ISN 0047 GO TO (110, 200, 300,300), IGO
ISN 0048 110 CALL INIT(2)
ISN 0049 CALL INPRNT
ISN 0050 CALL EALGET
ISN 0051 CALL COSCAL
ISN 0052 GO TO 100
ISN 0053 200 CONTINUE
ISN 0054 GO TO 100
ISN 0055 300 CONTINUE
ISN 0056 1000 CONTINUE
ISN 0057 ENDFILE 2
ISN 0058 STOP
ISN 0059 END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 58, PROGRAM SIZE = 974, SUBPROGRAM NAME = MAIN
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 920K BYTES OF CORE NOT USED
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

ISN 0002    SUBROUTINE NPAGE
ISN 0003    RETURN
ISN 0004    END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 3, PROGRAM SIZE = 164, SUBPROGRAM NAME = NPAGE

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ******

928K BYTES OF CORE NOT USED
OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NLIST NODECK OBJECT NOMAP NFORMAT GOSTMT NXREF ALC NOANSF NOTERM IBM FLAG(I)

** ISN 0002 BLOCK DATA
** ISN 0003 COMMON /TEMPC/ CONTP(25), DISTCT
** ISN 0004 COMMON /MECH/KXT, NRU, NLH, NDEL, AACR, IYR, JYR, CONSTR(20)
** ISN 0005 COMMON/HOR/A(10), B(10), C(10), DT(10), DF(10), S(10), T(10), TR(5), PI(5)
* PI(5), AC(5), AA, SAT(5), XMNW(18(10)), XKT
** ISN 0006 COMMON /EXTRA/ PTVTK, TPE, PTD, XMNDTK, XMOTK, NIS
** ISN 0007 COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
** ISN 0008 COMMON /EXPTV/ NPT, THICK(4), MTYIP(4), NLAY, IP, IF, IR, IC
** ISN 0009 COMMON /FUNDS/ APPOF(20, 2), RINT, RTNF
** ISN 0010 COMMON /IO/ LI, LG, LD
** ISN 0011 COMMON /LABELS/ MATLAB(5, 10)
** ISN 0012 COMMON /LMP/ XLM(30), YLM(30), POTLM(20, 2), OUTF(20, 2)
** ISN 0013 COMMON /MISC/ IPOT, IARMS, OLDMT, AGF
** ISN 0014 COMMON /DVRAY/ XHCX, XHCM, WLACE, WPSH, PPSH, CAC, CGR
** ISN 0015 COMMON /PSI/ PF, PICON, PTERM, PIVD, PTOV
** ISN 0016 COMMON /STEER/ EQFACT(15, 5), PST(4)
** ISN 0017 COMMON /STRUC/ SN, SS, R, D, AGG, XU, XK, E
** ISN 0018 COMMON /STROE/ STRCD(8), CC(11), NC, STRC(5), RFS(4), RFB(4)
** ISN 0019 COMMON /TIME/ ATPL, NPLI, NPL, YNTR, YR(40)
** ISN 0020 COMMON /SLVG/ ISLV, FLRP, VI(30), RI(30), VL(30), RL(30),
** U(30), PL(30), MI(30), PC(20), VP(20), RP(20),
** PB, VB, VPB, NS, NY, SW(6, 2), SVB, FLRTP(4)
** ISN 0021 COMMON /DOMI/ TOUTP(8), TTDC(8), TTT(8, 20, 3)

C ***********************************************************************
C VARIABLES COMPARISON BETWEEN AASHO & TEXAS EQUATIONS
C
C TEXAS AASHO DESCRIPTIONS
C
C C(1) ALF HARMONIC MEAN TEMPERATURE
C C(2) TI THORN THWAIT INDEX
C C(3) FTC ANNUAL AVERAGE FREEZE-THAW CYCLES
C C(4) WFTC
C C(5) PR ANNUAL AVERAGE RAINFALL
C C(6) TM MEAN MONTHLY TEMPERATURE
C D(1) DMD MAXIMUM DEFLEXION
C D(2) SCI SURFACE CURVATURE INDEX
C D(3) VOL VOLUME OF DINA FLEX BASIN
C DT(1) AS ASPHALT STIFFNESS
C S(1) TTC TEXAS TRIAXIAL CLASS
C S(2) SII LIQUID LIMIT
C S(3) SPI PLASTICITY INDEX
C S(4) SPP PERCENT PASSING #200
C T(1) T AGE IN YEARS
C TR(1) ADT AVERAGE DAILY TRAFFIC
C TR(2) 18-KIP 18-KIP SINGLE AXLE LOADS
C TR(NPT) W 18-KIP SINGLE AXLE LOADS
C
C *** REFER TO SUBROUTINES PSIT & RUTA
C
C ****************************~*********************

DATA NAPOV, PAPOV, SIZE, AVRG /21, 5.0, 2.0, 100./
DATA XHCIO/0.0/.XHCM/0.0/
DATA PICON, PTERM, PIOV, PTOV / 4*1.0 /
DATA IF, IR, IC / 1, 2, 3 /
DATA LI, L0, LD / 10, 6, 1 /
DATA SS, S, AGG, X, E / 3.1, 1, 195.43, 150.40 /
DATA NYAP, OVLIF, ATP, NYR / 20, 20, 20, 40 /
DATA RTINT, RTINF / 0.0, 0.0 /
DATA IF, IR, IC / 1, 2, 3 /
DATA NYAP, OVLIF, ATP, NYR / 20, 20, 20, 40 /
DATA PPF, TPF, PFNO / 0.0, 0.0, 0.0 /
DATA RTINT, RTINF / 0.0, 0.0 /
DATA PTST / 1.5, 2.0, 2.5, 3.0 /
DATA EQFACT / 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, 16.0, 18.0, 20.0, 22.0 /
1 24.0, 26.0, 28.0, 30.0
2 .0005, .0008, .04, .13, .28, .52, .92, 1.42, 2.12, 3 2.95, 4.02, 5.29, 6.73, 8.31, 10.19, 4 .0009, .01, .05, .14, .31, .54, .86, 1.31, 1.94, 5 2.52, 3.35, 4.4, 5.49, 6.67, 8.05, 6 .002, .02, .06, .18, .36, .62, .93, 1.33, 1.9, 2.44, 7 3.15, 3.95, 4.82, 5.83, 6.8, 8 .004, .03, .09, .23, .41, .66, .94, 1.28, 1.74, 9 2.16, 2.7, 3.28, 3.89, 4.59, 5.33, 6

C ****************************~*********************

DATA STRCD / .44, .34, .23, .14, .30, .18, .11, .14 /
DATA STRCD / .44, .34, .23, .14, .30, .18, .11, .14 /
DATA RFS / .9, .7, .5 /
DATA RFS / .9, .7, .5 /
DATA CC / 1.0, 0.85, 0.75, 0.65 /
*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINCOUNT(50) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 59, PROGRAM SIZE = 0, SUBPROGRAM NAME = TEMPC

*STATISTICS* NO DIAGNOSTICS GENERATED

***** ENDP OF COMPILATION *****

912K BYTES OF CORE NOT USED
SUBROUTINE COMGEN
COMMON /COMBI/ ICOMB, NVC, COFVCT(6)
DO 100 IND=1,NVC
100 CONTINUE
IND = 1
INQUOT = ICOMB
500 IF (INQUOT .EQ. 0) GO TO 900
QUOT = FLOAT(INQUOT)/2.
INQUOT = INT(QUOT)
IRES = INT((QUOT-FLOAT(INQUOT))*2.)
COFVCT(IND) = FLOAT(IRES)
IND = IND+1
GO TO 500
900 RETURN
END
C

SUBROUTINE INPUT (IGO)
COMMON /TEMPC/ CONTP(25),DISTCT
COMMON /EXTRA/ PTOVTK, TPE, PFD, XNQOTK, XMQOTK, NIS
COMMON /MECH/ KXT, RRL, NLM, ND, NOEL, IACR, IYR, JYR, CONSTR(20)
COMMON /EALPAY/ EALPT(10,2), APT(10,2), EALFCT(20), IEQTNP
COMMON /PTOV/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /FUNDS/ APQF(20,2), RTINT, RTINF
COMMON /INTVLS/ STARTS(6)
COMMON /LABELS/ MTLAB(5,10)
COMMON /LDS/ PGWVL, PSAL, PTAL, PTRLAL, FGVWL, FSAL, FTAL, FTRLAL
COMMOM /LMP/ XLM(30), YLM(30), PTLM(20,2), OUTP(20,2), TMTLM, PPF,
1 PTF, PFNO, NXL, NSLR, TOVL(30,2), XLM(20)
COMMON /MISC/ IPT, LARMS, LOLMNT, AGF
COMMON /NEWSYS/ NEWSYS
COMMON /NMTR/ SA(30,11), TA(30,11), TR(50,11), VE(30,11),
1 VG(75,11), NLDI(6), EPI(10), ST(30,11)
COMMON /OUTSW/ IOUT
COMMON /OVRLAY/ XHCIO, XHCIM, WLNE, WPSH, WGSH, PPVDSH, CAC, CGR
1 CSOCAT
COMMON /PSI/ PF, PICON, PTERM, PID, PTOV
COMMON /STRCD/ STRCD(8), CC(4), MC(11), NC, STRC(5), RFS(4), RFB(4)
COMMON /STRUC/ SN, SS, R, D, AGG, XJ, XK, E
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /TITLE/ TITLE(20,3), SCETTL(20)
COMMON /TRYP/ TRYP(2,10), TTYP(10,20,2), PCTTR(20,2), PERCT(4,
1 NAXL(10,4), NT(4), NTY, NAT, NTT, NEWTN
COMMON /SLVG/ ISLV, FLRP, VI(30), RI(30), VL(30), RL(30),
1 U(30), PL(30), WI(30), P(20), VP(20), RP(20),
2 PB, PB1, P, NS, NY, SV(6,2), SB, FLRPTP(4)
COMMON /SWTCH/ OVLIFE, PCTIIN, PCTINF, TFPCC, PFNDCP, AGR, SPCJT,
1 XMLI, CACI, CGL, IAC, AGCN, ICGR, GREDN,
2 INTT, SAVMNT, IDST, NLD, MCDE(5), TFCDNS
DIMENSION KWORD(5), IVAL(2), VAL(5), KEY(22), STRCIN(5)
DATA ISTOP /4HSTOP/
DATA SATP /0./
DATA KEY /4HSTOR, 4XEC, 4FLEX, 4HRIGI, 4HERF, 4HAGE, 4Hoyer,
1 4HMODE, 4HIST, 4HMDM, 4HTRUC, 4HYSY, 4HOLD, 4HURL, 4HLOAD, 4HSING, 4HTAND, 4HTRUD, 4HGVW, 4HHEMT, 4HSTEE, 4HOUTP/
DATA IACO /4HACO /
DATA NWKEY /22/
IDST = O
NEWTRK = O
NEWSYS = O
ATP = SATP
CALL NPAGE
READ AND ECHO PRINT A KEYWORD CARD
2 READ (LI,3) KWORD, IVAL, VAL
LEVEL 2.3.0 (JUNE 78)  INPUT  OS/360 FORTRAN H EXTENDED  DATE 84.262/15.03.07  PAGE 2

3 FORMAT(5A4,215,5F10.0)
WRITE (L0,4) KWORD, IVAL, VAL
4 FORMAT(1X,5A4,215,5(F10.2,2X))

TEST FOR NORMAL PROGRAM TERMINATION
IF (KWORD(1) .EQ. ISTOP) GO TO 9992
SEARCH THE KEY TABLE FOR THE KEYWORD READ IN
DO 10 I=1,NKEY
   IKEY = I
   IF (KWORD(1) .EQ. KEY(I)) GO TO 15
CONTINUE
GO TO 9996
15 GO TO (9998, 9997, 100, 200, 300, 400, 500, 600, 700, 800, 900,
   1 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900,
   2 2000), IKEY
*** FLEXIBLE SECTION ***
100 IP :: IF WLAN = VAL(1)
   WDTH = WLAN
   SS = VAL(2)
   R = VAL(3)
   PF=VAL(4)
   PFD=VAL(5)
READ A TITLE CARD FOR THIS SECTION
READ (LI,101) SECTTL
101 FORMAT (20A4)
WRITE (L0,103) SECTTL
103 FORMAT (1X,20A4)
   IF(IP.EQ.IR) GO TO 105
READ AND ECHO PRINT THE MATERIALS CARD
READ(LI,19) NDIST,NIS,NPT,NRU,NLH,NDEL,TPE,XMNOK,XMXOK,PTOVK,
   1 IACR,IYR,JYR
19 FORMAT(7I5,3F5.0,3I5)
WRITE(L0,21) NDIST,NIS,NPT,NRU,NLH,NDEL,TPE,XMNOK,XMXOK,PTOVK,
   1 IACR,IYR,JYR
21 FORMAT(1X,7I5,3F5.2,3I5)
READ(LI,20)(CONSTR(I),I=1,20)
20 FORMAT(15F5.0)
WRITE(L0,22) (CONSTR(I),I=1,20)
22 FORMAT(1X,15F8.1/1X,5F8.1)
READ (LI,110) (MCODE(I), THICK(I), STRCIN(I), I=1,4)
105 FORMAT(1X,110) (MCODE(I), THICK(I), STRCIN(I), I=1,4)
IF(IP.EQ.IR) GO TO 1010
MCODE(1)=MC(1)
MCODE(2)=MC(4)
MCODE(3)=MC(8)
C
C THICK REPRESENTS THE LAYER THICKNESSES OF REPRESENTATIVE
C SECTIONS

ISN 0085 IF(THICK(1).NE.0) GO TO 1010
ISN 0087 IF(NPT.NE.3.OR.NRU.NE.1) GO TO 50
ISN 0089 THICK(1)=.75
ISN 0090 THICK(2)=6.0
ISN 0091 GO TO 1010
ISN 0092 IF(NPT.NE.3.OR.NRU.NE.2) GO TO 51
ISN 0094 THICK(1)=0.75
ISN 0095 THICK(2)=8.0
ISN 0096 GO TO 1010
ISN 0097 IF(NPT.NE.1.OR.NRU.NE.1.OR.NLH.NE.1) GO TO 52
ISN 0099 THICK(1)=2.0
ISN 0100 THICK(2)=8.0
ISN 0101 GO TO 1010
ISN 0102 IF(NPT.NE.1.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 53
ISN 0104 THICK(1)=4.0
ISN 0105 THICK(2)=12.0
ISN 0106 GO TO 1010
ISN 0107 IF(NPT.NE.1.OR.NRU.NE.2.OR.NLH.NE.1) GO TO 54
ISN 0109 THICK(1)=2.0
ISN 0110 THICK(2)=8.0
ISN 0111 THICK(3)=6.0
ISN 0112 GO TO 1010
ISN 0113 IF(NPT.NE.1.OR.NRU.NE.2.OR.NLH.NE.2) GO TO 55
ISN 0115 THICK(1)=4.0
ISN 0116 THICK(2)=10.0
ISN 0117 THICK(3)=6.0
ISN 0118 GO TO 1010
ISN 0119 IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.1) GO TO 56
ISN 0120 MCODE(2)=MC(2)
ISN 0121 MCODE(3)=MC(4)
ISN 0122 MCODE(4)=MC(8)
ISN 0123 IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 57
ISN 0124 THICK(1)=2.0
ISN 0125 THICK(2)=2.0
ISN 0126 THICK(3)=8.0
ISN 0127 GO TO 1010
ISN 0128 IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 58
ISN 0130 THICK(1)=3.0
ISN 0131 THICK(2)=4.0
ISN 0132 THICK(3)=12.0
ISN 0133 GO TO 1010
ISN 0134 IF(NPT.NE.4.OR.NRU.NE.2.OR.NLH.NE.1) GO TO 59
ISN 0136 THICK(1)=2.0
ISN 0137 THICK(2)=2.0
ISN 0138 THICK(3)=8.0
ISN 0139 THICK(4)=6.0
ISN 0140 IF(NPT.NE.4.OR.NRU.NE.2.OR.NLH.NE.2) GO TO 1010
ISN 0142 THICK(1)=3.0
ISN 0143 THICK(2)=4.0
ISN 0144 THICK(3)=10.0
ISN 0145 THICK(4)=6.0
ISN 0146 CONTINUE
ISN 0147 110 FORMAT(5(A3.2X,2F5.0,1X))
ISN 0148 WRITE (LO,120) (MCODE(I), THICK(I), STRCIN(I), I=1,4)
Determine the number of layers in the pavement structure.

C

IPFLG = 0
DO 140 I=1,4
IF (THICK(I) .LE. 0.0) GO TO 160
NLAY = I
STRC(I) = STRCIN(I)
DO 135 J= 1,NC
IF (MCODE(I) .NE. MCODE(J)) GO TO 135
CONTINUE
GO TO 9993
140 CONTINUE
IF (IPFLG .EQ. 0) GO TO 165
IF (MTYPE(2) .NE. 9 .AND. MTYPE(2) .NE. 10) GO TO 9989
NIS = 1
IP = IC
165 STRC(5) = STRC(1)
MCODE(5) = IACO
GO TO 2

*** Rigid Section ***

C

200 IP = IR
WLANE = VAL(1)
WDTH = WLANE
XX = VAL(2)
IF (VAL(3) .NE. 0.0) AGG = VAL(3)
IF (VAL(4) .NE. 0.0) E = VAL(4)
IF (VAL(5) .NE. 0.0) DISTCT = VAL(5)
GO TO 101

*** Performance Section ***

C

300 PICON = VAL(1)
PTERM = VAL(2)
PIOV = VAL(3)
PTOV = PTERM
OVLIFE = VAL(4)
NVAP = NYAP
IF (VAL(4) .GT. 0.) OVLIF = VAL(4)
READ (LI,310) ATP
310 FORMAT(3F10.0)
WRITE (LO,320) ATP
IF(ATP.LT.1) ATP=13.
320 FORMAT(1X,8F10.2)
SATP = ATP
GO TO 2

*** Age Distribution Section ***

C

400 NASL = IVAL(1)
ISN 0207
ISLVS VAL(2)

ISN 0208
FLRP = VAL(1)

C READ AND ECHO PRINT THE DISTRIBUTION OF LANE MILES BY AGE

C

ISN 0209
READ (LI,410) (YL,(I),I=1,NASL)

ISN 0210
410 FORMAT (16F5.0,/,14F5.0)

ISN 0211
WRITE (LO,420) (YL,(I),I=1,NASL)

ISN 0212
420 FORMAT (1X,15F8.1/1X,15F8.1)

ISN 0213
IF (ISLV .EQ. 0) GO TO 404

ISN 0215
READ (LI,430) (VI(I),I=1,NASL)

ISN 0216
WRITE (LO,320) (VI(I),I=1,NASL)

ISN 0217
430 FORMAT (16F5.0)

ISN 0218
READ (LI,430) (RI(I),I=1,NASL)

ISN 0219
WRITE (LO,320) (RI(I),I=1,NASL)

ISN 0220
404 IF (NASL.LE.25) GO TO 421

ISN 0222
DO 422 I=26,NASL

ISN 0223
YLM(25)=YLM(25)+YLM(I)

ISN 0224
NASL=25

ISN 0225
421 CONTINUE

ISN 0226
GO TO 2

C *** OVERLAY SECTION ***

C

ISN 0227
500 ICAC = VAL(1)

ISN 0228
ICGR = VAL(2)

C READ AND ECHO PRINT THE OVERLAY PARAMETERS

C

ISN 0229
READ (LI,510) PPVDSH,WPSH, WGS, CACI, CGRI, ACDENS, GRDENS,CSCDA

ISN 0230
510 FORMAT(15F10.0)

ISN 0231
WRITE (LO,520) PPVDSH,WPSH, WGS, CACI, CGRI, ACDENS, GRDENS,CSCDA

ISN 0232
520 FORMAT (1X,15F8.2)

ISN 0233
GO TO 2

C *** MODEL MAINTENANCE SECTION ***

C

ISN 0234
600 IARMS = VAL(1)

ISN 0235

MFLG = 1

C READ AND ECHO PRINT THE UNIT COSTS FOR BOTH FLEXIBLE AND RIGID
C PAVEMENTS, AND THE JOINT SEALING PARAMETERS

C

ISN 0236
READ (LI,610) (UNTCST(I),I=1,3)

ISN 0237
610 FORMAT(3F10.0)

