TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Rehabilitation and Maintenance System State Optimal Fund Allocation - Program II (RAMS-SOFA-2)	January 1981
	6. Performing Organization Code
7. Author(s) Don T. Phillips, Chiyyarath V. Shanmugham, Farhad Ghasemi-Tari, Robert L. Lytton	8. Performing Organization Report No. Research Report 239-2
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, Texas 77843	10. Work Unit No.
	11. Contract or Grant No. Research Study 2-18-79-239
	13. Type of Report and Period Covered Interim - September, 1978 January, 1981
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation; Transportation Planning Division P. O. Box 5051 Austin, Texas 78763	14. Sponsoring Agency Code

15. Supplementary Notes

Research Study Title: Pavement Rehabilitation Fund A-location

16. Abstract

The Rehabilitation and Maintenance System - State Optimal Fund Allocation Program II (RAMS-SOFA-2) has been developed to aid the Texas State Department of Highways and Public Transportation in determining the optimal statewide strategy and fund allocation district by district. The program is specially designed to make decisions on rehabilitation and maintenance work on Interstate and spine networks on a statewide basis. The program uses a dynamic programming technique. The complete documentation on the program and an example problem are presented in this report. The model developed and the program can be utilized to determine the fund allocation to the residencies of an individual district.

17. Key Words	18. Distribution Statement
Pavements, Rehabilitation and Maintenance, Fund Allocation, Dynamic Programming	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	141	

Form DOT F 1700.7 (8-69)

REHABILITATION AND MAINTENANCE SYSTEM
STATE OPTIMAL FUND ALLOCATION - PROGRAM II
(RAMS-SOFA-2)

by

Don T. Phillips

Chiyyarath V. Shanmugham

Farhad Ghasemi-Tari

Robert L. Lytton

Research Report Number 239-2

Pavement Rehabilitation Fund Allocation

Research Project 2-18-79-239

conducted for
The Texas State Department of Highways and
Public Transportation

by the

Texas Transportation Institute
The Texas A&M University System

January 1981

ABSTRACT

The Rehabilitation And Maintenance System - State Optimal Fund Allocation Program II (RAMS-SOFA-2) has been developed to aid the Texas State Department of Highways and Public Transportation in determining the optimal statewide strategy and fund allocation district by district. The program is specially designed to make decisions on rehabilitation and maintenance work on Interstate and spine networks on a statewide basis. The program uses a dynamic programming technique. The complete documentation on the program and an example problem are presented in this report. The model developed and the program can be utilized to determine the fund allocation to the residencies of an individual district.

SUMMARY

This report describes the State Optimal Fund Allocation Program 2 of the RAMS (Rehabilitation And Maintenance System) family of computer programs developed by the Texas Transportation Institute to aid the Texas State Department of Highways and Public Transportation to optimally allocate rehabilitation and maintenance funds between the Districts. The report contains a detailed description of the mathematical model, an algorithm to solve the problem, a computer program based on the algorithm, and a user's manual.

An overview of all of the RAMS programs and how they are used sequentially in the fund allocation process is given in the first report of this series, Research Report 239-1, "Rehabilitation and Maintenance Systems: The Optimization Models."

The problem that is solved with the program described in this report is an integer nonlinear knapsack problem with multiple resource constraints. The algorithm developed uses dynamic programming methodology. A diverging branch dynamic programming model was developed for the problem. Each branch of the dynamic programming model is considered as a District, in which a set of maintenance strategies must be selected. The objective is to maximize the summation of calculated utilities subject to the limited resources of materials, equipment and manpower. The results obtained by solving each branch (District) are then used for allocation of the state-wide highway maintenance budget through maintenance Districts. That is, all the branches of the nonserial dynamic programming model are related to a single state variable.

(amount of the budget), while each branch of a serial dynamic programming problem with multi-dimensional state variables must be solved.

A computer program has been written based on the developed algorithm and has been tested on an example problem which has three Districts with 10 highway segments considered for rehabilitation or maintenance in each District. A user's guide and program listing is provided in the Appendix.

IMPLEMENTATION STATEMENT

The Texas Transportation Institute at Texas A&M University developed the Rehabilitation And Maintenance System - State Optimal Fund Allocation - Program II to help the Texas State Department of Highways and Public Transporation to determine and distribute optimally the rehabilitation and maintenance funds among the various districts in the State of Texas. The RAMS-SOFA-II is to be used in conjunction with the other programs in the RAMS family of programs. This report is intended as a working document which can be used by implementation workshops to train Texas SDHPT personnel in the use of RAMS-SOFA-II programs.