ISN 0238
READ (LI,620) UNTCST(4),DISS,DCON,DIN

ISN 0239
WRITE (LO,630) (UNTCST(I),I=1,4), DISS, DCON, DIN

ISN 0240
630 FORMAT(1X,3F10.2/1X,6F10.2,15)

ISN 0241
620 FORMAT(4F10.0,2F5.0,15)

ISN 0242
GO TO 2

C *** HISTORICAL MAINTENANCE SECTION ***

C

ISN 0243
700 IARMS = VAL(1)

ISN 0244

MFLG = 2

C READ AND ECHO PRINT THE MAINTENANCE COSTS PER LANE MILE BY AGE FOR
C FLEXIBLE PAVEMENTS
C READ (LI,710) (USRMDL(I,1),I=1,24)
ISN 0245
C 710 FORMAT(8F10.0)
ISN 0246
C WRITE (LO,720) (USRMDL(I,1),I=1,24)
ISN 0247
C 720 FORMAT(1X,8F10.0)
ISN 0248
C READ AND ECHO PRINT THE MAINTENANCE COSTS PER LANE MILE BY AGE FOR
C RIGID PAVEMENTS
C
ISN 0249
C READ (LI,710) (USRMDL(I,2),I=1,24)
C WRITE (LO,720) (USRMDL(I,2),I=1,24)
ISN 0250
C GO TO 2
C
ISN 0251
C *** NO MAINTENANCE SECTION ***
ISN 0252
C 800 MFLG = 0
ISN 0253
C GO TO 2
C
ISN 0254
C *** TRUCK TYPES SECTION ***
C
ISN 0255
C 900 NTTV = IVAL(1)
ISN 0256
C NATT = IVAL(2)
ISN 0257
C PERCT(1)=VAL(1)
ISN 0258
C PERCT(2)=VAL(2)
ISN 0259
C PERCT(3)=VAL(3)
ISN 0260
C PERCT(4)=VAL(4)
ISN 0261
C NEWTRK = NEWTRK + 1
ISN 0262
C IF ((NTTV+NATT) .GT. 10) GO TO 9995
ISN 0263
C NTT = NTTV
ISN 0264
C INTT = NTT + NATT
ISN 0265
C READ AND ECHO PRINT THE TRUCK LABELS
C
ISN 0266
C READ (LI,910) ((TTYP(M,J),M=1,2),J=1,INTT)
ISN 0267
C 910 FORMAT(8(2A4,2X))
ISN 0268
C WRITE (LO,920) ((TTYP(M,J),M=1,2),J=1,INTT)
ISN 0269
C 920 FORMAT(1X,8(2A4,2X))
ISN 0270
C READ AND ECHO PRINT THE AXLE CONFIGURATIONS
C
ISN 0271
C READ (LI,921) ((NAXLES(M,J),J=1,4),M=1,INTT)
ISN 0272
C 921 FORMAT(8(4I2,2X))
ISN 0273
C WRITE (LO,922) ((NAXLES(M,J),J=1,4),M=1,INTT)
ISN 0274
C 922 FORMAT(1X,8(4I2,2X))
ISN 0275
C DD 929 J=1,4
ISN 0276
C NT(J) = 0
ISN 0277
C DO 928 M=1,NTT
ISN 0278
C NT(J) = NT(J) + NAXLES(M,J)
ISN 0279
C 928 CONTINUE
ISN 0280
C CONTINUE
ISN 0281
C C READ AND ECHO PRINT THE TRUCK PERCENTAGES
ISN 0282
C 935 K = K+1
ISN 0283
C DD 930 N=1,NYAP
ISN 0284
C READ (LI,930) I, ((TTYP(J,1,K),J=1,10), PCTTR(I,K)
ISN 0285
C 930 FORMAT(I3,1X.11F6.0)
WRITE (LO, 940) I, (PTTYP(J, I, K), J=1, 10), PCTTR(I, K)
ISN 0285 950 CONTINUE
ISN 0286 IF ((NAT.GT. 0) .AND. (K.EQ. 1)) GO TO 935
ISN 0288 IF (K.EQ. 2) GO TO 2
ISN 0289 DO 970 J=1, 10
ISN 0290 DO 960 I=1, 20
ISN 0291 PTTYP(J, I, 2) = PTTYP(J, I, 1)
ISN 0292 960 CONTINUE
ISN 0293 970 CONTINUE.
ISN 0295 GO TO 2
C
C *** TITLE CARD SECTION. ***
C
C READ AND ECHO PRINT THE THREE TITLE CARDS
C
ISN 0296 1000 DD 1030 I=1, 3
ISN 0297 READ (LI, 102) (TITLE(I,J), I=1, 20)
ISN 0298 WRITE (LO, 103) (TITLE(I,J), I=1, 20)
ISN 0299 1030 CONTINUE
ISN 0300 NEWSYS = 
ISN 0301 GO TO 2
C
C *** OLD SECTIONS ***
C
ISN 0302 1100 SAVMAT = VAL(1)
ISN 0303 IPOT = IVAL(1)
ISN 0304 IPF = IVAL(2)
ISN 0305 IF (IPOT .EQ. 0 OR IPOT .EQ. 3) GO TO 2
ISN 0307 IF (IPOT .EQ. 1) GO TO 1150
ISN 0309 PFNPNC = VAL(3)
ISN 0310 PCTINF = VAL(4)
C
C READ AND ECHO PRINT THE ANNUAL PROJECTED OVERLAY FUNDS FOR PRESENT
C REGULATIONS
C
ISN 0311 READ (LI, 1110) (APOF(I, 1), I=1, NYPAP)
ISN 0312 1110 FORMAT(8F10.0)
ISN 0313 WRITE (LO, 1120) (APOF(I, 1), I=1, NYPAP)
ISN 0314 1120 FORMAT(8X,8F10.0)
ISN 0315 IF (IPF .EQ. 1) GO TO 1140
ISN 0317 DO 1130 I=1, NYPAP
ISN 0318 APOF(I, 2) = APOF(I, 1)
ISN 0319 1130 CONTINUE
ISN 0320 GO TO 2
C
C *** RUN PARAMETERS ***
C
ISN 0321 1140 READ (LI, 1110) (APOF(I, 2), I=1, NYPAP)
ISN 0322 WRITE (LO, 1120) (APOF(I, 2), I=1, NYPAP)
ISN 0323 GO TO 2
ISN 0324 1150 TPFPC = VAL(2)
ISN 0325 PFNPNC = VAL(3)
ISN 0326 GO TO 2
LEVEL 2.3.0 (JUNE 78)  INPUT  05/360 FORTRAN H EXTENDED  DATE 84.262/15.03.07  PAGE 8

ISN 0327  1200 IF (IVAL(1) .NE. 0) NYAP = MINO(IVAL(1),20)
ISN 0329  IEQTRP = IVAL(2)
ISN 0330  AGR = IVAL(1)
ISN 0331  PCTINT = VAL(2)
ISN 0332  IF(VAL(3).NE.0.O)XHCO=VAL(3)
ISN 0334  IF(VAL(4).NE.0.O)XHCM=VAL(4)
ISN 0336  TFCDNS=VAL(5)
ISN 0337  GO TO 2

C *** LOAD LIMITS SECTION ***
C READ THE PRESENT AND FUTURE LOAD LIMITS
C
ISN 0338  1300 Iews = IVAL(1)
ISN 0339  IDST = 1
ISN 0340  NEWTRK = NEWTRK + 2
ISN 0341  READ (LI,1310) PGVWL, PSAL, PTAL, PTRAL
ISN 0342  1310 FORMAT(4F10.0)
ISN 0343  WRITE (LO,1315) PGVWL, PSAL, PTAL, PTRAL
ISN 0344  1315 FORMAT(1X,4F10.2)
ISN 0345  READ (LI,1310) FGVWL, FSAL, FTAL, FTRAL
ISN 0346  WRITE (LO,1315) FGVWL, FSAL, FTAL, FTRAL

C READ THE PRESENT AND FUTURE STEERING AXLE WEIGHTS FOR EACH TRUCK TYPE
C
ISN 0347  NTT = INTT
ISN 0348  READ (LI,1320) (PSTAW(I),I=1,NTT)
ISN 0349  READ (LI,1320) (FSTAW(I),I=1,NTT)
ISN 0350  1320 FORMAT(10F8.0)
ISN 0351  WRITE (LO,1325) (PSTAW(I),I=1,NTT)
ISN 0352  WRITE (LO,1325) (FSTAW(I),I=1,NTT)
ISN 0353  1325 FORMAT(1X,10F8.0)

C READ THE NEW EMPTY WEIGHT (AS A PERCENTAGE OF THE CURRENT EMPTY WEIGHT)
C FOR EACH TRUCK TYPE
C
ISN 0354  IF (IEWS .EQ. 0) GO TO 2
ISN 0356  READ (LI,1320) (EP(I),I=1,NTT)
ISN 0357  WRITE (LO,1330) (EP(I),I=1,NTT)
ISN 0358  1330 FORMAT(1X,10F8.2)
ISN 0359  GO TO 2

C *** SINGLE AXLE SECTION ***
C
ISN 0360  1400 NLDI(1) = IVAL(1)
ISN 0361  NLD = IVAL(1)
ISN 0362  NTT = INTT
ISN 0363  STARTS(1) = VAL(1)
ISN 0364  NEWTRK = NEWTRK + 2

C READ THE LOAD INTERVALS AND, FOR EACH TRUCK TYPE, THE NUMBER OF
C SINGLE AXLES FOR EACH INTERVAL
C
ISN 0365  DO 1420 L=1,NLD
ISN 0366  READ (LI,1410) ELDINT, (SA(L,J),J=1,NTT)
ISN 0367  1410 FORMAT(F10.0,10F7.0)
ISN 0368  WRITE (LO,1415) ELDINT, (SA(L,J),J=1,NTT)
ISN 0369  1415 FORMAT(1X,F10.0,10F7.0)
ISN 0370  SA(L,11) = ELDINT
ISN 0371  1420 CONTINUE
ISN 0372  DO 1422 K=1,NLD
ISN 0373  SA(K,2)=0.000001
ISN 0374  SA(K,3)=0.000001
ISN 0375  1422 CONTINUE
ISN 0376  GO TO 2

C *** TANDEM AXLE SECTION ***
C
ISN 0377  1500 NLDI(2) = IVAL(1)
ISN 0378  NLD = IVAL(1)
ISN 0379  NTT = INTT
ISN 0380  STARTS(2) = VAL(1)
ISN 0381  NEWTRK = NEWTRK + 2

C READ THE LOAD INTERVALS AND NUMBER OF DOUBLES PER TRUCK TYPE PER INTERVAL
C
ISN 0382  DO 1510 L=1,NLD
ISN 0383  READ (LI,1410) ELDINT, (TA(L,J),J=1,NTT)
ISN 0384  WRITE (LO,1415) ELDINT, (TA(L,J),J=1,NTT)
ISN 0385  TA(L,11) = ELDINT
ISN 0386  1510 CONTINUE
ISN 0387  GO TO 2

C *** TRIPLE AXLE SECTION ***
C
ISN 0388  1600 NLDI(3) = IVAL(1)
ISN 0389  NLD = IVAL(1)
ISN 0390  NTT = INTT
ISN 0391  STARTS(3) = VAL(1)
ISN 0392  NEWTRK = NEWTRK + 2

C READ THE LOAD INTERVALS AND NUMBER OF TRIPLES PER TRUCK TYPE PER INTERVAL
C
ISN 0393  DO 1610 L=1,NLD
ISN 0394  READ (LI,1410) ELDINT, (TR(L,J),J=1,NTT)
ISN 0395  WRITE (LO,1415) ELDINT, (TR(L,J),J=1,NTT)
ISN 0396  TR(L,11) = ELDINT
ISN 0397  1610 CONTINUE
ISN 0398  GO TO 2

C *** GROSS VEHICLE WEIGHT SECTION ***
C
ISN 0399  1700 NLDI(4) = IVAL(1)
ISN 0400  NLD = IVAL(1)
ISN 0401  NTT = INTT
ISN 0402  STARTS(4) = VAL(1)
ISN 0403  NEWTRK = NEWTRK + 2

C READ THE LOAD INTERVALS AND THE NUMBER OF EACH TRUCK TYPE WHOSE GVW FALLS
C WITHIN EACH INTERVAL
C
ISN 0404  DO 1710 L=1,NLD
ISN 0405  READ (LI,1410) ELDINT, (VG(L,J),J=1,NTT)
ISN 0406  WRITE (LO,1415) ELDINT, (VG(L,J),J=1,NTT)
ISN 0407  VG(L,11) = ELDINT
ISN 0408  1710 CONTINUE
GO TO 2

*** EMPTY VEHICLE WEIGHT SECTION ***

1800 NLDI(5) = IVAL(1)
ISN 0411 NLD = IVAL(1)
ISN 0412 NTT = INTT
ISN 0413 STARTS(5) = VAL(1)
ISN 0414 NEWTRK = NEWTRK + 2
C
READ THE LOAD INTERVALS AND THE NUMBER OF EACH TRUCK TYPE WHOSE EVW FALLS
WITHIN EACH INTERVAL
C
DO 1810 L=1,NLD
ISN 0415 READ (LI,1410) ELDINT, (VE(L,J),J=1,NTT)
ISN 0416 WRITE (LO,1415) ELDINT, (VE(L,J),J=1,NTT)
ISN 0417 VE(L,11) = ELDINT
ISN 0418 1810 CONTINUE
ISN 0419 GO TO 2
C
*** STEERING AXLES SECTION ***

1900 NLDI(6) = IVAL(1)
ISN 0421 NLD = IVAL(1)
ISN 0422 NTT = INTT
ISN 0423 STARTS(6) = VAL(1)
ISN 0424 I DST = 6
ISN 0425 NEWTRK = NEWTRK + 2
C
READ THE LOAD INTERVALS AND, FOR EACH TRUCK TYPE, THE NUMBER OF
STEERING AXLES FOR EACH INTERVAL
C
DO 1910 L=1,NLD
ISN 0427 READ (LI,1410) ELDINT, (ST(L,J),J=1,NTT)
ISN 0428 WRITE (LO,1415) ELDINT, (ST(L,J),J=1,NTT)
ISN 0429 ST(L,11) = ELDINT
ISN 0430 1910 CONTINUE
ISN 0431 GO TO 2
C
*** OUTPUT KEYWORD SECTION ***

2000 IOUT = IVAL(1)
ISN 0433 GO TO 2
C
*** KEYWORD ERROR PROCESSING SECTION ***

9989 WRITE (LO,9089) IPFLG
ISN 0435 9089 FORMAT(/1X,19H*** ERROR IN LAYER ,I1,4H ***/
1 38H ACP NOT PERMITTED FOR RIGID PAVEMENT /
2 30H UNLESS ABOVE JCP OR CRC LAYER/ /
3 15H RUN TERMINATED)
ISN 0436 GO TO 9999
ISN 0437 9992 IGO = 3
ISN 0438 GO TO 99999
ISN 0439 9993 WRITE (LO,9093)
ISN 0440 9093 FORMAT(/1X,37H*** UNRECOGNIZABLE MATERIALS CODE ***/
1 15H RUN TERMINATED)
ISN 0441 GO TO 9999
ISN 0442
**OPTIONS IN EFFECT**

*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE) *

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLG(I) 

*STATISTICS* SOURCE STATEMENTS = 467, PROGRAM SIZE = 10756, SUBPROGRAM NAME = INPUT 

*STATISTICS* NO DIAGNOSTICS GENERATED 

***** END OF COMPILATION ***** 836K BYTES OF CORE NOT USED
SUBROUTINE INPRNT

COMMON /TEMPC/ CONTP(25), DISTCT
COMMON /EALPAY/ EALPT(10, 2), APPT(10, 2), EALFCT(20), IEQTRP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /FUNDS/ APOF(20, 2), RTINT, RTINF
COMMON /INTVLS/ STARTS(6)
COMMON /IO/ LI, LO, LD
COMMON /LABELS/ MATLAB(5, 10)
COMMON /LDS/ PGWV, PSAL, PTAL, PFGW, FSAL, FTAL, PTRL, PSTM(10), FSTM(10)
COMMON /LMP/ XLM(30), YLM(30), POTLM(20, 2), OUTP(20, 2), TOTALM
COMMON /LMP/ PPF, PTF, NASL, NSRL, TOVL(30, 2), XLMM(30)
COMMON /MNTPAR/ UNTCST(4), USERS(31, 4), WDTH, S, DISS, DCON, DIN, MFLG
COMMON /NEWSYS/ NEWSYS
COMMON /NMBR/ SA(30, 11), TA(30, 11), TR(50, 11), VE(30, 11)
COMMON /OUTSWH/ IOUT
COMMON /OVLAY/ XHCIO, XHCIM, WLANE, WPW, WPSH, PPW, CAC, CCR
COMMON /PSI/ PF, PICON, PTETM, PIPO, PTOV
COMMON /STRCDR/ STRCD(8), CC(4), MC(11) NC, STRC(5), RFS(4)
COMMON /STRML/ RFB(4)
COMMON /STRUC/ SN, SS, R, D, AGG, XJ, XE
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /TITLE/ TITLE(20, 3), TCTTL(20)
COMMON /TRYP/ TYP(2, 10), PTYP(10, 20, 2), PCTTR(20, 2), PERCT(4, 1)
COMMON /TITLE/ TITLE(20), SECTTL(20)
COMMON /TRYP/ TYP(2, 10), PTYP(10, 20, 2), PCTTR(20, 2), PERCT(4, 1)
COMMON /STTUS/ STRUC(8), CC(4), MC(11), NC, STRC(5), RFS(4)
COMMON /SWTCHS/ OVLIFE, PIINT, PIINF, TFPFC, PFNDPC, AGR, SPJCJT
COMMON /SWTCHS/ OVLIFE, PIINT, PIINF, TFPFC, PFNDPC, AGR, SPJCJT
RTINT = PCTINT * 0.01
RTINF = PCTINF * 0.01
TPF = TFPFC * 0.01
PFND = PFNDPC * 0.01
AGF = AGR * 0.01
CAC = CACI
SG = CGRI
IF (ICAC .EQ. 1) GO TO 4000
IF (ICAC .EQ. 2) GO TO 4010
IF (ICAC .EQ. 3) GO TO 4012
IF (ICAC .EQ. 4) GO TO 4014
GO TO 4016
99999 RETURN
*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 48, PROGRAM SIZE = 676, SUBPROGRAM NAME = INPRNT

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

920K BYTES OF CORE NOT USED
SUBROUTINE INIT (IGO)

COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IR, IC

COMMON /STRUC/ SN, SS, R, AGG, XJ, XK, E

COMMON /STRCOE/ STRCD(8), CC(4), MC(11), NC, STRC(5), RFS(4), RFB(4)

COMMON /TIME/ ATP, OVLIF, NYAP, NYR, YR(40)

DATA ICON, F /2.1.1/ C ICON IS THE INDEX ON CONDITION FACTOR USED TO RELATE AN OLD PCC PAVEMENT WITH AN AC OVERLAY TO AN EQUIVALENT SLAB THICKNESS. C F IS A FACTOR ALSO USED IN THE ABOVE RELATION.

GO TO (100, 200, 300), IGO

C HERE FOR PROGRAM INITIALIZATION, FIRST EXECUTION.

100 DD 110 J=1,NYR

CONTINUE

GO TO 900

C HERE FOR SET UP CHORES AFTER READING INPUT DATA.

200 CONTINUE

C WE HAVE ALL THE INPUT FOR A REPRESENTATIVE SECTION. DETERMINE -SN- OR -0- FOR COMPOSITE PAVTS, AS WELL AS SET UP STRUCTURAL COEF.

IF (IP .EQ. IR .OR. IP .EQ. IC) GO TO 230

SN = 0.

DO 215 L=1,NLAY

M = MTYPE(L)

IF (STRC(L) .NE. 0.) STRCD(M) = STRC(L)

IF (STRC(L) .EQ. 0.) STRC(L) = STRCD(M)

215 SN = SN + STRC(L)*THICK(L)

C SET -A- VALUE FOR OVERLAY = -A- FOR AC IF NOT READ IN SEPARATELY.

C IF (STRC(5) .EQ. 0.) STRC(5) = STRCD(1)

250 CONTINUE

GO TO 900

300 CONTINUE

RETURN

END
*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NDMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 42, PROGRAM SIZE = 992, SUBPROGRAM NAME = INIT

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION ***** 924K BYTES OF CORE NOT USED
SUBROUTINE DISTR (P, NHIST, NSLICE)

COMMON /SWTCHS/ OVLIFE, PCTINT, PCTINF, TPFPC, PFPONE, AGR, SPDCT, XMLI, CACI, CACR, ICAC, ACDEN, ICGR, GRDEN.