DISCLAIMER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

The allocation of funds for the rehabilitation and maintenance of the Texas State highway system required the use of a systematic approach to maximize return of the taxpayers dollars and to use all available resources most effectively. Until recently, the establishment of funding levels allocated to different highway segments have been based upon historical allocation formulas. Recently, a 0-1 integer linear programming model has been proposed for allocating the rehabilitation and maintenance funds to the highway segments in a network which is based upon needs and expected benefits, considering a one-year period of time (9). However, there is still a need for an appropriate model to project desirable funding levels for both single and multiple year rehabilitation and maintenance programs (10). A second major problem currently being approached is that of distributing the rehabilitation and maintenance funds among the 25 Districts within the State of Texas (8).

The strategic objective of the rehabilitation and maintenance of the Texas highway system is the selection of the optimal policy for each highway segment in a highway network in order to maximize the total effectiveness of all maintenance activities scheduled for the entire highway network in each year of a perpetual sequence of years. This is an optimization process, requiring a sequence of interrelated decisions within and between each funding period. Each single period can be considered individually as an optimization problem in which the objective is to find the most effective maintenance policy, subject to the existing manpower, equipment, materials, and cost limitations.

The single-period optimization is a "knapsack type" problem with multiple-resource constraints. An earlier attempt to find a solution to this single-period optimization problem was made by Ahmed (1) who formulated the problem as a 0-1 integer linear programming problem. As a result of this formulation, a large scale 0-1 problem was solved heuristically and a near-optimal solution was obtained.

An alternative approach to the single-period optimization is to formulate the problem as a "nonlinear knapsack problem" (NKP) which significantly reduces the number of variables and eliminates all the constraints except the resource constraints (8). A promising solution approach to handle NKP is discrete dynamic programming. Although this approach reduces the dimensionality of the decision variables, it suffers from the fact that the existence of more than three resource constraints renders this approach computationally intractable. This is the well-known problem of dimensionality of state variables in the dynamic programming technique. One way to reduce the M-state variable dynamic programming problem to a single-state variable problem is through the use of Lagrangian multipliers (3, 5). However, the problem of duality gaps, which is likely in the case of discrete variables, makes this approach somewhat dubious. An alternative method to reduce the dimensionality of the state variables is by employing the "imbedded state" technique (6). Although the comparative efficiency of both the Lagrangian and imbedded state approach is a questionable matter, and probably depends on the structure of the problem, the latter approach is reported to be relatively more efficient for NKP (7).

A fundamental advantage in using dynamic programming techniques for single period optimization is that it provides a bookkeeping record of returns for different maintenance funding levels which can be used for the overall distribution of funds, in an optimal manner, throughout different years in a given District. Development of a dynamic programming based model also provides the capability of distributing funds over all Districts in the State for a given budget cycle (8). Conversely, by using 0-1 integer programming for single-period optimization, the overall distribution of funding levels would be either impossible or a very difficult and time consuming task.

This report presents a model which is capable of distributing available rehabiliation and maintenance funds over a single period throughout all Districts in the State, and allocating other available resources within each District. The resulting model is also capable of optimally distributing limited funds through a finite time horizon for a single District or to allocate funds available in a single time period to different residencies within the District. When it is used for time-staging of projects, it is called RAMS-DTO-2 (District Time Optimization No. 2) and when it is used to optimally allocate funds between residencies for a single time period it is called RAMS-DO-2 (District Optimization, No. 2). A user's manual on RAMS-DTO-2 with changes in the program and the input data and results of an example problem will be presented in a forthcoming report.

When the same program is used to allocate funds available in a single time period between Districts, it is called RAMS-SOFA-2. (State ~~Optimal Fund Allocation Program No~~. ~~2)~~ ~~An overview of the RAMS (Rehabilitation~~ and Maintenance System) family of programs that have been developed by the Texas Transportation Institute to aid the Texas State Department of Highways and Public Transportation to optimally allocate rehabilitation and maintenance funds between and within the Districts is given in the first report of this series, Research Report 239-1, "Rehabilitation and Maintenance Systems: The Optimization Models." A new dynamic programming algorithm, capable of efficiently handling multiple constraints, will also be discussed.

CHAPTER 2

ANALYSIS OF THE HIGHWAY MAINTENANCE PROBLEM

Allocation of funds for highway maintenance operations is one of the basic components of the general rehabilitation and maintenance management system. In general, the basic components of a maintenance management system include maintenance standards, inventories of maintenance equipment, maintenance work loads, management information systems, and capital budgeting. The last component, capital budgeting, will likely become more stringently controlled in the future, and hence there exists a need for systematic, optimal allocation of these limited resources. Moreover, the use of an analytical technique for systematic allocation of available resources at the District level can identify maintenance practices that can potentially save money through more efficient utilization. Before describing the mathematical development of the problem, some useful terms should be defined:

a. Highway segment. A highway segment is a portion of a highway section or a combination of highway sections under consideration. This term is used to identify several highway sections which are similar or identical in traffic condition and environmental factors which affect the effectiveness of maintenance and rehabilitation activities.

b. Analysis period. The analysis period is a time duration greater than the expected life of any maintenance or rehabilitation method.

c. Types of distress. The usual categories of distress types for flexible pavements are: (1) rutting, (2) raveling, (3) flushing,

(4) corrugation, (5) roughness, (6) alligator cracking, (7) longitudinal cracking, (8) transverse cracking, and (9) patching.

d. Maintenance Strategy. A maintenance strategy is an activity selected for a highway segment in order to increase the pavement rating above a specified minimum requirement. Numerous strategies can be applied to each pavement segment. Among the more generally used strategies are the following: (1) strip seal, (2) fog seal, (3) seal coat, (4) light patching, (5) extensive patching and seal coating, (6) seal coat and planned thin overlay, (7) plant-mix seal or open-graded friction course, (8) thin overlay (less than two inches of asphalt concrete), (9) moderate overlay (two to three inches of asphalt concrete), (10) heavy overlay (three to six inches of asphalt concrete), and (11) reconstruction.

e. Pavement condition. The following criteria are used for determining current pavement condition: (1) the current pavement condition rating of each segment for each type of distress, (2) the potential gains of rating of each segment for each maintenance strategy and type of distress, (3) the pavement survival rate for each type of distress through a given time period for each type of pavement, (4) the minimum rating requirement of each segment for each type of distress over a specified time period.

f. Current pavement condition rating. The present condition of the pavement is determined by evaluation of the highway segment with respect to various types of distress. The rating score is obtained by subtracting the total "deduct values" associated with various types of distress from 100. Table 1 represents an example of deduct values for six types of distress for flexible pavement (2).

TABLE 1

Deduct Values of the Different

Distresses on Flexible Pavement

Type of Distress	Degree of Distress	Amount of Distress (1)	(2)	(3)
1. Rutting	Slight	0	2	5
	Moderate	5	7	10
	Severe	10	12	15
2. Ravelling	Slight	5	8	10
	Moderate	10	12	15
	Severe	15	18	20
3. Flushing	Slight	5	8	10
	Moderate	10	12	15
	Severe	15	18	20
4. Corrugation	Slight	5	8	10
	Moderate	10	12	15
	Severe	15	18	20
5. Alligator Cracking	Slight	5	10	15
	Moderate	10	15	20
	Severe	15	20	25
6. Patching	Slight	0	2	5
	Moderate	5	7	10
	Severe	7	15	20

g. Potential gain in rating. Potential gain in rating is defined as the net expected increase in pavement rating for each segment for each type of distress and maintenance strategy.

h. Pavement survival rate. When a maintenance strategy is applied to a highway segment, the pavement condition must achieve a high enough rating to survive for one year. Therefore, the survival probability immediately following a maintenance activity is one. As time passes, the pavement deteriorates and the survival probability of that segment is reduced.

i. Maintenance activity. Maintenance activity is a general term for the varous types of work that can be done to increase the rating score of a given pavement section.

j. Minimum rating requirement. The minimum rating requirement is used as an indication of when a maintenance activity must be scheduled. There are two such indicators: the first indicator results when the distress rating for any single type of distress falls below a minimum acceptable rating; the second indicator occurs when the total of all distress type ratings are less than the minimum total rating requirement.

k. Resource information. The restrictions on the availability of the resources usually appear as a set of constraints on the mathematical model. For strategic planning of pavement maintenance and rehabilitation, the resources are categorized in the following groups: (1) material and supply, (2) equipment, (3) personnel, and (4) District budget.

1. Management decisions. Management decisions determine: (1) the number of highway segments that will be considered for maintenance, (2) the types and number of maintenance strategies, (3) types and number of distress, (4) the planning horizon or analysis period, and (5) the amount of capital necessary to perform the required maintenance alternatives.

CHAPTER 3

FORMULATION OF THE MATHEMATICAL MODEL

The zero-one integer linear program formulation described by Phillips and Lytton (9) for optimal allocation of resources within a district is the basic model for pavement maintenance strategy planning. A detailed description of this model and a solution process is given in Ahmed (1) and Ahmed et al. (2). For a realistic problem, the model will involve a large number of zero-one decision variables and a very large number of constraints. For example, consider a District highway network with 200 highway segments, with ten maintenance alternatives per segment. Assume that there are 15 resource constraints and 5 different types of distress. With a planning horizon of 10 years as the period of analysis, the model will have approximately 2000 decision variables, 200 multiple choice constraints, 16 resource constraints including budget, 10,000 minimum rating constraints and 2,000 minimum overall rating constraints.