COMMON /BURKE/ XLAB, GAMMA, TFBAP

COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC

DIMENSION P(25)

IF (IP.EQ. IF) GO TO 100

C->FIX SURVIVAL CURVES FOR RIGID PAVEMENTS

P(1)=0
P(2)=0.
P(3)=0.
P(4)=0.0125
P(5)=.0125
P(6)=0.0295
P(7)=P(G)
P(8)=P(7)
P(9)=.03
P(10)=P(9)
P(11)=.085
P(12)=P(11)
P(13)=.0325
P(14)=P(13)
P(15)=.0595
P(16)=P(15)
P(17)=.0325
P(18)=.085
P(19)=.0325
P(20)=.03
P(21)=.03
P(22)=.0295
P(23)=P(22)
P(24)=P(23)
P(25)=.025

GO TO 999

100 CONTINUE

C->GET INITIAL TRAFFIC

AGF=AGR/100.
WO=TFBAP*(1+AGF)**(-NSLICE)

C->GET P(I) FOR I=1 TO NHIST

ACUM=0
ACPLRY=0
DO 10 I=1,NHIST

C-------->TRANSFORM YEARS INTO ACCUMULATED LOADS AT AGE I

ACUM=ACUM+WO*(1+AGF)**(-NSLICE)

C-------->GET CUMMULATIVE FRACTION OF PAVEMENTS THAT FAILED

ACUMIL = ACUM/1000000
POWER = -(XLAMB*ACUMIL)**GAMMA
IF (POWER .GT. -5.4E-79) POWER = -5.4E-79
ACPNOW=1-EXP(POWER)

C-------->GET FRACTION OF PAVEMENTS THAT FAILED DURING YEAR I

P(I)=ACPNOW-ACPLRY

GOTO 100
C-------->UPDATE POINTER AND DO IT AGAIN

ACPLYR=ACPNOW

ISN 0048
ISN 0049 10 CONTINUE
ISN 0050 999 RETURN
ISN 0051 END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 50, PROGRAM SIZE = 1066, SUBPROGRAM NAME = DISTR

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ******

924K BYTES OF CORE NOT USED
SUBROUTINE EALGET
C THIS ROUTINE CALCULATES THE RATIO OF EAL PER UNIT TIME UNDER THE
C PROPOSED REGULATIONS TO THAT UNDER THE PRESENT REGULATIONS,
C SUBJECT TO THE RESTRAINT OF EQUAL PAYLOAD PER UNIT TIME (IEQTRP=0),
C OR TO THE RESTRAINT OF EQUAL NUMBER OF TRIPS (IEQTRP=1).
ISN 0003 COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
ISN 0004 COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
ISN 0005 COMMON /PSI/ PF, PICON, PTERM, PIVO, PTOV
ISN 0006 COMMON /STRUC/ SN, SS, R, D, AGG, XJ, XK, E
ISN 0007 COMMON /TIME/ ATP, OVLIF, NVAP, NYAP, NVR, NR(40)
ISN 0008 COMMON /SWCHS/ DLIFE, PCTINT, PCTINF, TFPC, PFDPC, AGR, SPCJT,
  1 XMLI, CACI, CGRI, ICAC, ACDEN, ICGR, GREDN,
  2 INTT, SAVMT, IDST, NLD, MCODE(5), TFCDNS
ISN 0009 COMMON /TRTYP/ TYP(2,10), TYP(10,20,2), PCTTR(20,2), PERCT(4),
  1 NAXLES(10,4), NT(4), NTTY, NATT, NTT, NEWTRK
ISN 0010 COMMON /BURKE/ XLAMB, GAMMA, TFBAP
ISN 0011 DIMENSION S1(10), S2(10), T1(10), T2(10)
ISN 0012 IPVT = IP
ISN 0013 IF (IP .EQ. IC) IPVT = IR
ISN 0014 C CALL -TRAFFIC- ONLY IF NEW LIMITS OR WEIGHT DISTRIBUTIONS HAVE BEEN
ISN 0015 C READ FOR THIS PROBLEM
ISN 0016 IF (NEWTRK .GT. 1) CALL TRAFFIC
ISN 0017 CALL EAL18 (SN, D, PTERM, IPV)
ISN 0018 C EAL18 RETURNS 18K EAL PER AVERAGE TRUCK, EALPT, AND PAYLOAD PER
ISN 0019 C AVERAGE TRUCK, APPT, FOR EACH TRUCK TYPE.
ISN 0020 C FOR EACH YEAR OBTAIN THE (NORMALIZED) TOTAL PAYLOAD AND TOTAL 18K
ISN 0021 C EAL
ISN 0022 DO 10 J=1,NYAP
ISN 0023 CALL MULT (PTTYP(1,J,1), APPT(1,1), NTTY, S1)
ISN 0024 CALL MULT (PTTYP(1,J,1), EALPT(1,1), NTTY, T1)
ISN 0025 CALL SUM (S1, NTTY, SUM1)
ISN 0026 CALL SUM (T1, NTTY, TUM1)
ISN 0027 IF (J .EQ. 1) TFBA=TUM1*TFCDNS
ISN 0028 10 CONTINUE
ISN 0029 RETURN
ISN 0030 END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*STATISTICS* SOUR

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ****** 924K BYTES OF CORE NOT USED
SUBROUTINE OVCOST (THOV, OVCST)
OBTAINS COST/(LANE MILE) FOR GIVEN OVERLAY THICKNESS

COMMON /OVRLAY/ XHCIO,XHCIM,WLANE, WPSH, WGSH, PPVDSH, CAC, CGR
1 , CSCOAT

DATA C1/16.29629£3/
C COSTS ARE INPUT TO THIS ROUTINE IN DOLLARS/CU YD.
C C1 IS THE NUMBER OF CUBIC YDS IN A LAYER 1 MILE BY 1 FOOT BY 1 IN.
C
F = PPVDSH/100.
TH = THOV
C FIND THE VOLUME/(LANE MILE) OF ROAD OVERLAY, OF PAVED SHOULDER
C OVERLAY, AND OF GRANULAR SHOULDER OVERLAY

VPO = WLANE*TH*C1
VPSO = WPSH*TH*C1
VGSO = WGSH*TH*C1
C PAVEMENT OVERLAY COST
PVTOC = VPO*CAC
C UNPAVED SHOULDER OVERLAY COST
UPSHOC = CGR*(1.-F)*VGSO
C PAVED SHOULDER COST
PSHOC = CAC*F*VPSO
C TOTAL OVERLAY COST
OVCST = PVTOC + UPSHOC + PSHOC
RETURN
END
SUBROUTINE ACCTFC (TFC1, AGF, NYR, TFCA)

C CUMULATIVE TRAFFIC BY YEAR FROM BASE YEAR (18 KIP EAL).
C INPUT
C TFC1 - 18KIP EAL IN BASE YEAR (YEAR 1)
C AGF - ANNUAL GROWTH FACTOR (PERCENT/100.)
C NYR - NUMBER OF YEARS FOR WHICH ACCUMULATED TRAFFIC DESIRED.
C OUTPUT
C TFCA - ARRAY OF CUMULATIVE 18 KIP EAL THROUGH END OF INDEX YEAR.

DIMENSION TFCA (NYR)
TFC1 = TFC1
T = TFC1
DO 10 I=2,NYR
T = T*(1. + AGF)
TFCA(I) = TFCA(I-1) + T
10 CONTINUE
RETURN
END
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)  
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

ISN 0002   SUBROUTINE OUTPUT (LOCSW)
ISN 0003       RETURN
ISN 0004       END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)
*STATISTICS*  SOURCE STATMENTS = 3, PROGRAM SIZE = 176, SUBPROGRAM NAME =OUTPUT
*STATISTICS*  NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE OVTHKF (XNOV, THOV, VR)
REAL*8 THICK1(5), DMDRU, DMDRE
COMMON /MECH/XKT, NRU, NLH, ND, NDEL, IACR, IYR, JYR, CONSTR(20)
COMMON/HOR/A(10), B(10), C(10), DT(10), DF(10), S(10), T(10), TR(5), PI(5)
  * , PT(5), AC(5), AA, SCT(5), XMNW18(10), XKTO
COMMON /EXTRA/ PTOVTK, TPE, PF0, XMNOTK, XMOTK, NIS
COMMON /EXPT/ NPT, THICK(4), MTYPE(4), NLAV, IP, IF, IR, IC
COMMON /PSI/PFO, PICON, PTERM, PIOV, PTOV
COMMON /STRCOE/ STRCD(8), CC(4), MC(11), NC, STRC(5), RFS(4), RFB(4)
DIMENSION BET(5,2,2), CO(5,2,2)

BETA(1,1,1)=-1.5287
BETA(1,1,2)=-1.5387
BETA(3,1,1)=-1.4370
BETA(3,1,2)=-1.4370
BETA(4,1,1)=-1.5605
BETA(4,1,2)=-1.5776
BETA(1,2,1)=-1.53
BETA(1,2,2)=-1.562
BETA(3,2,1)=-1.4649
BETA(3,2,2)=-1.4649
BETA(4,2,1)=-1.5700
BETA(4,2,2)=-1.6085
CO(3,1,1)=600.
CO(3,1,2)=600.
CO(1,1,1)=10000.0
CO(1,1,2)=50000.0
CO(3,2,1)=1000.
CO(3,2,2)=1000.0
CO(1,2,1)=10000.0
CO(1,2,2)=10000.0
CO(4,2,1)=10000.0
CO(4,2,2)=10000.0
NLAY1=NLAY+1

DO 10 K=2, NLAY1
THICK1(K)=THICK1(K-1)
THICK1(1)=XMNOTK
IF (PF.GT.PTERM OR TPE.EQ.0) GO TO 100
TNPT=NPT
NPT=TNPT
IF (PFO.GE.PTOV) GOTO 3
GO TO 8
100 IF (PF0.GE.PTOV) GOTO 3
GO TO 8
ISN 0059  1 THICK1(1)=THICK1(1)+.25
ISN 0060  2 THOV=THICK1(1)
ISN 0061    GO TO 4
ISN 0062  3 THOV=XMNOTK
ISN 0063  4 RETURN
ISN 0064    END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*   SOURCE STATEMENTS =  63, PROGRAM SIZE =  1704, SUBPROGRAM NAME =OVTHKF

*STATISTICS*   NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

920K BYTES OF CORE NOT USED
FUNCTION RWT18L(D, PI, PT)
        AASHO-RIGID PREDICTION OF 18 KIP EAL TO TERMINAL PSI
        GT = ALOG10((PI-PT)/(PI-1.5))
        GTERM = GT/(1.+1.624E7/(D+1.)**8.46)
        RWT18L= 7.35*ALOG10(D+1.)-0.06+GTERM
        RETURN
        END
FUNCTION RNAASH(DA)

MODIFY AASHO-RIGID PREDICTION FOR NON-AASHO CONDITIONS

COMMON /STRUC/ SN, SS, R, D, AGG, XJ, XK, E

Z = E/XK

IKK = AGG

IF( IKK .EQ. 0 ) CT=204.16

D75 = DA**.75

RNAASH = ALOG10((CT/215.63)*(D75-1.132)/

1(D75-18.42/Z**0.25))

RETURN

END

***** END OF COMPILATION *****
FUNCTION GPSIR (XN, PI, D)
C
AASHO-RIGID PREDICTION OF PSI AFTER GIVEN 18 KIP EAL

DATA MAX, TEST /10, .001 /

EXP10(X) = EXP(2.302585*X)

PTN = 3.

ITER = 0

RN = RNAASH(D)

XNL = ALOG10(XN)

DT1 = 7.35*ALOG10(D+1.) - 0.06

DT2 = 1. + 1.624E7/(D+1.)**8.46

ITER = ITER + 1

IF (ITER .GT. MAX) GO TO 30

PT = PTN

GT = (XNL - DT1 - (4.22 - 0.32*PT)*RN)*DT2

PTN = PI - (PI - 1.5)*EXP10(GT)

IF (ABS(PTN - PT) .LT. TEST) GO TO 20

GO TO 10

20 GPSIR = PTN
RETURN

30 GPSIR = PTN
WRITE (6,1) MAX, PTN, PT, XN

1FORMAT ('FUNCTION GPSIR DID NOT CONVERGE AFTER, ', I5, 
1'11H ITERATIONS / 1X,3HLAST AND PREVIOUS PSI VALUES WERE, 
22F10.6 / 1X, 3HFOR, F10.0,26H 18KIP EAL TO DATE. ABORT.)

STOP
END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 25, PROGRAM SIZE = 944, SUBPROGRAM NAME = GPSIR

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

924K BYTES OF CORE NOT USED
SUBROUTINE GETD (W18, PI, PT, DB, DF)

AASHO-RIGID SLAB THICKNESS FOR GIVEN LIFE (18 KIP EAL) AND INITIAL
AND TERMINAL PSI

C EXP10(X) = EXP(2.302585*X)
CTR = 0
CT = CT + 1
IF (CT .GT. MAX) GO TO 99
O = O + W
W = RWT18L(D,PI,PT) + (4.22-.32*PT)*RNAASH(D)

DTERM = 7.35*ALOG10(D + 1.)

D1NLLOG = (W18 - (W - DTERM))/7.35

DN = EXP10(D1NLLOG) - 1.

IF (ABS(D-DN) .LT. TEST) GO TO 20
GO TO 10

20 OF = O
RETURN

99 OF 0
WRITE (6,1) O, ON
RETURN

1 FORMAT (1X, 27HToo MANY ITERATIONS IN GETD /
1 1X, 20HLAST TWO VALUES WERE , 2F8.4 /
2 1X, 36HINPUT LOG N18, PI, PT, STARTING D = /
3 1X, 4F10.4 /)

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 23, PROGRAM SIZE = 870, SUBPROGRAM NAME = GETD

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

924K BYTES OF CORE NOT USED
SUBROUTINE OVTHKR (D, EXD, TH)

OBTAIN THICKNESS OF AC OVERLAY TO BRING EQUIVALENT SLAB
THICKNESS, D, OF COMBINATION UP TO NEW DESIGN VALUE.
(EXISTING D DISCOUNTED FOR USE)

COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /PSI/ PF, PICON, PTERM, PIOV, PTGV
COMMON /STRCOE/

STRCD(8), CC(4), MC(11), NC, STRC(S), RFS(4), RFB(4)
DATA F/1.0/

INDX = 7.5 - 2.*PTERM
INDX = MINO(4, MAXO(1, INDX))
C = CC(INDX)
TH = 2.5*(F*D - C*EXD)
RETURN
END
SUBROUTINE TRAFFIC

THIS ROUTINE COMPUTES THE FOLLOWING

1. THE ADJUSTED AVERAGE EMPTY WEIGHT OF VEHICLES WEIGHED EMPTY
2. ADJUSTED GROSS WEIGHT AND TOTAL PAYLOAD CARRIED - PRESENT
   AND PROPOSED REGULATIONS
3. DISTRIBUTION OF AXLE WEIGHTS - PRESENT AND PROPOSED REGS.
4. AXLE WEIGHT DISTRIBUTIONS BY VEHICLE CLASSIFICATION -
   PROPOSED REGULATIONS

THE INPUTS ARE

1. NAXLES(10,4) - THE NUMBER OF SINGLE, TANDEM, TRIPLE AND
   STEERING AXLES FOR EACH TRUCK TYPE
2. NTTY - NUMBER OF TRUCK TYPES TO BE CONSIDERED (EXISTING)
3. NATT - NUMBER OF ADDED TRUCK TYPES (FUTURE DESIGN)
4. NEWTRK - SHIFTING INDICATOR
   0 - SHIFTING PROCEDURE TO BE DONE
   1 - SHIFTING PROCEDURE NOT TO BE DONE (ALREADY DONE)
5. SA(30,11) - NUMBER OF SINGLE AXLES WEIGHED BY INTERVAL AND
   TRUCK TYPE
6. TA(30,11) - NUMBER OF TANDEM AXLES WEIGHED BY INTERVAL AND
   TRUCK TYPE
7. TR(50,11) - NUMBER OF TRIPLE AXLES WEIGHED BY INTERVAL AND
   TRUCK TYPE
8. ST(30,11) - NUMBER OF STEERING AXLES WEIGHED BY INTERVAL AND
   TRUCK TYPE
9. VE(30,11) - NUMBER OF VEHICLES WEIGHED EMPTY BY INTERVAL AND
   TRUCK TYPE
10. VG(75,11) - NUMBER OF VEHICLES WEIGHED GROSS BY INTERVAL AND
    TRUCK TYPE
11. NLDI(6) - NUMBER OF INTERVALS INPUT FOR EACH OF THE ABOVE SIX
    ARRAYS, WHERE,
    1 = SA  2 = TA  3 = TR  4 = VG  5 = VE  6 = ST
12. EMPTY(10) - PERCENT INCREASE IN AVERAGE EMPTY WEIGHT FOR EACH
    TRUCK TYPE
13. PGWVL - PRESENT GROSS VEHICLE WEIGHT LIMIT
14. PSAL - PRESENT SINGLE AXLE WEIGHT LIMIT
15. PTAL - PRESENT TANDEM AXLE WEIGHT LIMIT
16. PTRAL - PRESENT TRIPLE AXLE WEIGHT LIMIT
17. PSTAW(10) - PRESENT STEERING AXLE WEIGHT LIMIT BY TRUCK TYPE
18-22.
   FGVWL, FSAL, FTAL, FTRAL, FSTAW(10) - SAME AS 13 THROUGH 17
   EXCEPT THAT THESE ARE VALUES UNDER PROPOSED REGULATIONS
23. SIZE - STANDARD INTERVAL SIZE (2-KIPS)
24. AVG - AVERAGE VARIABLE (AVG = 100. GIVES AVERAGE VALUES
    PER 100 TRUCKS)
25. NAPOV - NUMBER OF SELECTED CUMULATIVE PERCENTAGES FOR THE
    DISTRIBUTION OF AXLE WEIGHTS - PROPOSED REGS. SECTION
26. PAPOV - PERCENTAGE INCREMENT CORRESPONDING TO NAPOV ABOVE

COMMON /TRAFFIC/ ELWV(75), APVWE(75), APWVG(75), SAAVP(75),
                  TAPV(75), TRAPV(75), STAPV(75), NGWV
                  NTYPE(4), NLAY, IP, IF, IR, IC
LEVEL 2.3.0 (JUNE 78) TRAFIC OS/360 FORTRAN H EXTENDED DATE 84.262/15.03.27 PAGE 2

ISN 0005 COMMON /TRTYP/ TYP(2,10), PTTYP(10,20,2), PCTR(20,2), PERCT(4),
1 NAXLES(10,4), NT(4), NTY, NTT, NTT, NEWTRK
ISN 0006 COMMON /NUMBR/ SA(30,11), TA(30,11), TR(50,11), VE(30,11),
1 VG(75,11), NLDI(6), EMPTY(10), ST(30,11)
ISN 0007 COMMON /LDS/ PSWL, PSAL, PTL, PTRAL, FWWL, FSAL, FTAL, FTRAL,
1 PSTAW(10), PFAW(10)
ISN 0008 COMMON /CSN/ NOV, NAPDOV, PAPOV, SIZE, AVRG
ISN 0009 COMMON /INDEX/ ITT
ISN 0010 COMMON /ID/ LI, LD, LD
ISN 0011 COMMON /OUTPS/ TDA(10,6,2)
ISN 0012 COMMON EW(75), EVW(75), ELVW(75), GLVW(75), VWE(75),
1 PVWE(75), TWFAV(75), TPFAV(75), TVWE(75),
2 APPV(75), PPV(75), FACT(5), SAI(75), TAI(75), TRI(75),
3 SAA(75), TAA(75), TRA(75), SLA(75), TLA(75),
4 TRL(75), APSA(75), APA(75), ATR(75), APH(75),
5 GPA(75), GWA(75), GWAF(75), SLAR(75), TRAR(75),
6 SAN(75), TAN(75), TRN(75), PSA(75), PTA(75),
7 P(75), SLAT(75), TL(75), TR(75), STA(75),
8 PST(75), STL(75), STLAR(75), STR(75), APST(75),
9 A
1 ST(75), STN(75), NLD(6)
ISN 0013 IF (NEWTRK .EQ. 1) GO TO 9999
ISN 0015 DD 6 K=1,1
ISN 0016 DD 4 J=1,6
ISN 0017 DD 2 I=1,10
ISN 0018 TD4() = 0.0
ISN 0019 2 CONTINUE
ISN 0020 4 CONTINUE
ISN 0021 6 CONTINUE
ISN 0022 DD 7 I=1,6
ISN 0023 8 NLD(1) = NLD(1)
ISN 0024 7 CONTINUE
ISN 0025 DD 160 IT=1,NTT
ISN 0026 PERT = PERCT(NTT)
ISN 0027 ITT = IT
ISN 0028 VTN = 0.
ISN 0029 NSA = 0
ISN 0030 NTA = 0
ISN 0031 NTR = 0
ISN 0032 NNA = 0
ISN 0033 NNT = 0
ISN 0034 NNR = 0
ISN 0035 APV = 0.
ISN 0036 PAPV = 0.
ISN 0037 DD 8 I=1,75
ISN 0038 PSA(I) = 0.
ISN 0039 PTA(I) = 0.
ISN 0040 PTR(I) = 0.
ISN 0041 PST(I) = 0.
ISN 0042 SAI(I) = 0.
ISN 0043 TAI(I) = 0.
ISN 0044 TRI(I) = 0.
ISN 0045 STI(I) = 0.
ISN 0046 SAN(I) = 0.
ISN 0047 TAN(I) = 0.
ISN 0048 TRN(I) = 0.
ISN 0049 SNV(I) = 0.
ISN 0050 ELVWI(I) = 0.
ISN 0051 APVWE(I) = 0.
LEVEL 2.3.0 (JUNE 78) TRAFIC 05/360 FORTRAN H EXTENDED

00 ISN 0052 APWVG(I) = 0.
00 ISN 0053 SAAVPV(I) = 0.
00 ISN 0054 TAPAPV(I) = 0.
00 ISN 0055 TRAPPV(I) = 0.
00 ISN 0056 STAPPV(I) = 0.
00 ISN 0057 FACT(I) = 0.
00 ISN 0058 GLWNNI(I) = 0.
00 ISN 0059 APSAI(I) = 0.
00 ISN 0060 APITA(I) = 0.
00 ISN 0061 APTR(I) = 0.
00 ISN 0062 APST(I) = 0.
00 ISN 0063 8 CONTINUE
00 ISN 0064 DO 9 I=1,6
00 ISN 0065 NLDD(I) = NLDISV(I)
00 ISN 0066 9 CONTINUE
00 ISN 0067 C 35 CONTINUE
00 ISN 0068 50 CONTINUE
00 ISN 0069 IF (NAXLES(IT,1) .EQ. 0) GO TO 64
00 ISN 0071 SINGLE AXLES
00 ISN 0072 NLDS = NLDD(1)
00 ISN 0073 CALL COUNT (SA(1,IT), NLDS)
00 ISN 0074 CALL INTVL (SA, SAI, NLDS, NSA, 1, 30, SAA, IT)
00 ISN 0075 CALL PCTAGE (SAA, NSA, PSA)
00 ISN 0076 CALL ACMLTE (PSA, NSA, APSA)
00 ISN 0077 NNA = NSA
00 ISN 0078 64 IF (NAXLES(IT,2) .EQ. 0) GO TO 66
00 ISN 0079 TANDEM AXLES
00 ISN 0080 NLDS = NLDD(2)
00 ISN 0081 CALL COUNT (TA(1,IT), NLDS)
00 ISN 0082 CALL INTVL (TA, TAI, NLDS, NTA, 2, 30, TAA, IT)
00 ISN 0083 CALL PCTAGE (TAA, NTA, PTA)
00 ISN 0084 CALL ACMLTE (PTA, NTA, APTA)
00 ISN 0085 NNT = NTA
00 ISN 0086 66 IF (NAXLES(IT,3) .EQ. 0) GO TO 68
00 ISN 0087 TRIPLE AXLES
00 ISN 0088 NLDS = NLDD(3)
00 ISN 0089 CALL COUNT (TR(1,IT), NLDS)
00 ISN 0090 CALL INTVL (TR, TRI, NLDS, NTR, 3, 50, TRA, IT)
00 ISN 0091 CALL PCTAGE (TRA, NTR, PTR)
00 ISN 0092 CALL ACMLTE (PTR, NTR, APTR)
00 ISN 0093 NNR = NTR
00 ISN 0094 68 IF ((NAXLES(IT,4) .EQ. 0) .OR. (IP .NE. IF)) GO TO 69
00 ISN 0095 STEERING AXLES
00 ISN 0096 NLDS = NLDD(6)
00 ISN 0097 CALL COUNT (ST(1,IT), NLDS)
00 ISN 0098 CALL INTVL (ST, STI, NLDS, NST, 6, 30, STA, IT)
00 ISN 0099 CALL PCTAGE (STA, NST, PST)
00 ISN 0100 CALL ACMLTE (PST, NST, APST)
69 IF (IT .GT. NITY) GO TO 146

NGVW = NJ

C *** DISTRIBUTION OF SINGLE/TANDEM/TRIDEM AXLE WEIGHTS - PROPOSED LIMITS **

C SET UP THE TABLE OF SELECTED CUMULATIVE PERCENTAGES DEFINING THE

C GROSS WEIGHT AND AXLE WEIGHT CURVES

P = 0.0

DO 70 I=1,NAPOV
APOV(I) = P
70 CONTINUE

C GO TO 150

DO 147 I=1,NSA

C SAAPV(I) = APSA(I)
ISN 0111

C SANOV(I) = PSA(I)
ISN 0112

C PSA(I) = 0.
ISN 0113

C CONTINUE
ISN 0114

147 CONTINUE

NNA = NSA

DO 148 I=1,NTA

C TAAPV(I) = APTA(I)
ISN 0115

C TANOV(I) = PTA(I)
ISN 0116

C PTA(I) = 0.
ISN 0117

C CONTINUE
ISN 0118

148 CONTINUE

NNT = NTA

DO 149 I=1,NTR

C TRAPV(I) = APTR(I)
ISN 0119

C TRNOV(I) = PTR(I)
ISN 0120

C PTR(I) = 0.
ISN 0121

C CONTINUE
ISN 0122

149 CONTINUE

NNS = NST

DO 152 I=1,NJ

C STAPV(I) = APST(I)
ISN 0123

C STNOV(I) = PST(I)
ISN 0124

C PST(I) = 0.
ISN 0125

C CONTINUE
ISN 0126

151 CONTINUE

NNS = NST

DO 152 I=1,NJ

C APVWG(I) = APVWE(I)
ISN 0127

C CONTINUE
ISN 0128

152 CONTINUE

NGVW = MAXO(NSA,NTA,NTR,NST,NJ)

C WRITE TO DISK FOR RECALL IN EQUIVALENT LOAD APPLICATIONS ROUTINE

C

ISN 0138

150 CALL OUTPUT (3)

WRITE (LD) NSA, NTA, NTR, NST, NNA, NNT, NNR, NNS,
1 (PSA(I),I=1,NNA), (PTA(I),I=1,NNT), (PTR(I),I=1,NNR),
2 (PST(I),I=1,NST), (SANOV(I),I=1,NSA),
3 (TANOV(I),I=1,NTA), (TRNOV(I),I=1,NTR),
4 (STNOV(I),I=1,NST), (SAI(I),I=1,NSA), (TAI(I),I=1,NTA),
5 (TRI(I),I=1,NTR). (STI(I),I=1,NST), VTN, APV, PAPV

ISN 0139

160 CONTINUE

ISN 0140

9999 RETURN

ISN 0141

END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
LEVEL 2.3.0 (JUNE 78) TRAFIC 05/360 FORTRAN H EXTENDED DATE 84.262/15.03.27 PAGE 5

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 141, PROGRAM SIZE = 3510, SUBPROGRAM NAME = TRAFIC

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

900K BYTES OF CORE NOT USED
SUBROUTINE EAL18 (STRNUM, SLBTHK, TPSI, IPVT)

THIS ROUTINE CALCULATES THE EQUIVALENT 18-KIP AXLE LOAD
APPLICATIONS FOR EACH VEHICLE USING INFORMATION WRITTEN ON DISK BY
SUBROUTINE TRAFIC

THE INPUTS ARE
1. STRNUM - STRUCTURAL NUMBER FOR A FLEXIBLE PAVEMENT
2. SLBTHK - SLAB THICKNESS FOR A RIGID PAVEMENT
3. TPSI --- TERMINAL PSI
4. IPVT --- PAVEMENT TYPE SWITCH
5. APPT(10,2) - AVERAGE PAYLOAD PER VEHICLE, PRESENT + PROPOSED
6. COFVCT(6) - A VECTOR WITH ZERO-ONE ELEMENTS THAT DEFINE THE
   PRESENT TRAFFIC COMBINATION BEING CONSIDERED

THE OUTPUT IS
EALPT(10,2) - 18-KIP EAL PER TRUCK - PRESENT AND PROPOSED REGS.

DIMENSION PSA(75), PTA(75), PTR(75), SANOV(75), TANOV(75),
1   TRNOV(75), EFS(75), EFTA(75), EFTTR(75), SAN18(75),
2   TAN18(75), TRN18(75), SPN18(75), DPN18(75), TPN18(75),
3   SAI(75), TAI(75), TRI(75), SAM(75), TAM(75), TRM(75),
4   PST(75), STNOV(75), EFST(75), STN18(75), STPN18(75),
5   STI(75), STM(75),

COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
COMMON /TRTYP/ TTYP(2,10), PTTYP(10,20,2), PCTTR(20,2), PERCT(4),
1   NAXLES(10,4), NT(4), NTTY, NATT, NTT, NEWTRK
COMMON /10/ LI, LO, LD
COMMON /PSI/ PF, PICON, PTERM, PIDV, PTDV
COMMON /COMBI/ ICOMB, NVC, COFVCT(6)
DATA PSI1, PK1, PSI2, PK2 /4.2, 2.7, 4.5, 3.0/
REWIND 1
DO 1000 IT=1,NTT
1000 READ (LD) NSA, NTA, NTR, NST, NNA, NNT, NNR, NNS,
1   (PSA(I),I=1,NNA), (PTA(I),I=1,NNT), (PTR(I),I=1,NNR),
2   (PST(I),I=1,NNS), (SANOV(I),I=1,NSA),
3   (TANOV(I),I=1,NTA), (SAN18(I),I=1,NSA),
4   (TAN18(I),I=1,NTA), (SAN18(I),I=1,NSA),
5   (TRNOV(I),I=1,NTR), (SANOV(I),I=1,NST), (SAI(I),I=1,NSA),
APPT(IT,1) = APV
IF (NAXLES(IT,1) .EQ. 0) GO TO 50

READ FROM DISK THE INFORMATION STORED BY SUBROUTINE TRAFIC

COMPUTE THE 18-KIP EAL FOR EACH AXLE TYPE

TSN18 = 0.
IF (NAXLES(IT,1) .EQ. 0) GO TO 50
CALL MIDPNT (SAI, NSA, SAM)
IF (IPVT .EQ. 2) GO TO 10
GT = ALOG10((PSI1 - TPSI) / PK1)
CALL FLEXEO (SAM, NSA, 1.0, STRNUM, GT, EFSA)
GO TO 20
GT = ALOG10((PSI2 - TPSI) / PK2)
CALL RIGEO (SAM, NSA, 1.0, SLBTHK, GT, EFSA)
CALL SUM (SAN18, NNA, TSN18)
50 CONTINUE
TDN18 = 0.
TYN18 = 0.
IF (NAXLES(IT,2) .EQ. 0) GO TO 100
TAN18 = 0.
TRN18 = 0.
IF (NAXLES(IT,3) .EQ. 0) GO TO 150
TRI = -1*PSI + 2*TPSI
IA = MAXO(1, MINO(4,IA))
CALL STEREO (IA, EFST, NST, STM)
CALL SUM (STN18, NNR, TSN18)
200 EALPT(IT,1) = (TSN18*FLOAT(NAXLES(IT,1)) + TDN18 *
2  TSTN18*FLOAT(NAXLES(IT,4)) * 0.01
ISN 0074  EALPT(IT,1) = EALPT(IT,1)*COFVCT(IT)
ISN 0075  1000 CONTINUE
ISN 0076  RETURN
ISN 0077  END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NDXREF ALC NOANSF NOTERM IBM FLAG(1)

*STATISTICS*  SOURCE STATEMENTS = 76, PROGRAM SIZE = 9882, SUBPROGRAM NAME = EAL18

*STATISTICS*  NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****  908K BYTES OF CORE NOT USED
SUBROUTINE R1GEQ (XL, NL, ST, D, GT, EQ)
DIMENSION XL(1), EQ(1)
D1 = D + 1.0
D1P = D1 ** 8.46
C = 3.28 * ALOG10(ST)
GTB18 = GT / (1.0 + 1.620E+7 / D1P)
STP = ST ** 3.52
CON = 5.908 + C - GTB18
D = 10 L=1,NL
B2 = 3.63 * (XL(L) + ST) ** 5.20
BX = 1.0 + B2 / (D1P * STP)
E = CON - 4.62 * ALOG10(XL(L) + ST) + GT / BX
10 EQ(L) = 10.0 ** (-E)
RETURN
END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE)
*OPTIONS IN EFFECT*SOURC EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 15. PROGRAM SIZE = 836. SUBPROGRAM NAME = R1GEQ

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
LEVEL 2.3.0 (JUNE 78)  

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

ISN 0002  SUBROUTINE FLEXEQ (XL, NL, ST, SN, GT, EQ)
ISN 0003  DIMENSION XL(1), EO(1)
ISN 0004  SNP = (SN + 1.0) ** 5.19
ISN 0005  GTB18 = GT / (0.40 + 1094.0 / SNP)
ISN 0006  B1 = SNP * ST ** 3.23
ISN 0007  CON = 6.125 + 4.33 * ALOG10(ST) - GTB18
ISN 0008  DD 20 L=1,NL
ISN 0009  B2 = 4.79 * ALOG10(XL(L) + ST)
ISN 0010  BX = 0.40 + 0.081 * (XL(L) + ST) ** 3.23 / B1
ISN 0011  E = CON - B2 + GT / BX
ISN 0012  20 EQ(L) = 10.0 ** (-E)
ISN 0013  RETURN
ISN 0014  END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*  SOURCE STATEMENTS = 13, PROGRAM SIZE = 810, SUBPROGRAM NAME = FLEXEQ

*STATISTICS*  NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****  928K BYTES OF CORE NOT USED
SUBROUTINE STEREQ (IEQ, SEQ, NEQ, EQM)

THIS ROUTINE COMPUTES STEERING AXLE EQUIVALENCY FACTORS

THE INPUTS ARE
1. EQM - ARRAY OF INTERVAL MIDPOINTS
2. NEQ - NUMBER OF MIDPOINTS IN EQM
3. IEQ - INDICATES WHICH COLUMN OF THE EQUIVALENCY FACTOR TABLE
   (BY PSI) IS TO BE USED

THE OUTPUT IS
SEQ - ARRAY OF STEERING AXLE EQUIVALENCIES

DIMENSION SEQ(1), EQM(1)
COMMON /STEER/ EQFACT(15,5), PTST(4)

EQFACT(J,1) CONTAINS THE LOAD VALUES (J).
EQFACT(J,K) CONTAINS THE EQUIVALENCY FOR LOAD J, TERM PSI PTST(K-1)

DO 30 I=1,NEQ
   IF (EQM(I) .LT. EQFACT(1,1)) GO TO 25
   DO 10 J=2,15
      IF (EQFACT(J,1).GE. EQM(I)) GO TO 20
      CONTINUE
   5EQ(I) = EQFACT(J,IEQ)
   20 K = J-1
   25 SEQ(I) = EQFACT(K,IEQ) + (EQM(I) - EQFACT(K,1)) * 
      & ((EQFACT(J,IEQ)-EQFACT(K,IEQ))/(EQFACT(J,1)-EQFACT(K,1))**2)
   30 CONTINUE
RETURN
END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*STATISTICS* SOURCE STATEMENTS = 18, PROGRAM SIZE = 852, SUBPROGRAM NAME = STEREQ

***** END OF COMPILATION *****
SUBROUTINE INTVL (A1, A2, N, N1, IS, NN, A3, NM)

THIS ROUTINE CONVERTS THE END-OF-INTERVAL KIP TABLES TO EVENLY DISTRIBUTED INTERVALS BASED ON THE VARIABLE *SIZE*.

THE INPUTS ARE
1. A1 - ARRAY OF END-OF-INTERVAL KIP VALUES
2. N - NUMBER OF VALUES IN A1
3. IS - ARRAY IDENTIFIER WHERE,
   IS=1 - SINGLE AXLE ARRAY
   IS=2 - TANDEM AXLE ARRAY
   IS=3 - TRIPLE AXLE ARRAY
   IS=4 - GROSS WEIGHT ARRAY
   IS=5 - EMPTY WEIGHT ARRAY
   IS=6 - STEERING AXLE ARRAY
4. NN - MAXIMUM ALLOWABLE ROW LENGTH OF A1
5. NM - INDICATES WHICH TRUCK TYPE IS CURRENTLY BEING CONSIDERED

THE OUTPUTS ARE
1. N1 - THE NEW LENGTH OF THE END-OF-INTERVAL KIP TABLE
2. A2 - THE NEW END-OF-INTERVAL KIP TABLE
3. A3 - THE NUMBER OF TRUCKS (OR AXLES) WEIGHED IN EACH INTERVAL

COMMON /INTVLS/ STARTS(6)
COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
DIMENSION A1(NN,11), A2(75), A3(75), ACC(75)
XMLOAD = A1(N,11)
A2(1) = SIZE

SET *S* TO THE LARGEST EVEN NUMBER GREATER THAN OR EQUAL TO THE FIRST END-OF-INTERVAL KIP VALUE

S = 0.
K = 0
5 IF (S .GE. STARTS(IS)) GO TO 7
S = S + SIZE
K = K+1
GO TO 5

SET UP THE EVENLY DISTRIBUTED END-OF-INTERVAL KIP TABLE AND ZERO ALL INTERVALS AT BEGINNING OF TABLE IN WHICH NO TRUCKS/AXLES WERE WEIGHED

I = 1
J = 1
10 IF (A2(I) .GE. XMLOAD) GO TO 20
I = I+1
A2(I) = A2(J) + SIZE
J = J+1
GO TO 10
N1 = I
DO 30 I=1,K
A3(I) = 0.
30 Continue
ISN 0026  J0 CONTINUE
ISN 0027   I = K+1
ISN 0028  CALL ACMLTE (A1(1,NM), N, ACC)
ISN 0029  CALL ITRP (A1(1,11), ACC, A2, I, N1, N, A3, I)
ISN 0030  RETURN
ISN 0031  END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NULL)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*  SOURCE STATEMENTS = 30, PROGRAM SIZE = 1236, SUBPROGRAM NAME = INTVL

*STATISTICS*  NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE ITRP (V1, V2, V3, LIS, NV, NL, V4, IV)

THE INPUTS ARE:
1. V1 -- ARRAY OF X1 VALUES
2. V2 -- ARRAY OF F2(X) VALUES
3. V3 -- ARRAY OF X-VALUES
4. LIS -- FIRST NON-ZERO VALUE IN V3
5. NV -- LAST VALUE IN V3
6. NL -- LAST VALUE IN V1
7. IV -- INTERPOLATION INDICATOR WHERE
   IV=1 -- VALUES ARE CUMULATIVE
   0 -- VALUES ARE NOT CUMULATIVE

THE OUTPUT IS:
V4 -- ARRAY OF INTERPOLATED RESULTS

DIMENSION V1(75), V2(75), V3(75), V4(75)
IF (LIS .EQ. 1) V4(1) = 0.0
J = 1
DD 50 I=LIS,NV
DD 10 K=J,NL

FIND THE SMALLEST X1 GREATER THAN OR EQUAL TO X
IF (V1(K) .GE. V3(I)) GO TO 20

SET X1 AND F1 VALUES APPROPRIATELY, THEN INTERPOLATE

20 J = K
L = K-1
IF (L .EQ. 0) GO TO 30
25 F1 = V2(L)
X1 = V1(L)
GO TO 40
30 X1 = 0.0
GO TO 40
35 F1 = V4(1)
40 V4(I) = F1 + (V3(I)-X1) * ((V2(K)-F1) / (V1(K)-X1))
IF (K .LE. NL) GO TO 50
V2(K) = V2SV
V1(K) = V1SV
GO TO 25