For the state-wide system optimization, a zero-one integer linear programming model whould be approximately 20 times larger.

3.1 A Dynamic Programming Model for Maintenance Strategic Planning for a Single District

An alternative approach, which will alleviate the difficulties identified in the previous section, is the development of a dynamic programming model for problem solution. Both the problem of allocating resources within a District and the problem of allocation of total budget between Districts can be handled by such a model. The decomposition of dynamic programming converts the larger problem into a sequence of smaller problems through which the process of achieving an optimal solution to the overall problem becomes easier. Further, the capability of obtaining the optimal solution values as a function of resource availability provides an inherent sensitivity analysis that can take into account the different budget levels at each District. As described in a subsequent section, the optimization process can be performed only once in each District, and the required information for distributing the budget to the different Districts can be achieved by solving a single dynamic programming problem.

This dynamic programming model for allocation of resources at the District level can be represented as a nonlinear knapsack problem. In general, a nonlinear knapsack model can be presented in this form shown below.

Problem A:

$$\text{Max. } f(\underset{\sim}{X}) = \sum_{j=1}^{N} r_j(\underset{\sim}{X}_j)$$

Subject to:

$$\sum_{j=1}^{N} a_{ij}(\underset{\sim}{X}_j) \leq b_i \qquad i = 1, 2, \ldots, M$$

X_j is contained in S_j.

$$S_j = (1, 2, \ldots, K_j) \qquad j = 1, 2, \ldots N$$

Problem A is a general form of the resource allocation problem. In the specific case of the highway maintenance allocation problem N is the number of highway segments in a district; K_j is the number of different types of maintenance strategies that can be applied to highway segment j; b_i is the available amount of resource type i; X_j is a decision variable taking values 1, 2, ..., K_j which indicates what strategy is being selected; a_{ij} is the amount of resource type i consumed by highway segment j if strategy X_j is being selected; $r_j\ (X_j)$ is the return benefit obtained from using maintenance strategy X_j on highway segment j, and M is the total number of limited resources.

The form of the model presented as Problem A can be expanded by defining its individual terms as follows:

Problem B

$$\text{Max.} \sum_{j=1}^{N} r_j(X_j) = \sum_{j=1}^{N} L_{1j}\, L_{2j} \left[\sum_{k=1}^{N_D} \sum_{t=1}^{N_T} D_{jk}(X_j)\, P_{jkt}(X_j) \right] \tag{3.1}$$

Subject to:

$$\sum_{j=1}^{N} S_{gj}(X_j)\, L_{1j}\, L_{2j} \leq TS_g \qquad g = 1, 2, \ldots, N_G \tag{3.2}$$

$$\sum_{j=1}^{N} E_{fj}(X_j)\, L_{1j}\, L_{2j} \leq TE_f \qquad f = 1, 2, \ldots, N_F \tag{3.3}$$

$$\sum_{j=1}^{N} H_{qj}(X_j)\, L_{1j}\, L_{2j} \leq TH_q \qquad q = 1, 2, \ldots, N_Q \tag{3.4}$$

$$\sum_{j=1}^{N} C_j(X_j)\, L_{1j}\, L_{2j} \leq TC \tag{3.5}$$

where the terms of the model are defined as follows:

C_j = the overhead cost function of strategy X_j at highway segment j.

E_{fj} = consumption of equipment type f at highway segment j.

H_{qj} = consumption of work force type q as a function of strategy X_j at highway segment j.

L_{1j} = the length of highway segment j.

L_{2j} = the width of highway segment j.

N = the member of highway segments.

N_D = the number of distress types.

N_F = the number of different types of equipment.

N_G = the number of different types of material.

N_Q = the number of different types of workforce.

N_T = the number of years in the analysis period.

P_{jkt} = pavement survival probability as a funtion of strategy X_j for the type of distress k at period t, in highway segment j.

S_{gj} = consumption of type g material per unit surface area as a function of strategy X_j at highway segment j.

TC = total budget available, in dollars.

TE_f = total amount of type f equipment available (equipment-day).

TH_q = total amount of q work force (human resource) available (person-day).

TS_g = total amount of type g material available.

In order to compare the dimensionality of the 0-1 integer linear program and dynamic programming, consider the example used earlier. It was stated that the zero-one integer linear program formulated for the problem involves approximately 2000 decision variables and 12,216 constraints. The nonlinear knapsack model for the same problem involves only 200 decision variables and only 16 inequality constraints. This illustrates a significant reduction of 1800 decision variables and 12,200 inequality constraints. It must be recognized that the decision variables in the later model take on 10 different values. However, the use of a proper solution technique, dynamic programming, will yield an efficient solution to the problem.