CONTINUE
C INTERVAL

ISN 0034  IF (IV .EQ. 0) GO TO 999
ISN 0036     J = NV
ISN 0037     DO 60 I=2,NV
ISN 0038     V4(J) = V4(J) - V4(J-1)
ISN 0039     J = J-1
ISN 0040     60 CONTINUE
ISN 0041     999 RETURN
ISN 0042     END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 41, PROGRAM SIZE = 1132, SUBPROGRAM NAME = ITRP

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE PCTAGE (Pi, NP, P2)
C
C
C
C
C
C
C
C
DIMENSION P1(75), P2(75)

TOT = 0.0

DO 10 I=1,NP

TOT = TOT + P1(I)

10 CONTINUE

DO 20 I = 1,NP

P2(I) = P1(I) / TOT * 100.0

20 CONTINUE

RETURN

END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 11, PROGRAM SIZE = 440, SUBPROGRAM NAME = PCTAGE

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE COUNT(CA. ICA)

THIS ROUTINE DETERMINES WHICH OF THE *ICA* VALUES IN ARRAY CA IS
THE LAST NON-ZERO VALUE

DIMENSION CA(75)
DD 10 I=1,ICA
IF (CA(I) .GT. 0.0) J = I
10 CONTINUE
ICA = J
RETURN
END

*STATISTICS* SOURCE STATEMENTS = 9, PROGRAM SIZE = 326, SUBPROGRAM NAME = COUNT
*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE ACMLTE (AIN, NA, AOUT)

C
C THIS ROUTINE CONVERTS ARRAY AIN TO A CUMULATIVE ARRAY
C
DIMENSION AIN(75), AOUT(75)

IN 0004  AOUT(I) = AIN(I)
IN 0005  NB = NA - 1
IN 0006  DO 10 I = 1, NB
IN 0007  J = I + 1
IN 0008  AOUT(J) = AOUT(I) + AIN(J)
10 CONTINUE
RETURN
END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 10, PROGRAM SIZE = 410, SUBPROGRAM NAME =ACMLTE

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE MIDPNT (Xi, NM, X2)
C
C THIS ROUTINE DETERMINES THE MIDPOINT OF EACH INTERVAL IN ARRAY X1,
C WHERE EACH VALUE IN X1 IS AN END-OF-INTERVAL KP VALUE
C
COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
DIMENSION X1(75), X2(75)
I = 0
J = 1
ELI = X1(NM)
X2(1) = X1(1) - (SIZE/2.)
10 I = I+1
J = J+1
X2(J) = X2(I) + SIZE
IF (X1(J) .LT. ELI) GO TO 10
RETURN
END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 14, PROGRAM SIZE = 448, SUBPROGRAM NAME =MIDPNT

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED
SUBROUTINE MULT (YA, YB, NU, YC)

C THIS ROUTINE MULTIPLIES TWO VECTORS SUCH THAT YC(I) = YA(I)*YB(I)

DIMENSION YA(75), YB(75), YC(75)

DO 10 I=1,NU

YC(I) = YA(I) * YB(I)

10 CONTINUE

RETURN

END
SUBROUTINE AVRGE (AV, NV, AN, AVG)

THIS ROUTINE COMPUTES THE AVERAGE OF THE VALUES IN ARRAY AV OVER *AN*

DIMENSION AV(75)

AVG = 0.0

DO 10 I=1,NV

AVG = AV(I) + AVG

10 CONTINUE

AVG = AVG / AN

RETURN

END
SUBROUTINE SUM (S1, NS, S2)

C
C  THIS ROUTINE COMPUTES THE SUM OF THE VALUES IN ARRAY S1
C

DIMENSION S1(75)

S2 = 0.0

DO 10 I = 1, NS
  S2 = S2 + S1(I)
10  CONTINUE

RETURN

END
SUBROUTINE ZERO (A,N)
DIMENSION A(N)
DO 10 I=1,N
10  A(I) = 0.
RETURN
END
SUBROUTINE PSIT(P, PF, W, XKT)
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
DOUBLE PRECISION PWR
COMMON/HOR/A(10), B(10), C(10), DT(10), S(10), T(10), TR(S), PI(S)
*PT(5), AC(5), AA, SC(S), XMNST(10), XKTO
COMMON /EXTRA/ PTVKT, PTE, PF0, XMDOKT, XMOKT, NIS
W = TR(1) * 0.1
P = PI(NPT)
GO TO (10, 20, 30, 40, 50). NPT
10 CONTINUE
XKT = 89.15 + .00367 * T(1) ** .99 * DT(1) ** (-2.83) * S(1) ** 2.1 * DF(2) ** .85
PF = 3.663 + 1236.1 * S(2) ** (-.086) * C(6) ** .3 * C(3) ** .12 * C(4) ** (-.21)
** C(2) ** (-.22) * DT(1) ** .25 * C(1) ** (-3.13) * C(5) ** .31
GO TO 60
11 CONTINUE
XKT = 92.83 + .27 * 10.0 ** (-12.0) * S(2) ** 1.64 * DF(1) ** (-.46) * C(1) ** 7.97
** DT(1) ** (-1.45) * C(5) ** (-3.38) * TR(NPT) ** (-.25) * T(1) ** 1.09
PF = 3.667 + 117.44 * S(2) ** (-.018) * DF(2) ** (-.034) * C(1) ** (-1.67) * C(4) **
*(.085) * DT(1) ** .49 * T(1) ** (.059) * C(5) ** .25
GO TO 60
12 CONTINUE
XKT = 91.51 + .6637 * DF(1) ** .23 * C(2) ** .38 * C(4) ** (-.18) * TR(NPT) ** (-.15)*
** T(1) ** 1.45
PF = 2.367 + 15.59 * S(3) ** (-.018) * C(1) ** (-.55) * C(3) ** (-.24) * S(1) **
*(-.17) * T(1) ** (*0.85) * TR(NPT) ** .03
GO TO 60
13 CONTINUE
XKT = 81.84 + .052 * DF(2) ** (-.32) * DF(1) ** 1.4 * C(2) ** .89 * T(1) ** .25*
** S(1) ** (-1.74)
PF = 3.719 + 3.327 * C(1) ** (-.38) * C(4) ** .033 * S(3) ** (-.09) * C(3) **
*(-.061) * S(4) ** .071 * S(2) ** .16 * S(1) ** (-.017) * T(1) ** (-.075)
GO TO 60
14 CONTINUE
XKT = -.0737 + 231.63 * T(1) ** 1.26 * S(4) ** 3 * TR(NPT) ** (-.47)
PF = .00804 + 7.6131 * DF(2) ** (-.15) * T(1) ** .021 * C(5) ** (-1.37)
GO TO 60
15 CONTINUE
PFO = PF
RETURN
END

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 31, PROGRAM SIZE = 3838, SUBPROGRAM NAME = PSIT
*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ******

912K BYTES OF CORE NOT USED
SUBROUTINE RUSIAN (DPTEMP, DMORU, NLAY, NPT, NRU, NLH)

IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 MTERM, NTERM
DIMENSION X(5), EX(5), DP(5), R(S), E(6), WHAT(5), DPTEMP(5)

IF(NPT.NE.3.0R.NRU.NE.1) GO TO 112
E(1)=65000.0
E(2)=12000.0
E(3)=5000.0

112 IF(NPT.NE.3.0R.NRU.NE.2) GO TO 120
E(1)=65000.0
E(2)=12000.0
E(3)=5000.0

120 IF(NPT.NE.1.0R.NRU.NE.1.0R.NLH.NE.1) GO TO 30
E(1)=300000.0
E(2)=80000.0
E(3)=12000.0
E(4)=6000.0

30 IF(NPT.NE.1.0R.NRU.NE.1.0R.NLH.NE.2) GO TO 40
E(1)=305000.0
E(2)=100000.0
E(3)=16000.0
E(4)=6000.0

40 IF(NPT.NE.1.0R.NRU.NE.2.0R.NLH.NE.1) GO TO 50
E(1)=325000.0
E(2)=130000.0
E(3)=90000.0
E(4)=16800.0
E(5)=6000.0

50 IF(NPT.NE.1.0R.NRU.NE.2.0R.NLH.NE.2) GO TO 60
E(1)=325000.0
E(2)=95000.0
E(3)=35000.0
E(4)=18500.0
E(5)=6000.0

60 IF(NPT.NE.4.0R.NRU.NE.1.0R.NLH.NE.1) GO TO 70
E(1)=325000.0
E(2)=130000.0
E(3)=90000.0
E(4)=16800.0
E(5)=6000.0

70 IF(NPT.NE.4.0R.NRU.NE.1.0R.NLH.NE.2) GO TO 80
E(1)=325000.0
E(2)=130000.0
E(3)=90000.0

END
IF (NPT .NE. 4.0 R. NRU .NE. 2.0 R. NLH .NE. 1) GO TO 90
E(1) = 325000.
E(2) = 130000.0
E(3) = 90000.0
E(4) = 38000.0
E(5) = 19000.0
E(6) = 6000.
GO TO 915
K = 1, 5
RTEMP = 10.**2. + (12*(K-1)**2.)
R(K) = DSQRT(RTEMP)
K = 1, NLAV
DP(K) = DPTEMP(K)
IPVMT = 3
NW = 5
LEQ = 0
NOL = NLAV
DO 5 K = 1, NL

5 X(K) = E(K)/1000000.

IF( DP( NOL-1 ) .LE. 10.0 ) LEQ = 1
NC = NOL
IF( LEQ .EQ. 1 ) NC = NOL - 1
IF( LEQ .EQ. 1 ) X(NOL-1) = X(NOL)

53 BTERM = 10.0 ** (-0.05071) * DP(1) ** 0.10148
ISN 0094
NTERM = 10.0 ** (-0.50233) * DP(1) ** 0.087879
ISN 0095
CTERM = 10.0 ** (-0.060039) * DP(1) ** 0.0095198
ISN 0096
MTERM = 0.704 - 0.026 * DP(1)
ISN 0097
HTERM = 10.0 ** 1.8631 * DP(1) ** (-0.0038499)
ISN 0098
TMB = 2.0 * MTERM * BTERM
ISN 0099
NL = NLAV
EI = X(NC)
N1 = NL - 1
ISN 012 SUM = 0.0
ISN 013 DO 10 I = 1, N1
ISN 014 10 SUM = SUM + DP(I)
ISN 015 HS = HTERM - SUM
ISN 016 DP( NL ) = HS
ISN 017 NT = NC - 1
ISN 018 DO 11 I = 1, NT
ISN 019 11 EX(I) = X(I)
ISN 020 EX(NL) = X(NC)
ISN 021 EXT = EX(NL) * 1000000.0
ISN 022 IF( LEQ .EQ. 0 ) GO TO 14
C
GO TO ( 12, 12, 13, 13, 12, 12 ), IPVMT
C
ISN 0115  GO TO 14
C
ISN 0116  12 EX(NL - 1) = EX(NL) * (1.0 + 7.18 * DLOG10( DP( NC)) - 1.56 *
( DLOG10( EXT ) - DLOG10( DP( NC)) ) )
C
ISN 0117  GO TO 14
C
ISN 0118  13 EX(NL - 1) = EX(NL) * (1.0 + 10.52 * DLOG10( DP( NC)) - 2.10 *
( DLOG10( EXT ) - DLOG10( DP( NC)) ) )
C
ISN 0119  14 CONTINUE
C
ISN 0120  HPR = 0.0
ISN 0121  DO 15 I = 1, NL
ISN 0122  XNUM = EX(I)/EI
ISN 0123  HPR = HPR + ( XNUM ** NTERM ) * DP( I)
ISN 0124  15 CONTINUE
C
ISN 0125  PHIALF = TMB * ((TMB + 1.0)/(TMB - 1.0)) ** 0.5
ISN 0126  ALPHA = PHIALF/HPR
ISN 0127  TTM = 2.0 * MTERM
C
ISN 0128  DO 20 I = 1, NW
ISN 0129  ARG = ALPHA * R(I)
C
ISN 0130  WHAT( I) = 0.47746 * (CTERM/( EI * 1000000.0)) * (1000.0/HPR) *
( TTM + 1.0) * BESJO( ARG )
C
ISN 0131  20 CONTINUE
ISN 0132  DMORU*WHAT(1)
ISN 0133  RETURN
ISN 0134  END

*OPTIONS IN EFFECT* NAME(Main) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NONMAP NFORMAT GSTMT NDXREF ALC NDANSF NOTERM IBM FLAG(I)

*STATISTICS*   SOURCE STATEMENTS =  133,  PROGRAM SIZE =  4166,  SUBPROGRAM NAME =RUSIAN

*STATISTICS*   NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****  908K BYTES OF CORE NOT USED
FUNCTION BESJO ( X )
IMPLICIT REAL*8 (A-H, O-Z)
A FUNCTION TO CALCULATE BESSEL FUNCTION JOX) USING POLYNOMIAL APPROXIMATION
REFERENCE HANDBOOK OF MATH. FUNCTIONS, BUREAU OF STANDARDS, PAGES 369-370

ASSIGN 2 TO JOJ1
CONTINUE
X3 = X/3.0
IF ( X.GT. 3.0)
X3 = X3*X3
X32 = X3*X3
X33 = X32*X3
X34 = X32*X32
X35 = X32*X33
X36 = X33'*X33
GO TO JOJ1(2,10)

IF (DABS(X) .LE. 3.3)
GO TO 3
Xi = X - 0.7853982 - 0.04166397*X3 - 0.3954E-04*X32 + 0.262573E-02*X33 + 0.54125E-03*X34 + 0.29333E-03*X35 + 0.13558E-03*X36
BESJO = 0.7978846 - 0.77E-6*X3 + 0.552740E-02*X32 + 0.9512E-04*X33 + 0.137237E-02*X34 - 0.72805E-03*X35 + 0.14476E-03*X36
+ IDSQRT(X) * DCOS(X1)
RETURN

X = 0.3163866 * X3 + 0.000210*X32 + 2.2499997*X34 + 1.2656208*X36 + 0.0444479*(X34*X34) - 0.0039444*(X35*X35) + 0.0444479*(X36*X36)
RETURN
END

* OPTIONS IN EFFECT * NAME (MAIN) NOOPTIMIZE LINECOUNT (60) SIZE (MAX) AUTODBL (NONE) SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG (I) * OPTIONS IN EFFECT * SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG (I) * STATISTICS * SOURCE STATEMENTS = 31, PROGRAM SIZE = 1934, SUBPROGRAM NAME = BESJO * STATISTICS * NO DIAGNOSTICS GENERATED
LEVEL 2.3.0 (JUNE 78)  OS/360 FORTRAN H EXTENDED  DATE 84.262/15.04.02  PAGE 2

****** END OF COMPILATION *****  924K BYTES OF CORE NOT USED
SUBROUTINE SURVIV (IACR)
COMMON /EXPVT/ NPT, THICK (4), MTYPE(4), NLAY, IP, IF, IR, IC
COMM/ /PSI/ PF, PICON, PTERM, PTOV, PTOV
COMMON /BURKE/ XLAM, GAMMA, TFBAP
DIMENSION XLAM (5,3), GAM(5,3)
C->THIS SUBROUTINE SETS SURVIVAL CURVE PARAMETER VALUES (XLAMB AND GAMMA)
C FOR FLEXIBLE PAVEMENTS
IF (IP .NE. IF) GO TO 999
C->FIND OUT TYPE OF FAILURE (PSI OR DISTRESS)
IF (PF .GE. PTERM) GO TO 200
C->SET XLAMB AND GAMMA FOR PSI SURVIVAL CURVES BY QUALITY STD. AND PAVT TYPE
XLAM (1,1)=0.276
GAM (1,1)=2.111
XLAM (1,2)=0.417
GAM (1,2)=1.549
XLAM (1,3)=0.607
GAM (1,3)=1.497
XLAM (3,1)=0.423
GAM (3,1)=1.363
XLAM (3,2)=0.687
GAM (3,2)=1.365
XLAM (3,3)=0.787
GAM (3,3)=1.012
XLAM (4,1)=0.327
GAM (4,1)=1.524
XLAM (4,2)=0.555
GAM (4,2)=1.163
XLAM (4,3)=0.818
GAM (4,3)=1.088
GO TO 300
200 CONTINUE
C->SET XLAMB AND GAMMA FOR DISTRESS SURV. CURVES BY QUAL. STD. AND PAVT. TYPE
XLAM (1,1)=0.004
GAM (1,1)=2.681
XLAM (1,2)=0.007
GAM (1,2)=4.380
XLAM (1,3)=0.009
GAM (1,3)=5.000
XLAM (3,1)=0.003
GAM (3,1)=2.129
XLAM (3,2)=0.006
GAM (3,2)=3.065
XLAM (3,3)=0.008
GAM (3,3)=4.023
XLAM (4,1)=0.004
GAM (4,1)=2.443
XLAM (4,2)=0.008
GAM (4,2)=3.382
XLAM (4,3)=0.011
GAM (4,3)=4.119
GO TO 300
300 XLAM=XLAM (NPT, IACR)
GAMMA=GAM(NPT, IACR)
999 RETURN
*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 51, PROGRAM SIZE = 918, SUBPROGRAM NAME = SURVIV

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ******

928K BYTES OF CORE NOT USED
SUBROUTINE COSCAL
COMMON /PSI/ PF, PICON, PTERM, PIDV, PTOV
COMMON /TITLE/ TITLE(20,3), SecTtl(20)
COMMON /MECH/ XKT, NRU, NLH, ND, NDEL, IACR, IYR, JYR, Constr(20)
COMMON /LMP/ XLM(30), YLM(30), Potlm(30,2), oUT(20,2)
  TOTALM, PPF, PTF, PFND, NASL, NSL, Tovlm(30,2), XLM(20)
COMMON /DRLAY/ XHCIO, XHCIM, WLNE, WPSH, WGHG, PPDVDSH, CAC, CGR, CSCODAT
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /MTNPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMMON /SWTCHS/ DLLIFE, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACI, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /COST/ COSTH(20), COSTRM(20), COSTPM(20)
COMMON /IO/ LI, LO, LD
COMMON /COMBI/ ICORB, NVC, CFOCFT(6)
COMMON /LMP/ PR(50), PR1(30,2), PR2(19,20), ZLM(30), YMILES(20),
  ZMILES(20), REHPLM(20)
COMMON /MNTPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMM/ON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /SWTCHS/ DLIILF, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACL, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /COST/ COSTH(20), COSTRM(20), COSTPM(20)
COMMON /IO/ LI, LO, LD
COMMON /COMBI/ ICORB, NVC, CFOCFT(6)
COMMON /LMP/ PR(50), PR1(30,2), PR2(19,20), ZLM(30), YMILES(20),
  ZMILES(20), REHPLM(20)
COMMON /MNTPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /SWTCHS/ DLIILF, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACL, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /COST/ COSTH(20), COSTRM(20), COSTPM(20)
COMMON /IO/ LI, LO, LD
COMMON /COMBI/ ICORB, NVC, CFOCFT(6)
COMMON /LMP/ PR(50), PR1(30,2), PR2(19,20), ZLM(30), YMILES(20),
  ZMILES(20), REHPLM(20)
COMMON /MNTPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /SWTCHS/ DLIILF, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACL, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /COST/ COSTH(20), COSTRM(20), COSTPM(20)
COMMON /IO/ LI, LO, LD
COMMON /COMBI/ ICORB, NVC, CFOCFT(6)
COMMON /LMP/ PR(50), PR1(30,2), PR2(19,20), ZLM(30), YMILES(20),
  ZMILES(20), REHPLM(20)
COMMON /MNTPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /SWTCHS/ DLIILF, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACL, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /COST/ COSTH(20), COSTRM(20), COSTPM(20)
COMMON /IO/ LI, LO, LD
COMMON /COMBI/ ICORB, NVC, CFOCFT(6)
COMMON /LMP/ PR(50), PR1(30,2), PR2(19,20), ZLM(30), YMILES(20),
  ZMILES(20), REHPLM(20)
COMMON /MNTPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /SWTCHS/ DLIILF, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACL, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /COST/ COSTH(20), COSTRM(20), COSTPM(20)
COMMON /IO/ LI, LO, LD
COMMON /COMBI/ ICORB, NVC, CFOCFT(6)
COMMON /LMP/ PR(50), PR1(30,2), PR2(19,20), ZLM(30), YMILES(20),
  ZMILES(20), REHPLM(20)
COMMON /MNTPAR/ UNTCT(4), USRM(31,3), WTH, S, DISS, DCON, DIN, MFLG
COMMON /TIME/ ATP, OVLIF, NYAP, NVR, YR(40)
COMMON /SWTCHS/ DLIILF, PCTINT, PCTINF, PFPCC, PFNDPC, AGR, SPCT, 1
  XMII, CACL, C OR, ICAC, ACENS, ICGR, GRDNS, 2
  INTT, SAVMT, IDST, NLD, MCDOE(5), TFCDNS
DO 20 I=2,NYAP
XMAIN1=0.
FAILML=0.
DO 30 K=1,NASL
FAILML=FAILML+PR1(K,I)*YLM(K)
ZLM(K)=ZLM(K)-YLM(K)*PR1(K,I)
XMAIN1=XMAIN1+ZLM(K)*EARMAR(K+I-1)
30 CONTINUE
DO 20 I=2,NYAP
XMAIN2=0.
IF (PF.GE.PTERM.AND.NDEL.GT.0.AND.I.GT.NDEL) GO TO 70
LL=I-1
DO 40 L=1,LL
FAILML=FAILML+YMILES(L)*PR2(L,I)
ZMILES(L)=ZMILES(L)-YMILES(L)*PR2(L,I)
XMAIN2=XMAIN2+ZMILES(L)*EARMAR(I-L)
40 CONTINUE
COSTRH(I)=(FAILML*REHPLM(I)**(1.+XHCIO)**I
YMILES(I)=CONSTR(I)+FAILML
ZMILES(I)=YMILES(I)
COSTRM(I)=(XMAIN1+XMAIN2)*(1.+XHCIM)**I
GO TO 20
70 DELTA=YMILES(I-NDEL)-CONSTR(I-NDEL)
YMILES(I-NDEL)=CONSTR(I-NDEL)
LL=I-1
DO 50 L=1,LL
FAILML=FAILML+YMILES(L)*PR2(L,I)
ZMILES(L)=ZMILES(L)-YMILES(L)*PR2(L,I)
XMAIN2=XMAIN2+ZMILES(L)*EARMAR(I-L)
50 CONTINUE
PROB=0.
ML=NDEL-1
DO 60 M=1,ML
PROB=PROB+PR2(I-NDEL,I-NDEL+M)
60 CONTINUE
YMILES(I)=CONSTR(I)+FAILML+DELTA*(1-PROB)
ZMILES(I)=YMILES(I)
COSTRH(I)=((FAILML+DELTA*(1-PROB))*REHPLM(I)**(1.+XHCIO)**I
COSTRM(I)=(XMAIN1+XMAIN2)*(1.+XHCIM)**I
GO TO 20
DO 100 I=1,20
COSTPM(I)=0.
100 CONTINUE
IF (IP.NE.IF.OR.JYR.EQ.0) GO TO 99
TINTML=0.
DO 110 K=1,NASL
TINTML=TINTML+YLM(K)
110 CONTINUE
TCNSTR=0.
DO 120 I=1,NYAP
TCNSTR=TCNSTR+CONSTR(I)
120 CONTINUE
COSTPM(I)=CSCOAT*(TINTML+(TCNSTR/2)/FLOAT(JYR)
COSTPM(I)=COSTPM(I)*(1.+XHCIM)**I
WRITE (LO,S13) (SECTTL(J),J=1,20)
613 FORMAT (//,20X,20A4,//)
ISN 0109 600 FORMAT (10X,5HYEAR,10X,10HRUT MAINT,10X,10H REHAB,10X, 
1 10H PREV MAINT.,/25X,10H COST,10X,10H COST,10X, 
2 10H COST ,/)
ISN 0110 WRITE (L0,601) (I,COSTRM(I),COSTRH(I),COSTPM(I),I=1,NYAP)
ISN 0111 601 FORMAT ((10X,I5,3(10X,E10.3)))
ISN 0112 WRITE (L0,620)
ISN 0113 620 FORMAT (1H1)
ISN 0114 99 RETURN
ISN 0115  END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 114, PROGRAM SIZE = 8812, SUBPROGRAM NAME =COSCAL

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION *****

900K BYTES OF CORE NOT USED
SUBROUTINE RHBLT(REHPLM)

COMMON /MISC/ IPOT, IARMS, OLMNT, AGF
COMMON /BURKE/ XLAMB, GAMMA, TFBAP
COMMON /TIME/ ATP, OVLIF, NYAP, YR(40)
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /PSI/ PF, PICON, PTERM, PI0V, PTOV
COMMON /ID/ LI, LO, LD
COMMON /STRUC/ SN, SS, R, D, AGG, XJ, XK, E
COMMON /TIME/ ATP, OVLIF, NYAP, YR(40)
COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
COMMON /PSI/ PF, PICON, PTERM, PI0V, PTOV

ISN 0002
ISN 0003
ISN 0004
ISN 0005
ISN 0006
ISN 0007
ISN 0008
ISN 0009
ISN 0010
ISN 0011
ISN 0012
DO 10 I=1,NYAP

ISN 0013
ISN 0014
ISN 0016
ISN 0017
ISN 0019
ISN 0020
ISN 0021
ISN 0022
ISN 0023
ISN 0024
ISN 0025
ISN 0026
ISN 0027
ISN 0028
ISN 0029
ISN 0030

DO 10 I=1,NYAP

IF (IP. NE. IF) GO TO 20

CALL OVTHKR(DOV,D,THOV)

CALL OVCOST(THOV,REHPLM(I))

10 CONTINUE

GO TO 99

99 RETURN

END

*OPTIONS IN EFFECT* NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 29, PROGRAM SIZE = 970, SUBPROGRAM NAME = RHBLT

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

928K BYTES OF CORE NOT USED

*STATISTICS* NO DIAGNOSTICS THIS STEP
APPENDIX 2

BASIC INPUT FOR THE MODIFIED RENU2 PROGRAM
### SAMPLE SECTION DATA

<table>
<thead>
<tr>
<th>Item</th>
<th>Section 1</th>
<th>Section 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of vehicle classes</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Analysis Period</td>
<td>18 yrs.</td>
<td>18 yrs.</td>
</tr>
<tr>
<td>Annual Growth Rate in 18-kip ESAL per year</td>
<td>3.35 %</td>
<td>3.35 %</td>
</tr>
<tr>
<td>Interest Rate</td>
<td>4.0 %</td>
<td>4.0 %</td>
</tr>
<tr>
<td>Average No. of Vehicles per Year</td>
<td>320,000</td>
<td>320,000</td>
</tr>
<tr>
<td>Type of Pavement</td>
<td>Overlaid</td>
<td>Hot Mix</td>
</tr>
<tr>
<td>Pavement Classification</td>
<td>Interstate Flexible</td>
<td>U. S. Flexible</td>
</tr>
<tr>
<td>Truck Types</td>
<td>2D, 3A, 3-S2, 2-S1-2</td>
<td>2D, 3A, 3-S2, 2-S1-2</td>
</tr>
<tr>
<td>Load Limits:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GVV</td>
<td>80,000 lb.</td>
<td>80,000 lb.</td>
</tr>
<tr>
<td>Single Axle</td>
<td>20,000 lb.</td>
<td>20,000 lb.</td>
</tr>
<tr>
<td>Tandem Axle</td>
<td>34,000 lb.</td>
<td>34,000 lb.</td>
</tr>
<tr>
<td>Critical PSI</td>
<td>1.50</td>
<td>1.50</td>
</tr>
<tr>
<td>PSI after Overlay</td>
<td>4.80</td>
<td>4.70</td>
</tr>
<tr>
<td>Overlay Design Life</td>
<td>20 yrs.</td>
<td>20 yrs.</td>
</tr>
<tr>
<td>Avg. percent of Paved Shoulders</td>
<td>95 %</td>
<td>10 %</td>
</tr>
<tr>
<td>Avg. Paved Shoulder Width per lane</td>
<td>4.75 ft.</td>
<td>0.80 ft.</td>
</tr>
<tr>
<td>Avg. Granular Shoulder Width per lane</td>
<td>0.25 ft.</td>
<td>7.20 ft.</td>
</tr>
<tr>
<td>VEHICLE CLASSES</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>RUN PARAMETERS</td>
<td>18 0 3.35 4.00 0.0 0.0 320000.00</td>
<td></td>
</tr>
<tr>
<td>SYSTEM TITLE</td>
<td>0 0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>INTERSTATE FLEX PAVEMENTS DISTRICT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TEXAS TRANSPORTATION INSTITUTE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAMPLE RUN FOR FLEXIBLE PAVEMENTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLEXIBLE</td>
<td>0 0 12.00 0.0 0.0 2.10 2.10</td>
<td></td>
</tr>
<tr>
<td>INFLEX</td>
<td>0 0 1.00 0.0 0.0 2.10 2.10</td>
<td></td>
</tr>
<tr>
<td>TITLE</td>
<td>0 0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>FLEXIBLE</td>
<td>0 0 12.00 0.0 0.0 2.10 2.10</td>
<td></td>
</tr>
<tr>
<td>INFLEX</td>
<td>0 0 1.00 0.0 0.0 2.10 2.10</td>
<td></td>
</tr>
<tr>
<td>TRUCK TYPE</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>LOAD LIMITS</td>
<td>0 0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>SINGLE AXLES</td>
<td>11 0 3.00 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>TANDEM AXLES</td>
<td>16 0 3.00 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>44</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GVW</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>130</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>86</td>
<td>34</td>
</tr>
<tr>
<td>22</td>
<td>33</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>33</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>19</td>
<td>3</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>65</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>70</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>85</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>95</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EMPTY</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>78</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>143</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>107</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>47</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>35</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PERFORMANCE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OVERLAY</td>
<td>95.00</td>
<td>4.75</td>
</tr>
<tr>
<td>MODEL MAINT</td>
<td>3.47</td>
<td>0.25</td>
</tr>
<tr>
<td>OLD SECTIONS</td>
<td>3.86</td>
<td>8.75</td>
</tr>
</tbody>
</table>

Note: The table above represents various numerical data points, possibly related to performance metrics, overlay settings, model maintenance, and section data, but without specific context, the exact interpretation is unclear.
<table>
<thead>
<tr>
<th></th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUT</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>FLEXILE</td>
<td>0</td>
<td>0</td>
<td>12.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.70</td>
<td>2.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USFLx</td>
<td>B US RURAL HOT MIX HIGH TRAFFIC</td>
<td>17</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>1.25</td>
<td>6.00</td>
<td>3.30</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>ACP</td>
<td>4.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGB</td>
<td>12.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTS</td>
<td>0.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGE DISTRIBUTION</td>
<td>30</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>8.0</td>
<td>8.0</td>
<td>2.0</td>
<td>1.0</td>
<td>9.0</td>
<td>2.0</td>
<td>15.0</td>
<td>2.0</td>
<td>12.0</td>
<td>6.0</td>
<td>3.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>6.00</td>
<td>3.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINGLE AXLES</td>
<td>11</td>
<td>0</td>
<td>3.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>354.</td>
<td>10.</td>
<td>26.</td>
<td>12.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.0</td>
<td>53.</td>
<td>7.</td>
<td>117.</td>
<td>5.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.0</td>
<td>75.</td>
<td>38.</td>
<td>856.</td>
<td>36.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16.0</td>
<td>44.</td>
<td>14.</td>
<td>24.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16.0</td>
<td>23.</td>
<td>1.</td>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>19.0</td>
<td>3.</td>
<td>0.</td>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.0</td>
<td>2.</td>
<td>0.</td>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>22.0</td>
<td>4.</td>
<td>0.</td>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.0</td>
<td>2.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26.0</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TANDEM AXLES</td>
<td>16</td>
<td>0</td>
<td>6.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.0</td>
<td>0.0</td>
<td>12.0</td>
<td>237.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18.0</td>
<td>0.0</td>
<td>16.0</td>
<td>442.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.0</td>
<td>0.0</td>
<td>7.0</td>
<td>238.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.0</td>
<td>0.0</td>
<td>20.0</td>
<td>347.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32.0</td>
<td>0.0</td>
<td>5.0</td>
<td>187.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.0</td>
<td>0.0</td>
<td>1.0</td>
<td>62.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>34.0</td>
<td>0.0</td>
<td>125.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>36.0</td>
<td>0.0</td>
<td>1.0</td>
<td>154.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>38.0</td>
<td>0.0</td>
<td>4.0</td>
<td>98.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40.0</td>
<td>0.0</td>
<td>2.0</td>
<td>51.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42.0</td>
<td>0.0</td>
<td>0.0</td>
<td>29.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>44.0</td>
<td>0.0</td>
<td>1.0</td>
<td>22.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>46.0</td>
<td>0.0</td>
<td>0.0</td>
<td>11.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GVW</td>
<td>29.0</td>
<td>0.0</td>
<td>10.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.0</td>
<td>0.0</td>
<td>3.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.0</td>
<td>0.0</td>
<td>8.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>22.0</td>
<td>0.0</td>
<td>8.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.0</td>
<td>0.0</td>
<td>15.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26.0</td>
<td>0.0</td>
<td>29.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28.0</td>
<td>0.0</td>
<td>5.0</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.0</td>
<td>0.0</td>
<td>1.0</td>
<td>13.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32.0</td>
<td>0.0</td>
<td>3.0</td>
<td>44.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>34.0</td>
<td>0.0</td>
<td>1.0</td>
<td>59.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>36.0</td>
<td>0.0</td>
<td>3.0</td>
<td>53.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>38.0</td>
<td>0.0</td>
<td>7.0</td>
<td>61.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40.0</td>
<td>0.0</td>
<td>4.0</td>
<td>44.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45.0</td>
<td>0.0</td>
<td>8.0</td>
<td>59.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50.0</td>
<td>0.0</td>
<td>4.0</td>
<td>46.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>3</td>
<td>46</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>1</td>
<td>49</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>0</td>
<td>1</td>
<td>58</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>0</td>
<td>0</td>
<td>88</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>0</td>
<td>0</td>
<td>79</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>0</td>
<td>117</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OLD SECTIONS**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>750.00</td>
<td>10.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**PERFORMANCE**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>4.70</td>
<td>1.50</td>
<td>4.70</td>
<td>20.00</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OVERLAY**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10.00</td>
<td>0.80</td>
<td>7.20</td>
<td>66.00</td>
<td>0.50</td>
<td>0.0</td>
<td>0.0</td>
<td>3200.00</td>
</tr>
</tbody>
</table>

**EXECUTE**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
APPENDIX 3

COST OUTPUT FROM THE MODIFIED RENU2 PROGRAM
## VEHICLE COMBINATION (1=IN, 0=OUT)

**1 2 3 4 5 6**

**1.0.0.0.**

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.193E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.235E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.272E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.337E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.389E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.456E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.528E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.615E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.349E+06</td>
<td>0.707E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.811E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.350E+06</td>
<td>0.112E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.128E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.147E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.169E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.191E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.218E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.245E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.350E+06</td>
<td>0.278E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.926E-02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.389E-01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.100E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.389E-01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.106E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.109E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.141E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.109E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.349E+06</td>
<td>0.145E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.211E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.350E+06</td>
<td>0.193E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.254E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.350E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.395E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.456E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.602E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.574E+00</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.350E+06</td>
<td>0.350E+00</td>
<td>0.304E+05</td>
</tr>
</tbody>
</table>

VEHICLE COMBINATION (1=IN, O=OUT)

1 2 3 4 5 6

0 1 0 0
<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.305E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.370E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.448E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.533E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.622E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.732E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.846E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.980E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.349E+06</td>
<td>0.113E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.129E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.350E+06</td>
<td>0.179E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.206E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.236E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.270E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.306E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.348E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.393E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.350E+06</td>
<td>0.444E+02</td>
<td>0.304E+05</td>
</tr>
</tbody>
</table>
VEHICLE COMBINATION (1=IN, 0=OUT)
1 2 3 4 5 6
0.0.1.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.937E+04</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.191E+06</td>
<td>0.113E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.234E+06</td>
<td>0.134E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.276E+06</td>
<td>0.158E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.304E+06</td>
<td>0.185E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.317E+06</td>
<td>0.214E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.321E+06</td>
<td>0.246E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.319E+06</td>
<td>0.280E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.315E+06</td>
<td>0.318E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.309E+06</td>
<td>0.358E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.303E+06</td>
<td>0.490E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.296E+06</td>
<td>0.552E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.289E+06</td>
<td>0.615E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.282E+06</td>
<td>0.688E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.274E+06</td>
<td>0.760E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.266E+06</td>
<td>0.834E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.257E+06</td>
<td>0.910E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.249E+06</td>
<td>0.987E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.101E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.191E+06</td>
<td>0.121E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.233E+06</td>
<td>0.144E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.275E+06</td>
<td>0.170E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.303E+06</td>
<td>0.198E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.316E+06</td>
<td>0.229E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.319E+06</td>
<td>0.263E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.317E+06</td>
<td>0.300E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.313E+06</td>
<td>0.339E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.307E+06</td>
<td>0.382E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.300E+06</td>
<td>0.523E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.293E+06</td>
<td>0.588E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.285E+06</td>
<td>0.657E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.277E+06</td>
<td>0.729E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.269E+06</td>
<td>0.804E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.261E+06</td>
<td>0.881E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.252E+06</td>
<td>0.959E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.243E+06</td>
<td>0.104E+06</td>
<td>0.304E+05</td>
</tr>
</tbody>
</table>
VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.1.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.951E+04</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.191E+06</td>
<td>0.115E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.234E+06</td>
<td>0.136E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.276E+06</td>
<td>0.160E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.304E+06</td>
<td>0.187E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.317E+06</td>
<td>0.217E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.321E+06</td>
<td>0.249E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.319E+06</td>
<td>0.284E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.315E+06</td>
<td>0.322E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.309E+06</td>
<td>0.363E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.302E+06</td>
<td>0.497E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.296E+06</td>
<td>0.560E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.288E+06</td>
<td>0.626E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.281E+06</td>
<td>0.696E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.273E+06</td>
<td>0.769E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.265E+06</td>
<td>0.844E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.256E+06</td>
<td>0.921E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.248E+06</td>
<td>0.998E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.102E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.191E+06</td>
<td>0.123E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.233E+06</td>
<td>0.146E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.275E+06</td>
<td>0.172E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.303E+06</td>
<td>0.201E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.316E+06</td>
<td>0.232E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.319E+06</td>
<td>0.266E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.317E+06</td>
<td>0.304E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.312E+06</td>
<td>0.344E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.306E+06</td>
<td>0.387E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.299E+06</td>
<td>0.530E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.292E+06</td>
<td>0.595E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.284E+06</td>
<td>0.665E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.276E+06</td>
<td>0.738E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.268E+06</td>
<td>0.813E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.260E+06</td>
<td>0.890E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.251E+06</td>
<td>0.969E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.242E+06</td>
<td>0.105E+06</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.625E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.757E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.903E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.107E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.127E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.148E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.172E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.199E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.349E+06</td>
<td>0.229E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.263E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.350E+06</td>
<td>0.363E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.417E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.478E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.545E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.622E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.706E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.798E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.350E+06</td>
<td>0.902E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.202E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.244E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.293E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.347E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.410E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.479E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.558E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.645E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.348E+06</td>
<td>0.742E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.851E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.349E+06</td>
<td>0.117E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.135E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.155E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.177E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.201E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.228E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.259E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.350E+06</td>
<td>0.292E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.842E+01</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.102E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.122E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.145E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.171E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.200E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.233E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.268E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.349E+06</td>
<td>0.310E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.355E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.350E+06</td>
<td>0.488E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.563E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.646E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.737E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.839E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.952E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.108E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.350E+06</td>
<td>0.122E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.244E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.194E+06</td>
<td>0.295E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.239E+06</td>
<td>0.355E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.284E+06</td>
<td>0.420E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.316E+06</td>
<td>0.495E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.334E+06</td>
<td>0.579E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.343E+06</td>
<td>0.674E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.347E+06</td>
<td>0.779E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.348E+06</td>
<td>0.898E+02</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.349E+06</td>
<td>0.103E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.349E+06</td>
<td>0.142E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.350E+06</td>
<td>0.163E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.350E+06</td>
<td>0.187E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.350E+06</td>
<td>0.214E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.350E+06</td>
<td>0.243E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.350E+06</td>
<td>0.276E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.350E+06</td>
<td>0.313E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.349E+06</td>
<td>0.353E+03</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.105E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.191E+06</td>
<td>0.126E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.233E+06</td>
<td>0.150E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.275E+06</td>
<td>0.177E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.302E+06</td>
<td>0.206E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.315E+06</td>
<td>0.238E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.318E+06</td>
<td>0.274E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.316E+06</td>
<td>0.312E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.311E+06</td>
<td>0.353E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.306E+06</td>
<td>0.397E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.298E+06</td>
<td>0.543E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.290E+06</td>
<td>0.610E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.283E+06</td>
<td>0.681E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.275E+06</td>
<td>0.755E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.266E+06</td>
<td>0.831E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.258E+06</td>
<td>0.908E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.249E+06</td>
<td>0.988E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.240E+06</td>
<td>1.07E+06</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.112E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.190E+06</td>
<td>0.135E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.233E+06</td>
<td>0.161E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.274E+06</td>
<td>0.183E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.301E+06</td>
<td>0.220E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.314E+06</td>
<td>0.254E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.317E+06</td>
<td>0.292E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.314E+06</td>
<td>0.332E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.309E+06</td>
<td>0.375E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.302E+06</td>
<td>0.421E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.295E+06</td>
<td>0.576E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.287E+06</td>
<td>0.647E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.279E+06</td>
<td>0.720E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.270E+06</td>
<td>0.797E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.261E+06</td>
<td>0.875E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.252E+06</td>
<td>0.955E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.243E+06</td>
<td>0.104E+06</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.235E+06</td>
<td>0.112E+06</td>
<td>0.304E+05</td>
</tr>
</tbody>
</table>
VEHICLE COMBINATION (1=IN, 0=OUT)
1 2 3 4 5 6