3.2 A Dynamic Programming Model for Maintenance Strategic Planning Considering Multiple-Districts

The models discussed in Section 3.1 have been developed to allocate funds within a specific District. In this section, the problem of state-wide fund allocation will be discussed. In particular, a model will be developed to allocate the state-wide budget to the Districts and at the same time, to allocate resources within individual Districts. The task of projecting the required budget levels for the annual maintenance program is also considered in this model.

Consider the nonlinear knapsack model discussed as Problem A. This formulation is a general representation of the allocation of resources within a highway District. The availability of each resource, such as equipment, materials, and manpower is determined by the District engineer and usually has fixed values. However, the amount of funds available to the Texas State Department of Highways and Public Transportation is determined by the State legislature. Recently, a systematic method for allocating the statewide budget to Districts has been proposed by Phillips and Lytton (8). The proposed method provides a range of budget allocations to each District. An optimal maintenance policy is determined for the selected number of possible budget levels within each District. The overall maintenance policy at this state level is then determined through an overall synthesis and optimization model based upon dynamic programming.

In order to be certain that the Interstate and other spine networks are maintained at acceptable levels of service in all Districts, a more desirable approach to the problem of allocating the statewide spine network budget to individual Districts is to develop a model capable of handling both the within and between District allocation process at the same time. The mathematical representation of such a model in the form of a nonlinear knapsack problem is presented below.

Problem C

$$\text{Max.} \quad \sum_{d=1}^{D} \sum_{j=1}^{N} r_{jd}(X_j)$$

Subject to:

$$\sum_{j=1}^{N} a_{ijd}(X_j) \leq b_{id} \qquad i = 1, 2, \ldots, M-1 \quad d = 1, 2, \ldots, D$$

$$\sum_{d=1}^{D} \sum_{j=1}^{N} C_{jd}(X_j) \leq TC$$

$$X_j \text{ is contained in } S_{jd} \qquad d = 1, 2, \ldots, D \quad j = 1, 2, \ldots, N$$

$$S_{jd} = (1, 2, \ldots, K_{jd})$$

where

a_{ijd} = the amount of resource type i (excluding overhead cost) consumed as a function of strategy X_j, for highway segment j at District d.

b_{id} = total amount of type i available resource (excluding budget level) at District d.

C_{jd} = the amount of consumption costs which is a function of the strategy X_j, for highway segment j in District d.

D = the number of Districts in the analysis.

K_{jd} = the number of maintenance strategies that can be applied to highway segment j in District d.

M = the number of resource constraints excluding costs.

r_{jd} = the return function of strategy X_j, for highway segment j, in District d.

TC = total amount of available budget for entire state.

x_j = the decision variable indicating the type of strategy to be selected.

Problem C can be decomposed into two separate problems. The first is a decomposition of the problem according to Districts. Each District can then be considered as a single state in a statewide dynamic programming formulation. The second problem is a decomposition of all District subproblems into individual highway segments which yields a problem similar to Problem A. This process can be illustrated by expanding Problem C.

Problem D

$$\text{Max. } \sum_{j=1}^{N} r_{j1}(X_j) + \sum_{j=1}^{N} r_{j2}(X_j) + \ldots + \sum_{j=1}^{N} r_{jD}(X_j)$$

Subject to:

$$
\begin{array}{lllll}
\sum_{j=1}^{N} a_{ij1}(X_j) & & & & \leq \underset{\sim}{b}_{i1} \\
 & \sum_{j=1}^{N} a_{ij2}(X_j) & & & \leq \underset{\sim}{b}_{i2} \\
 & & \ddots & & \vdots \\
 & & & \sum_{j=1}^{N} a_{ijD}(X_j) & \leq \underset{\sim}{b}_{iD}
\end{array}
$$

for $i = 1, 2, \ldots, M-1$

$$\sum_{j=1}^{N} C_{j1}(X_j) + \sum_{j=1}^{N} C_{j2}(X_j) + \sum_{j=1}^{N} C_{jD}(X_j) \leq TC$$

Referring to Problem D, the limitations on all the resources are considered independently for each District with the exception of the limitation on the budget level (TC) which interrelates the decisions in all Districts. However, the allocation process within each District could be developed independently if it were developed as a function of budget level in that District. That is, a vector presenting the optimal return as a function of budget level in each District could be obtained. These District benefits and associated cost levels could be used for the allocation of the total budget to individual Districts.