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.107E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.191E+06</td>
<td>0.128E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.233E+06</td>
<td>0.152E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.274E+06</td>
<td>0.179E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.302E+06</td>
<td>0.209E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.315E+06</td>
<td>0.242E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.318E+06</td>
<td>0.277E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.316E+06</td>
<td>0.316E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.311E+06</td>
<td>0.358E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.304E+06</td>
<td>0.402E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.297E+06</td>
<td>0.500E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.290E+06</td>
<td>0.618E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.282E+06</td>
<td>0.689E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.274E+06</td>
<td>0.764E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.265E+06</td>
<td>0.840E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.256E+06</td>
<td>0.919E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.248E+06</td>
<td>0.998E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.239E+06</td>
<td>1.084E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.160E+06</td>
<td>0.114E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.190E+06</td>
<td>0.137E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.233E+06</td>
<td>0.163E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.274E+06</td>
<td>0.192E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.301E+06</td>
<td>0.223E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.314E+06</td>
<td>0.258E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.316E+06</td>
<td>0.296E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.314E+06</td>
<td>0.336E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.308E+06</td>
<td>0.380E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.301E+06</td>
<td>0.427E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.294E+06</td>
<td>0.583E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.286E+06</td>
<td>0.654E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.278E+06</td>
<td>0.729E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.269E+06</td>
<td>0.806E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.260E+06</td>
<td>0.885E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.251E+06</td>
<td>0.965E+05</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.242E+06</td>
<td>0.105E+06</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.233E+06</td>
<td>0.113E+06</td>
<td>0.304E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.108E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.549E+06</td>
<td>0.134E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.593E+06</td>
<td>0.160E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.635E+06</td>
<td>0.193E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.671E+06</td>
<td>0.226E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.703E+06</td>
<td>0.263E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.732E+06</td>
<td>0.316E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.759E+06</td>
<td>0.365E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.785E+06</td>
<td>0.425E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.804E+06</td>
<td>0.487E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.817E+06</td>
<td>0.678E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.823E+06</td>
<td>0.784E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.826E+06</td>
<td>0.807E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.827E+06</td>
<td>0.104E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.828E+06</td>
<td>0.120E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.828E+06</td>
<td>0.137E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.828E+06</td>
<td>0.157E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.828E+06</td>
<td>0.179E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.888E-02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.549E+06</td>
<td>0.333E-02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.593E+06</td>
<td>0.388E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.635E+06</td>
<td>0.155E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.671E+06</td>
<td>0.621E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.703E+06</td>
<td>0.277E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.732E+06</td>
<td>0.344E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.759E+06</td>
<td>0.732E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.785E+06</td>
<td>0.566E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.804E+06</td>
<td>0.910E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.817E+06</td>
<td>0.832E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.823E+06</td>
<td>0.943E-01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.826E+06</td>
<td>0.162E+00</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.827E+06</td>
<td>0.136E+00</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.828E+06</td>
<td>0.203E+00</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.828E+06</td>
<td>0.243E+00</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.828E+06</td>
<td>0.195E+00</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.828E+06</td>
<td>0.321E+00</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.184E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.549E+06</td>
<td>0.222E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.593E+06</td>
<td>0.270E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.635E+06</td>
<td>0.318E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.671E+06</td>
<td>0.382E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.703E+06</td>
<td>0.443E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.732E+06</td>
<td>0.521E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.759E+06</td>
<td>0.611E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.785E+06</td>
<td>0.713E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.804E+06</td>
<td>0.819E+01</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.817E+06</td>
<td>0.913E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.823E+06</td>
<td>0.131E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.826E+06</td>
<td>0.152E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.827E+06</td>
<td>0.175E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.828E+06</td>
<td>0.201E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.828E+06</td>
<td>0.229E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.828E+06</td>
<td>0.263E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.828E+06</td>
<td>0.298E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.123E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.541E+06</td>
<td>0.147E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.580E+06</td>
<td>0.175E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.615E+06</td>
<td>0.207E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.644E+06</td>
<td>0.243E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.666E+06</td>
<td>0.283E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.684E+06</td>
<td>0.328E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.699E+06</td>
<td>0.377E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.711E+06</td>
<td>0.431E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.716E+06</td>
<td>0.490E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.712E+06</td>
<td>0.672E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.702E+06</td>
<td>0.762E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.688E+06</td>
<td>0.858E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.671E+06</td>
<td>0.961E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.652E+06</td>
<td>1.07E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.631E+06</td>
<td>1.18E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.610E+06</td>
<td>1.30E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.588E+06</td>
<td>1.41E+06</td>
<td>0.720E+05</td>
</tr>
</tbody>
</table>

VEHICLE COMBINATION (1=IN, 0=OUT)
1 2 3 4 5 6
0 0 1 0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC
<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.133E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.541E+06</td>
<td>0.159E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.579E+06</td>
<td>0.189E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.613E+06</td>
<td>0.224E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.642E+06</td>
<td>0.262E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.663E+06</td>
<td>0.305E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.680E+06</td>
<td>0.353E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.695E+06</td>
<td>0.405E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.706E+06</td>
<td>0.463E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.709E+06</td>
<td>0.525E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.705E+06</td>
<td>0.720E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.684E+06</td>
<td>0.815E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.678E+06</td>
<td>0.916E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.660E+06</td>
<td>0.102E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.640E+06</td>
<td>0.114E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.619E+06</td>
<td>0.125E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.597E+06</td>
<td>0.137E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.574E+06</td>
<td>0.149E+06</td>
<td>0.720E+05</td>
</tr>
</tbody>
</table>
### VEHICLE COMBINATION (1=IN, 0=OUT)
1 2 3 4 5 6
0.1.1.0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.125E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.541E+06</td>
<td>0.150E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.580E+06</td>
<td>0.178E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.615E+06</td>
<td>0.211E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.643E+06</td>
<td>0.247E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.666E+06</td>
<td>0.288E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.683E+06</td>
<td>0.333E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.698E+06</td>
<td>0.383E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.710E+06</td>
<td>0.438E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.715E+06</td>
<td>0.497E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.711E+06</td>
<td>0.682E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.700E+06</td>
<td>0.773E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.686E+06</td>
<td>0.870E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.668E+06</td>
<td>0.974E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.649E+06</td>
<td>1.08E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.629E+06</td>
<td>1.20E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.607E+06</td>
<td>1.31E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.585E+06</td>
<td>0.143E+06</td>
<td>0.720E+05</td>
</tr>
</tbody>
</table>
### VEHICLE COMBINATION (1=IN, 0=OUT)

1 2 3 4 5 6

1. S.R.R.L.

**USFLX A US RURAL HOT MIX HIGH TRAFFIC**

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.135E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.541E+06</td>
<td>0.162E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.579E+06</td>
<td>0.192E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.613E+06</td>
<td>0.227E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.641E+06</td>
<td>0.266E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.663E+06</td>
<td>0.310E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.680E+06</td>
<td>0.358E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.694E+06</td>
<td>0.411E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.704E+06</td>
<td>0.470E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.708E+06</td>
<td>0.533E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.703E+06</td>
<td>0.730E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.692E+06</td>
<td>0.826E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.676E+06</td>
<td>0.928E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.658E+06</td>
<td>0.104E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.637E+06</td>
<td>0.115E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.616E+06</td>
<td>0.127E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.594E+06</td>
<td>0.139E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.571E+06</td>
<td>0.150E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>5.05E+06</td>
<td>4.06E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>2</td>
<td>5.49E+06</td>
<td>4.86E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>3</td>
<td>5.93E+06</td>
<td>5.82E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>4</td>
<td>6.35E+06</td>
<td>6.97E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>5</td>
<td>6.71E+06</td>
<td>8.23E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>6</td>
<td>7.03E+06</td>
<td>9.66E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>7</td>
<td>7.32E+06</td>
<td>1.14E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>8</td>
<td>7.59E+06</td>
<td>1.33E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>9</td>
<td>7.85E+06</td>
<td>1.54E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>10</td>
<td>8.04E+06</td>
<td>1.79E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>11</td>
<td>8.17E+06</td>
<td>2.46E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>12</td>
<td>8.23E+06</td>
<td>2.85E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>13</td>
<td>8.26E+06</td>
<td>3.30E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>14</td>
<td>8.27E+06</td>
<td>3.80E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>15</td>
<td>8.28E+06</td>
<td>4.36E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>16</td>
<td>8.28E+06</td>
<td>4.99E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>17</td>
<td>8.28E+06</td>
<td>5.71E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>18</td>
<td>8.28E+06</td>
<td>6.50E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.146E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.549E+06</td>
<td>0.177E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.593E+06</td>
<td>0.212E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.635E+06</td>
<td>0.252E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.671E+06</td>
<td>0.299E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.703E+06</td>
<td>0.352E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.732E+06</td>
<td>0.413E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.759E+06</td>
<td>0.482E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.784E+06</td>
<td>0.560E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.804E+06</td>
<td>0.648E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.817E+06</td>
<td>0.895E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.823E+06</td>
<td>0.104E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.826E+06</td>
<td>0.120E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.827E+06</td>
<td>0.138E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.828E+06</td>
<td>0.158E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.828E+06</td>
<td>0.181E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.828E+06</td>
<td>0.207E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.828E+06</td>
<td>0.236E+03</td>
<td>0.720E+05</td>
</tr>
</tbody>
</table>
## VEHICLE COMBINATION (1=IN,0=OUT)

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.05E+06</td>
<td>5.59E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>2</td>
<td>5.49E+06</td>
<td>6.74E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>3</td>
<td>5.93E+06</td>
<td>8.10E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>4</td>
<td>6.35E+06</td>
<td>9.66E+01</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>5</td>
<td>6.71E+06</td>
<td>1.15E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>6</td>
<td>7.03E+06</td>
<td>1.35E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>7</td>
<td>7.32E+06</td>
<td>1.58E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>8</td>
<td>7.59E+06</td>
<td>1.84E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>9</td>
<td>7.85E+06</td>
<td>2.14E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>10</td>
<td>8.04E+06</td>
<td>2.48E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>11</td>
<td>8.17E+06</td>
<td>3.42E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>12</td>
<td>8.23E+06</td>
<td>3.97E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>13</td>
<td>8.26E+06</td>
<td>4.58E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>14</td>
<td>8.27E+06</td>
<td>5.27E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>15</td>
<td>8.28E+06</td>
<td>6.07E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>16</td>
<td>8.28E+06</td>
<td>6.94E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>17</td>
<td>8.28E+06</td>
<td>7.93E+02</td>
<td>7.20E+05</td>
</tr>
<tr>
<td>18</td>
<td>8.28E+06</td>
<td>9.04E+02</td>
<td>7.20E+05</td>
</tr>
</tbody>
</table>
### VEHICLE COMBINATION (1=IN, 0=OUT)
1 2 3 4 5 6
1.1.0.1.

**USFLX A US RURAL HOT MIX HIGH TRAFFIC**

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.180E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.549E+06</td>
<td>0.217E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.593E+06</td>
<td>0.261E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.635E+06</td>
<td>0.311E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.671E+06</td>
<td>0.368E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.703E+06</td>
<td>0.434E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.732E+06</td>
<td>0.508E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.759E+06</td>
<td>0.599E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.784E+06</td>
<td>0.689E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.804E+06</td>
<td>0.798E+02</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.817E+06</td>
<td>0.110E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.823E+06</td>
<td>0.128E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.826E+06</td>
<td>0.147E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.827E+06</td>
<td>0.170E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.828E+06</td>
<td>0.195E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.828E+06</td>
<td>0.223E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.828E+06</td>
<td>0.255E+03</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.828E+06</td>
<td>0.291E+03</td>
<td>0.720E+05</td>
</tr>
</tbody>
</table>
VEHICLE COMBINATION (1=IN,0=OUT)

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROUT MAINT COST</th>
<th>REHAB COST</th>
<th>PREV MAINT COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.139E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.540E+06</td>
<td>0.167E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.578E+06</td>
<td>0.198E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.612E+06</td>
<td>0.234E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.640E+06</td>
<td>0.274E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.661E+06</td>
<td>0.319E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.678E+06</td>
<td>0.369E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.692E+06</td>
<td>0.423E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.702E+06</td>
<td>0.483E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.705E+06</td>
<td>0.547E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.700E+06</td>
<td>0.750E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.688E+06</td>
<td>0.848E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.672E+06</td>
<td>0.952E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.653E+06</td>
<td>1.06E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.633E+06</td>
<td>1.18E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.611E+06</td>
<td>1.29E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.588E+06</td>
<td>1.41E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.566E+06</td>
<td>1.53E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.150E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.540E+06</td>
<td>0.179E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.577E+06</td>
<td>0.213E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.611E+06</td>
<td>0.251E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.638E+06</td>
<td>0.294E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.698E+06</td>
<td>0.342E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.674E+06</td>
<td>0.395E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.687E+06</td>
<td>0.453E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.696E+06</td>
<td>0.516E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.698E+06</td>
<td>0.584E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.692E+06</td>
<td>0.800E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.679E+06</td>
<td>0.803E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.662E+06</td>
<td>0.101E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.642E+06</td>
<td>0.113E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.620E+06</td>
<td>0.124E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.598E+06</td>
<td>0.136E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.575E+06</td>
<td>0.149E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.552E+06</td>
<td>0.161E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.141E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.540E+06</td>
<td>0.169E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.578E+06</td>
<td>0.201E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.612E+06</td>
<td>0.238E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.646E+06</td>
<td>0.278E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.667E+06</td>
<td>0.324E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.677E+06</td>
<td>0.374E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.691E+06</td>
<td>0.429E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.701E+06</td>
<td>0.490E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.704E+06</td>
<td>0.555E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.686E+06</td>
<td>0.761E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.670E+06</td>
<td>0.859E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.650E+06</td>
<td>0.965E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.651E+06</td>
<td>0.108E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.630E+06</td>
<td>0.119E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.608E+06</td>
<td>0.131E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.585E+06</td>
<td>0.143E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.563E+06</td>
<td>0.158E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>YEAR</td>
<td>ROUT MAINT COST</td>
<td>REHAB COST</td>
<td>PREV MAINT COST</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
<td>0.505E+06</td>
<td>0.152E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>2</td>
<td>0.540E+06</td>
<td>0.182E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>3</td>
<td>0.577E+06</td>
<td>0.216E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>4</td>
<td>0.610E+06</td>
<td>0.255E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>5</td>
<td>0.637E+06</td>
<td>0.299E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.658E+06</td>
<td>0.347E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>7</td>
<td>0.673E+06</td>
<td>0.401E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.686E+06</td>
<td>0.459E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.695E+06</td>
<td>0.523E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.697E+06</td>
<td>0.582E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>11</td>
<td>0.690E+06</td>
<td>0.811E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.677E+06</td>
<td>0.914E+05</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>13</td>
<td>0.659E+06</td>
<td>0.102E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.639E+06</td>
<td>0.114E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.618E+06</td>
<td>0.126E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>16</td>
<td>0.595E+06</td>
<td>0.138E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>17</td>
<td>0.572E+06</td>
<td>0.150E+06</td>
<td>0.720E+05</td>
</tr>
<tr>
<td>18</td>
<td>0.549E+06</td>
<td>0.162E+06</td>
<td>0.720E+05</td>
</tr>
</tbody>
</table>
APPENDIX 4

MODIFIED INCREMENTAL APPROACH BASIC CODE
THE MODIFIED INCREMENTAL APPROACH IS EXECUTED BY CALLING THIS ROUTINE, WHICH ASKS FOR INPUT DATA AND CALCULATES COST PARTITIONS (P) AS DISCUSSED IN CHAPTER 4.

```basic
1 COMMON NVC,NDIM,P()
10 ' INPUT ROUTINE
20 ' 35 PRINT CHRS(26):PRINT:PRINT:PRINT
40 INPUT "NUMBER OF VEHICLE CLASSES";NVC
41 LPRINT "NUMBER OF VEHICLE CLASSES";NVC
50 NDIM = 2^NVC - 1
60 DIM COMBI(NDIM),BIN(NVC),P(NDIM)
61 DIM SUMP(NDIM)
70 PRINT: PRINT:PRINT:PRINT:PRINT:PRINT
80 FOR INDEX=O TO NDIM
85 PRINT
90 IEXP = 1
95 IF INDEX=0 THEN PRINT"COST OF DESIGN FOR NO CLASSES";LPRINT "COST OF DESIGN FOR NO CLASSES";GOTO 220
100 PRINT "COST OF DESIGN FOR CLASSES ";
101 LPRINT "COST OF DESIGN FOR CLASSES ";
110 INQUOT = INDEX
120 QUOT = INQUOT/2
130 INQUOT = INT(QUOT)
140 RES = (QUOT-INQUOT)*2
150 IND = RES*IEXP
160 IF IND <> 0 THEN PRINT IND;LPRINT IND;
170 WHILE INQUOT>0
180 IF IND<>0 THEN PRINT ",";LPRINT ",";
190 IEXP = IEXP+1
200 GOTO 120
210 WEND
215 PRINT TAB(60);LPRINT TAB(60);
220 INPUT ": ,COMBI(INDEX)
221 LPRINT ": ,COMBI(INDEX)
230 NEXT INDEX
240 GOSUB 1000
300 ' CREATE COST PARTITIONS
310 ' CONVERT INDEXES TO BINARY
315 PRINT CHRS(26) .
320 '
```

FILE NAME: MIA
330 FOR IOTA=1 TO NDOM
340 FOR J=1 TO NVC
350 BIN(J)=0
360 NEXT J
370 INQUOT=IOTA
380 J=1
390 QUOT=INQUOT/2
400 INQUOT=INT(QUOT)
410 RES=(QUOT-INQUOT)*2
420 BIN(J)=RES
430 IF INQUOT=0 THEN GOTO 493
440 J=J+1 : GOTO 390
493 ' 600 ' IDENTIFY POSITIONS FOR ONES
610 FOR K=1 TO NVC: PST(K)=0:NEXT K
620 FOR J=1 TO NVC
630 IDX=0
640 IF BIN(J)=0 THEN GOTO 680
650 IDX=IDX+1
660 PST(IDX)=J
670 NEXT J
680 NCOMB=2^IDX-2
690 DIM CB(NCOMB)
700 FOR J=1 TO NCOMB
710 CB(J)=0
720 INQUOT=J
730 PINX=0
740 PINX=1
750 QUOT=INQUOT/2
760 INQUOT=INT(QUOT)
770 RES=(QUOT-INQUOT)*2
780 PINX=PINX+RES*(2^PST(INX)-1)
790 INX=INX+1
800 IF INQUOT<>0 THEN GOTO 760
810 CB(J)=P(PINX)
820 NEXT J
830 ' SUM OF COMBINATIONS
840 SUMP(IOTA)=0
850 FOR J=1 TO NCOMB
860 SUMP(IOTA)=SUMP(IOTA)+CB(J)
870 NEXT J
880 P(IOTA)=P(IOTA)-SUMP(IOTA)
890 PRINT"COST DUE TO THE ACTION OF CLASSES";
900 FOR W=1 TO IDX:PRINT PST(W);:NEXT W:PRINT "IS ";
910 PRINT P(IOTA):PRINT
920 ERASE CB
930 NEXT IOTA
940 CHAIN "GENER", ALL

159
1000 FOR INDEX=0 TO NDIM
1010 P(INDEX)=COMBI(NDIM)-COMBI(NDIM-INDEX)
1020 NEXT INDEX
1030 RETURN
THIS PROGRAM ACCEPTS VEHICLE-MILES-OF-TRAVEL DATA FOR EACH
VEHICLE CLASS, CALCULATES ALLOCATED COSTS, AND REPORTS
RESULTS FROM THE MODIFIED INCREMENTAL APPROACH.