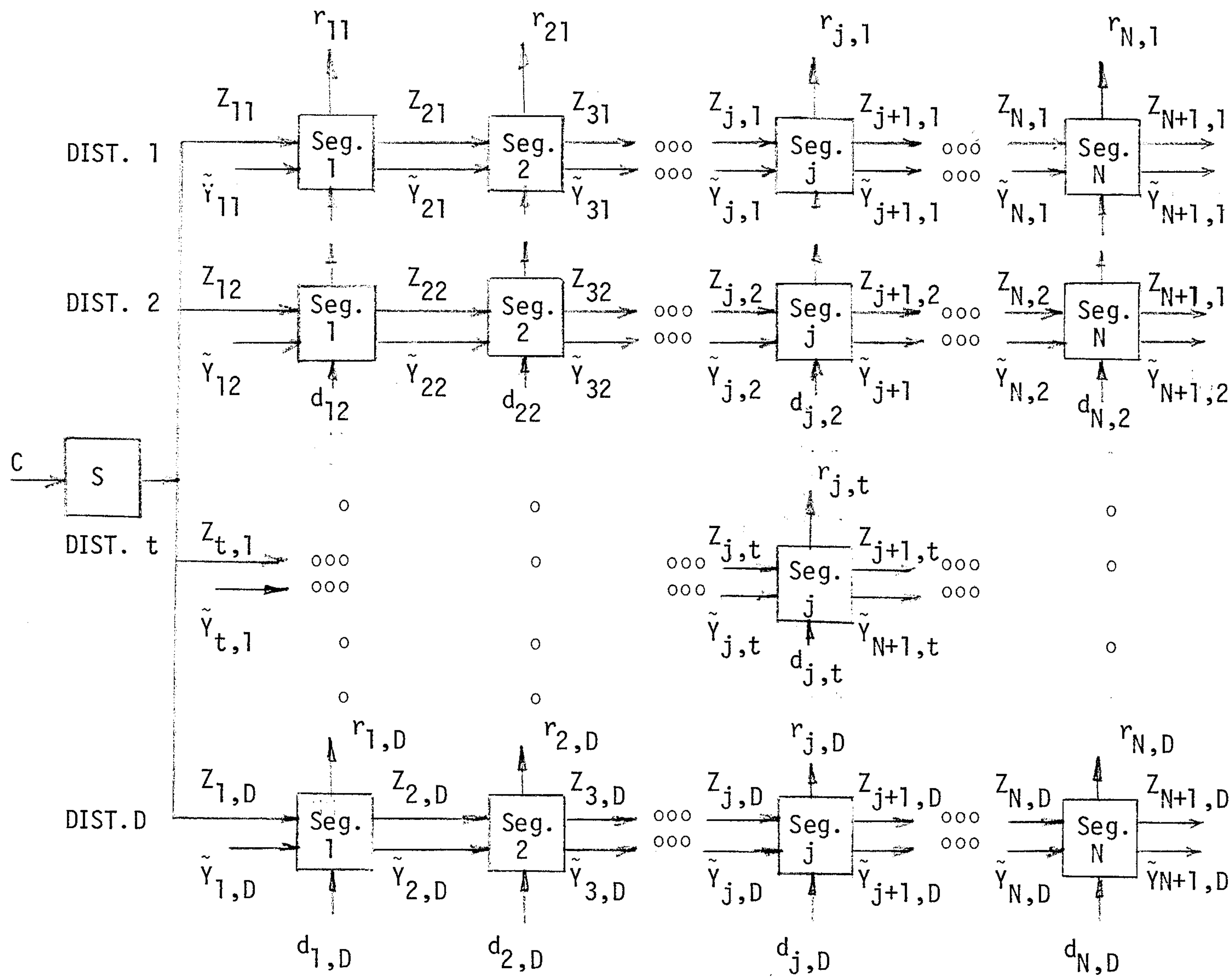

Figure 1. Schematic Representation of the Allocation of Funds Within and Between Districts

This two-level allocation process can be suitably performed using a non-serial dynamic programming model. This model is illustrated schematically in Figure 1. In this figure, each branch represents the allocation of resources within a District, and the node S from which each branch diverges represents the allocation of the total budget to each individual District.

In each branch there will be N stages representing the number of segments in that branch (Districts), and D branches diverging from Node S. Each branch may be solved as an initial-value problem in terms of Z_{jt}. This is accomplished using forward recursion carrying Z_{jt} as an extra state variable. At the final stage the return vector, which is a function of the state variables, will be obtained for each branch. The state variables represent the consumption of the resources such as types of equipment, materials, personnel, and the total budget level. Among these state variables, only the consumption of the budget is the subject of further optimization and all other state variable inputs are fixed. As a result, the returns in each District, as a function of budget level, are obtained. Considering each District as a single stage in the dynamic programming model, a decision must be made with regard to the allocation of budget levels to each District in order to obtain the maximum return.

Referring to Figure 1, it can be seen that each branch involves a multiple-constraint dynamic programming problem. These constraints are divided into two groups. The first group is represented by a state vector y_{jd}. The second group, rehabilitation cost, is represented by a single-state variable, Z_{jd}. This separation has just been justified; i.e., the cost constraint interrelates the decision-making process between the different branches, which enables the group of constraints represented by y_{it} to be considered independently within each branch.

Consider District d; allocation of resources to this District using a dynamic programming technique results in the following recursive equations:

$$R_{d1}\ (Z_{1d}, \tilde{Y}_{1d}) = \text{Max.}\ r_{1d}(X_j)$$

over

$$0 \leq \tilde{A}_{1d}(X_j) \leq \tilde{Y}_{1d} \qquad \text{for a fixed } d$$

$$0 \leq C_{11}(X_j) \leq Z_{1d}$$

$$R_{dj}(Z_{jd}, \tilde{Y}_{jd}) = \text{Max.}\ \ r_{jd}(Z_j) + \text{Max.}\ \ R_{j-1,d}(Z_{j=1,t}, \tilde{Y}_{j-1,d})$$

$$\text{for } j = 2, 3, \ldots, N$$

over

$$0 \leq \tilde{A}_{jd}(X_j) \leq \tilde{Y}_{1d} \qquad \text{for } j = 2, 3, \ldots, N$$

The state recursion equations are:

$$Z_{1t} = TC$$

$$\tilde{Y}_{1t} = \tilde{TR}$$

$$Z_{j-1,t} = Z_{jt} - C_{jt}(X_j) \qquad j = 2, 3, \ldots, N$$

$$\tilde{Y}_{j-1,t} = \tilde{Y}_{jt} - \tilde{A}_{jt}(X_j) \qquad j = 2, 3, \ldots, N$$

where the state variables are defined as

Z_{jt} = the amount of budget available for stages (segments) j, j + 1, ..., N.

$\tilde{Y}_{jt}$ = the vector whose components represent the amount of each type of resource available for stages (segments) j, j + 1, ..., N.

and

$$\tilde{TR} = (b_{1t}, b_{2t}, \ldots, b_{M-1,t})$$

The recursive equations developed for District t can be applied to all Districts, i.e., $d = 1, 2, \ldots, D$. After a dynamic programming solution procedure is applied to each District, the return $R_{Nd}\ (Z_{Nd}, \tilde{Y}_{Nd})$ will be obtained. Since the first group of constraints is not involved in the allocation of budget to Districts, let $R_d\ (Z_d) = R_{Nd}\ (Z_{Nd}, \tilde{Y}_{Nd})$. The distribution of budget levels to each District is then obtained by solving the following problem:

Problem E:

$$\text{Max.} \quad \sum_{d=1}^{D} R_d^1(Z_d^1)$$

Subject to:

$$\sum_{d=1}^{D} Z_d^1 = TC$$

$$Z_d \varepsilon S_d^1$$

where

S_d^1 = the vector of the budget levels in the final stage of branch d (District d).

R_d^1 = the return vector obtained in the final stage of branch d (District d).

Problem E is a one-dimensional (single linking constraint) nonlinear knapsack model which can also be solved with dynamic programming techniques. The optimal solution resulting from solving Problem E will define the optimal budget level, Z_d^1*, and after obtaining this value, the optimal set of maintenance policies for every segment of each District can be recovered.

CHAPTER 4

OPTIMIZATION OF THE MODEL

This chapter presents the development of the required algorithm for solving separable nonlinear, multi-dimensional knapsack problems. The algorithm is called a "hybrid algorithm", and it is essentially a dynamic programmming approach in the sense that the problem is divided into smaller subproblems. However, the idea of fathoming the partial solution by branch and bound is incorporated within the algorithm. The main feature of the hybrid algorithm is its capability of reducing the state-space which otherwise would present an obstacle in solving multiple-constraint dynamic programming problems. Part of this reduction is due to the use of the imbedded-state approach, which reduces an M-dimensional dynamic program to a one-dimensional problem. Other reductions are made through fathoming the state-space and subsequent elimination of state-space regions, which tend to eliminate inferior solutions when compared to the predetermined lower bound or updated lower bound.