10 DIM PMAT(NVC,NDIM),VREC(NDIM),VMT(NVC),VMTMAT(NVC,NVC)
20 '  *** INPUT VALUES OF VMT ***
30 ' 40 ' 45 LPRINT:LPRINT:LPRINT
50 FOR I=1 TO NVC
60 PRINT "VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS";I;
70 INPUT "",VMT(I)
71 LPRINT "VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS";
80 I;" ";VMT(I)
80 NEXT I
90 ' 100 ' 110 ' 120 FOR J=1 TO NDIM
130 POSITION=1
140 SUMVMT=0
150 ' 160 ' CONVINAT J TO BINARY
170 ' 180 INQUOT=J
190 QUOT=INQUOT/2
200 INQUOT=INT(QUOT)
210 RES=(QUOT-INQUOT)*2
220 ' STORE P(J) IN APPROPRIATE POSITION IN COLUMN J
230 PMAT(POSITION,J)=RES*P(J)
240 ' 250 SUMVMT=SUMVMT+RES*(VMT(POSITION))
260 POSITION=POSITION+1
270 IF INQUOT>0 THEN GOTO 190
280 ' GENERATE J'th ELEMENT OF VREC
290 IF SUMVMT=0 THEN VREC(J)=0 ELSE VREC(J)=1/SUMVMT
300 NEXT J
310 ' 320 ' 330 ' 340 FOR I=1 TO NVC
350 FOR J=1 TO NVC
360 IF I=J THEN VMTMAT(I,J)=VMT(I) ELSE VMTMAT(I,J)=0
370 NEXT J:NEXT I
380 ' 390 ' 400 ' 410 ' 420 FLAG=0
430 ADIM=NVC
510 ABDIM=NVC
520 BDIM=NDIM
530 DIM A(ADIM,ABDIM),B(ABDIM,BDIM)
540 FOR I=1 TO ADIM
550 FOR J=1 TO ABDIM
560 A(I,J)=VMTMAT(I,J)
570 NEXT J
580 FOR K=1 TO BDIM
590 B(I,K)=PMAT(I,K)
600 NEXT K
610 NEXT I
620 GOSUB 2000
630 DIM ALLOC(NVC,NDIM)
640 FOR I=1 TO NVC
650 FOR J=1 TO NDIM
660 ALLOC(I,J)=C(I,J)
670 NEXT J:NEXT I
680 ERASE A,B,C
690 ADIM=NVC
700 ABDIM=NDIM
710 BDIM=1
720 DIM A(ADIM,ABDIM),B(ABDIM,BDIM)
730 FOR J=1 TO ABDIM
740 FOR I=1 TO ADIM
750 A(I,J)=ALLOC(I,J)
760 NEXT I
770 B(J,BDIM)=VREC(J)
780 NEXT J
785 FLAG=1
790 GOSUB 2000
800 ERASE ALLOC
810 PRINT CHR$(26):PRINT:PRINT:PRINT
811 LPRINT:LPRINT:LPRINT
820 FOR I=1 TO NVC
830 ALLOC(I)=C(I,1)+VMT(I)*VREC(NDIM)*COMBI(0)
840 PRINT:LPRINT
850 PRINT "COST ALLOCATED TO VEHICLE CLASS";I;": $";
851 LPRINT "COST ALLOCATED TO VEHICLE CLASS";I;": ";
852 ALLOC(I)
860 NEXT I
870 ZARRAP$=INKEY$
890 IF ZARRAP$="" THEN GOTO 870
900 END
2000 CHAIN "MATPROD",ALL
2010 IF FLAG THEN 800 ELSE 630
2020 RETURN
FILE NAME: MATPROD.BAS

THIS ROUTINE PERFORMS MATRIX PRODUCTS.

10 DIM C(ADIM,BDIM)
20 ' PERFORM MATRIX MULTIPLICATION
30 ' FOR I=1 TO ADIM
40 ' FOR J=1 TO BDIM
50 FOR I=1 TO ADIM
60 FOR J=1 TO BDIM
70 C(I,J)=0
80 FOR K=1 TO ABDIM
90 C(I,J)=C(I,J)+A(I,K)*B(K,J)
100 NEXT K
110 NEXT J
120 NEXT I
130 CHAIN "GENER",2010,ALL
APPENDIX 5

SAMPLE RUN FOR THE MODIFIED INCREMENTAL APPROACH
NUMBER OF VEHICLE CLASSES: 4

COST OF DESIGN FOR NO CLASSES: 0
COST OF DESIGN FOR CLASSES 1 : 1.06
COST OF DESIGN FOR CLASSES 2 : .76
COST OF DESIGN FOR CLASSES 1, 2 : 1.11
COST OF DESIGN FOR CLASSES 3 : 1.87
COST OF DESIGN FOR CLASSES 1, 3 : 2.04
COST OF DESIGN FOR CLASSES 2, 3 : 1.9
COST OF DESIGN FOR CLASSES 1, 2, 3 : 2.06
COST OF DESIGN FOR CLASSES 4 : 1.18
COST OF DESIGN FOR CLASSES 1, 4 : 1.46
COST OF DESIGN FOR CLASSES 2, 4 : 1.24
COST OF DESIGN FOR CLASSES 1, 2, 4 : 1.51
COST OF DESIGN FOR CLASSES 3, 4 : 2.105
COST OF DESIGN FOR CLASSES 1, 3, 4 : 2.22
COST OF DESIGN FOR CLASSES 2, 3, 4 : 2.13
COST OF DESIGN FOR CLASSES 1, 2, 3, 4 : 2.24

VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 1 : 96.43
VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 2 : 1.18
VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 3 : 2.06
VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 4 : .33

COST ALLOCATED TO VEHICLE CLASS 1 : 1.03065
COST ALLOCATED TO VEHICLE CLASS 2 : .0393068
COST ALLOCATED TO VEHICLE CLASS 3 : .95709
COST ALLOCATED TO VEHICLE CLASS 4 : .212958
GENERALIZED METHOD BASIC CODE

FILE NAME: GM2.BAS

THIS ACCEPTS INPUT DATA AND CONSTRUCTS THE INITIAL MATRIX FOR THE LINEAR PROGRAMMING PROCEDURE INVOLVED IN THE GENERALIZED METHOD.

10 PRINT CHR$(26)
20 INPUT "Number of vehicle classes: ", NVC
21 LPRINT "Number of vehicle classes: "; NVC
30 PRINT CHR$(26)
40 NDIM = 2^NVC - 1
50 DIM INTABLE(NDIM+2, NVC+2)
60 FOR I = 1 TO NDIM+2
70 FOR J = 1 TO NVC+2
80 INTABLE(I, J) = 0
90 NEXT J
100 NEXT I
105 LPRINT: LPRINT: LPRINT
110 FOR I = 1 TO NDIM
120 JCOLUMN = 1
130 INQUOT = I
140 PRINT "COST WHEN SYSTEM IS USED BY VEHICLE CLASSES";
150 LPRINT "COST WHEN SYSTEM IS USED BY VEHICLE CLASSES";
160 QUOT = INQUOT / 2
170 INQUOT = INT(QUOT)
180 RES = (QUOT - INQUOT) * 2
190 INTABLE(I, JCOLUMN) = RES
200 LABEL = RES * JCOLUMN
210 IF LABEL > 0 THEN PRINT LABEL;: LPRINT LABEL;
220 WHILE INQUOT > 0
230 IF LABEL < 0 THEN PRINT ",", : LPRINT ",", ;
240 JCOLUMN = JCOLUMN + 1
250 GOTO 140
260 WEND
270 PRINT TAB(60);: LPRINT TAB(60);
280 INPUT ": ", INTABLE(I, NVC+2)
290 LPRINT ": "; INTABLE(I, NVC+2)
300 NEXT I
310 INTABLE(NDIM, NVC+1) = 1
320 FOR J = 1 TO NVC+2
330 INTABLE(NDIM+1, J) = -INTABLE(NDIM, J)
340 NEXT J
350 INTABLE(NDIM+2, NVC+1) = 1
360 PRINT CHR$(26)
370 FOR I = 1 TO NDIM + 2
380     FOR J = 1 TO NVC + 2
390         IF J > NVC + 2 THEN PRINT INTABLE(I, J); " ";
             ELSE PRINT INTABLE(I, J)
        NEXT J: NEXT I
400 NEXT J: NEXT I
410 DIM TABLE(NDIM + 2, NVC + 2), IROW(NDIM + 1), JCOL(NVC + 1),
             R$(NDIM + 1), C$(NVC + 1)
420 NR = NDIM + 1; NC = NVC + 1; NI = NR + 1; NJ = NC + 1
430 CHAIN "SLP3", ALL
440 END
FILE NAME: SLP.BAS

This file solves the linear programming problem associated with the generalized method and performs a test for multiple solutions. The linear programming problem is solved through the so-called symmetric method.

30 X$="X":Y$="Y":TOL=.0001
40 GOSUB 1200
50 FOR I=1 TO NR:R$(I)=Y$:IROW(I)=I:NEXT I
60 FOR J=1 TO NC:C$(J)=X$:JCOL(J)=J:NEXT J
70 GOTO 90
80 GOTO 390
90 PIV=-1E+30
100 LL=1
110 MM=5
120 IF MM>NC THEN MM=NC
130 REM PRINT TABLE HEADINGS. COLUMN INDICATORS
140 PRINT:PRINT " ----------------- STEP -----------------"
150 PRINT:PRINT "KNT;" " ----------------- "
160 PRINT:PRINT TAB(9);
170 FOR J=LL TO MM
180 PRINT USING" !";C$(J);
190 PRINT USING"##";JCOL(J);
200 NEXT J
210 IF MM=NC THEN PRINT " R.H.S." ELSE PRINT
220 PRINT
230 MMM=MM+1
240 FOR I=1 TO NR
250 PRINT USING" !";R$(I);
260 PRINT USING"##";IROW(I);
270 FOR J=LL TO MMM
280 PRINT USING" ##.###©©©© ";TABLE(I,J);
290 NEXT J:PRINT
300 NEXT I:PRINT
310 PRINT " OBJ ";
320 FOR J=LL TO MMM
330 PRINT USING" ##.###©©©© ";TABLE(NI,J);
340 NEXT J:PRINT:PRINT
350 LL=LL+5
360 IF LL>NC GOTO 390
370 MM=MM+5
380 GOTO 120
470 IF TABLE(I,J)>=0 GOTO 530
480 IF TABLE(NI,J)>0 GOTO 530
490 VALUE=TABLE(NI,J)/TABLE(I,J)
500 IF VALUE>=CH GOTO 530
510 CH=VALUE
520 JC=J
530 NEXT J
540 IF CH=1E+38 GOTO 600
550 VALUE=-TABLE(NI,JC)*TABLE(I,NJ)/TABLE(I,J)
560 IF VALUE<=PIV GOTO 600
570 PIV=VALUE
580 II=I
590 JJ=JC
600 NEXT I
610 REM CHECK COLUMN INDICATORS
620 FOR J=NEP TO NC
630 IF TABLE(NI,J)<=0 GOTO 800
640 MM=1
650 CH=1E+38
660 FOR I=1 TO NR
670 IF TABLE(I,J)<=0 GOTO 730
680 IF TABLE(I,NJ)<0 GOTO 730
690 VALUE=TABLE(I,NJ)/TABLE(I,J)
700 IF VALUE>=CH GOTO 730
710 CH=VALUE
720 IR=I
730 NEXT I
740 IF CH=1E+38 GOTO 800
750 VALUE=TABLE(NI,J)*TABLE(IR,NJ)/TABLE(IR,J)
760 IF VALUE<PIV GOTO 800
770 PIV=VALUE
780 II=IR
790 JJ=J
800 NEXT J
810 REM OPTIMAL SOLUTION
820 IF MM=0 GOTO 1050
830 REM INFEASIBLE SOLUTION
840 IF PIV=-1E+38 GOTO 1070
850 REM PERFORM INVERSION WITH INDICATED PIVOT ELEMENT
860 FOR I=1 TO NI
870 IF I=II GOTO 920
880 FOR J=1 TO NJ
890 IF J=JJ GOTO 910
900 TABLE(I,J)=TABLE(I,J)-TABLE(II,J)*TABLE(I,JJ)/TABLE(II,JJ)
910 NEXT J
920 NEXT I
930 FOR J=1 TO NJ
940 IF J=JJ GOTO 960
950 TABLE(II,J)=TABLE(II,J)/TABLE(II,JJ)
960 NEXT J
970 FOR I=1 TO NI
IF I=II GOTO 1000
TABLE(I,JJ)=TABLE(I,JJ)/TABLE(II,JJ)
NEXT I
TABLE(II,JJ)=1/TABLE(II,JJ)
SWAP IROW(II),JCOL(JJ)
SWAP R$(II),C$(JJ)
GOTO 90
PRINT "***** OPTIMAL SOLUTION"
GOTO 1080
PRINT "***** INFEASIBLE SOLUTION"
FOR J=1 TO NC
IF TABLE(NI,J)<0 THEN GOTO 1130
PRINT "***** MULTIPLE SOLUTIONS"
INPUT "Press <RETURN> to continue",DUM$
CHAIN "EFECT",ALL
NEXT J
INPUT "Press <RETURN> to continue",DUM$
CHAIN "RESULTS",ALL
FOR I=1 TO NI
FOR J=1 TO NJ
TABLE(I,J)=INTABLE(I,J)
NEXT J
NEXT I
RETURN
FILE NAME: EFFECT.BAS

THIS ROUTINE CALCULATES RELATIVE STATISTICAL COST EFFECTS FOR EACH OF THE VEHICLE CLASSES.

10 DIM EFFC(NVC),A1(NDIM+1,NVC),B(NDIM+1)
20 SUMEFC=0
30 INPUT "FRACTION OF THE TOTAL COST ATTRIBUTABLE TO THE ENVIRONMENT :", PENV
40 FOR JCL=1 TO NVC
50   EFFC(JCL)=0
60   FOR IRW=1 TO NDIM
70     IF INTABLE(IRW,JCL)=1 THEN EFFC(JCL)=EFFC(JCL)+INTABLE(IRW,NVC+2)
      ELSE EFFC(JCL)=EFFC(JCL)-INTABLE(IRW,NVC+2)
75     A1(IRW,JCL)=INTABLE(IRW,JCL)
80   NEXT IRW
85   A1(NDIM+1,JCL)=-A1(NDIM,JCL)
90   EFFC(JCL)=(EFFC(JCL)-PENV*INTABLE(NDIM,NVC+2))/2^(NVC-1)
100  SUMEFC=SUMEFC+EFFC(JCL)
110 NEXT JCL
120 FOR JCL=1 TO NVC
130   EFFC(JCL)=EFFC(JCL)/SUMEFC
140  PRINT EFFC(JCL),EFFC(JCL)*SUMEFC
150 NEXT JCL
160 FOR I=1 TO NDIM+1
170   B(I)=INTABLE(I,NVC+2)
180 NEXT I
190 FOR I=1 TO NR
200   IF IROW(I)=NVC+1 THEN IF R$(I)="X" THEN T=TABLE(I,NJ)
        : GOTO 250
210 NEXT I
220 T=0 : PRINT "***** NO BASIC SOLUTION FOR T *****"
250 CHAIN "NEWMAT",ALL
260 END
AN INITIAL LINEAR PROGRAMMING MATRIX FOR THE SECOND PHASE OF
THE GENERALIZED METHOD IS GENERATED IN THIS ROUTINE. THIS
ROUTINE IS EXECUTED ONLY WHEN THE FIRST PHASE YIELDS
MULTIPLE SOLUTIONS.

5 ERASE INTABLE, TABLE, R$, C$, IROW, JCOL
10 DIM INTABLE(2\*NVC+2*NVC+1, 3*NVC+1)
20 FOR IRW=1 TO 2\*NVC
30 FOR JCL=1 TO NVC
40 INTABLE(IRW, JCL)=A1(IRW, JCL)
45 NEXT JCL
50 IF IRW<2\*NVC-1 THEN INTABLE(IRW, 3*NVC+1)=B(IRW)-T
ELSE INTABLE(IRW, 3*NVC+1)=B(IRW)
60 NEXT IRW
100 FOR JCL=1 TO NVC
110 IRW=2\*JCL-1
120 IX=2\*NVC+IRW
130 JX=NVC+IRW
140 INTABLE(IX, JCL)=1/B(2\*NVC-1)
150 INTABLE(IX+1, JCL)=-1/B(2\*NVC-1)
160 INTABLE(IX, JX)=1
170 INTABLE(IX+1, JX)=-1
180 INTABLE(IX, JX+1)=-1
190 INTABLE(IX+1, JX+1)=1
200 INTABLE(IX, 3*NVC+1)=EFFC(JCL)
210 INTABLE(IX+1, 3*NVC+1)=-EFFC(JCL)
220 NEXT JCL
230 FOR J=NVC+1 TO 3*NVC
240 INTABLE(2\*NVC+2*NVC+1, J)=-1
250 NEXT J
260 FOR I=1 TO 2\*NVC+2*NVC+1
270 FOR J=1 TO 3*NVC+1
280 IF J<>3*NVC+1 THEN PRINT INTABLE(I, J);" ";
ELSE PRINT INTABLE(I, J)
290 NEXT J, I
300 DIM TABLE(2\*NVC+2*NVC+1, 3*NVC+1), IROW(2\*NVC+2*NVC+1),
JCOL(3*NVC+1)
310 DIM R$(2\*NVC+2*NVC+1), C$(3*NVC+1)
320 NR=2\*NVC+2*NVC : NC=3*NVC
330 NI=NR+1 : NJ=NC+1
340 CHAIN "SLP2", ALL
FILE NAME:  RESULTS.BAS

FINAL RESULTS OF THE GENERALIZED METHOD ARE OUTPUT USING THIS PROGRAM.

10 DIM ALLOC(NVC)
20 REM THIS PROGRAM PRINTS THE RESULTS OF THE
30 REM GENERALIZED COST ALLOCATION METHOD
40 '
50 ' LOOK FOR BASIC VARIABLES
60 '
70 FOR I=1 TO NR
80 IF R$(I)<"X" THEN GOTO 110
90 IF IROW(I)>NVC THEN GOTO 110
100 ALLOC(IROW(I))=TABLE(I,NJ)
110 NEXT I
120 '
130 ' PRINT COST ALLOCATIONS
140 '
145 LPRINT:LPRINT:LPRINT
150 PRINT CHR$(26):PRINT:PRINT:PRINT
160 FOR J=1 TO NVC
170 PRINT: LPRINT
180 PRINT "COST ALLOCATED TO VEHICLE CLASS";J;": ";
181 ALLOC(J)
181 LPRINT "COST ALLOCATED TO VEHICLE CLASS";J;": ";
182 ALLOC(J)
190 NEXT J
200 END
APPENDIX 7

SAMPLE RUN FOR THE GENERALIZED METHOD
Number of vehicle classes: 4

<table>
<thead>
<tr>
<th>COST FOR VEHICLE CLASSES 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COST FOR VEHICLE CLASSES 2</td>
<td>.76</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 2</td>
<td>1.11</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 3</td>
<td>1.87</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 3</td>
<td>2.04</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 2, 3</td>
<td>1.9</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 2, 3</td>
<td>2.06</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 4</td>
<td>1.18</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 4</td>
<td>1.46</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 2, 4</td>
<td>1.24</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 2, 4</td>
<td>1.51</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 3, 4</td>
<td>2.105</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 3, 4</td>
<td>2.22</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 2, 3, 4</td>
<td>2.13</td>
</tr>
<tr>
<td>COST FOR VEHICLE CLASSES 1, 2, 3, 4</td>
<td>2.24</td>
</tr>
</tbody>
</table>

COST ALLOCATED TO VEHICLE CLASS 1 : .41
COST ALLOCATED TO VEHICLE CLASS 2 : .32
COST ALLOCATED TO VEHICLE CLASS 3 : 1.03
COST ALLOCATED TO VEHICLE CLASS 4 : .48