The use of a surrogate constraint methodology is implemented in the algorithm to obtain initial lower and upper bounds for the objective function. At each stage, the lower and upper bounds are also updated by use of a surrogated problem, and the updated upper bound is used for termination criteria. The procedure for updating lower and upper bounds in the surrogated problem is very efficient. In addition, the primary advantages of using the surrogate problem to estimate these bounds, are (1) it provides a narrow range between the lower and upper bound, and (2) it might provide the optimal solution to the problem at the first step.

A modification of the hybrid algorithm has been developed for application to large scale NKP's. However, the modified algorithm, though computationally much faster, may not provide an optimal solution to some problems, but rather will obtain a near-optimal solution. The modified algorithm follows roughly the same procedure as the hybrid algorithm. However, instead of evaluating all promising solution spaces, it attempts only to improve the lower bound calculated by the surrogated problem.

The details of the hybrid and the modified hybrid algorithm are presented in Appendix A. The documentation and the user's guide to the computer programs are given in Appendices B and C. The algorithms will now be explained by use of a simple example.

4.1: The Imbedded State Approach

Consider a District highway network problem, with 4 highway segments. There are 5 maintenance strategies available per segment. The two constraints deal with the budget and one type of resource. Therefore:

$$N = 4$$

$$K_j = 5$$

$$M = 2$$

Considering the model presented in Problem A, X_j is contained in S_j i.e.

$$X_j \varepsilon S_j = (1, 2, 3, 4, 5)$$
$$\text{for } j = 1, 2, 3, 4.$$

The objective function coefficients and the constraints are given in Table II and Table III respectively. The right hand sides (availability of resources) are 28 in each constraint.

TABLE II

$R_j(X_j)$ - OBJECTIVE FUNCTION COEFFICIENTS

	$X_j \ \varepsilon \ S_j$				
j	1	2	3	4	5
1	0	2	3	5	8
2	0	3	4	5	6
3	0	6	9	11	13
4	0	4	7	10	11

*j is the index on the highway segments.

TABLE III

$A_{ij}\ (X_j)$ - CONSTRAINT COEFFICIENT

		$X_j \ \varepsilon \ S_j$				
i	j	1	2	3	4	5
1	1	0	6	8	9	11
2	1	0	3	4	5	7
1	2	0	7	10	12	14
2	2	0	4	6	8	10
1	3	0	8	10	12	15
2	3	0	6	8	9	12
1	4	0	5	6	9	10
2	4	0	4	8	12	5

*i = the index on the constraints.
j = the index on the highway segments.

Solution:

Stage 1 Calculations

$j = 1\ ,\quad K = K_1 = 5$

The imbedded-state space for stage 1 is

$$G_1 = \left\{ (0,0),\ (6,3),\ (8,4),\ (9,5),\ (11,7) \right\}$$

$$F_1 = G_1$$

$$r_1 = (0,\ 2,\ 3,\ 5,\ 8)$$

The T_1 and TS_1, matrices are created using elements of F_1 and r_1.

T_1 TS_1

Row	X_1	Pointer to Stage 0	g_1		r_1
1	1	--	0	0	0
2	2	--	6	3	2
3	3	--	8	4	3
4	4	--	9	5	5
5	5	--	11	7	8

Since each element of F_1 satifies the feasibility conditions,

$$(0,0) < (28,\ 28),$$
$$(6,3) < (28,\ 28),$$
$$(8,4) < (28,\ 28),$$
$$(9,5) < (28,\ 28),$$
$$(11,7) < (28,\ 28),$$

none of these solutions are eliminated and hence

$$F_1^{\,f} = F_1$$

and

$$F_1^{\,e} = F_1^{\,f}$$

The updated version of TS_1, and T_1 will be the same as the ones before since no points are eliminated.

Stage 2 Calculations

$J = 2,\quad K = K_2 = 5$

$$G_2 = \left\{(0,0)\ (7,4),\ (10,6),\ (12,8),\ (14,10)\right\}$$

$$F_2 = G_2 \circ F_1^{\,e}$$

The number of elements in F_2 is the product of K_2 and the elements in $F_1^{\,e}$.

$$F_2 = \{(0,0),\ (6,3),\ (8,4),\ (9,5),\ (11,7),\ (7,4),\ (13,7)\ (15,8),$$
$$(16,9),\ 18,11),\ (10,6),\ (16,9),\ (18,11),\ (10,6),\ (16,9),,$$
$$(18,11),\ (10,6),\ (16,9)\ (18,10),\ (19,11),\ (21,13),\ (12,8),$$
$$(18,11),\ (20,12),\ (21,13),\ 23,15),\ (14,10),\ (20,13),\ (22,12),$$
$$(23,15),\ (25,17)\}.$$

The T_2 and TS_2 matrices are generated as follows.

T_2 TS_2

Row	X_2	Pointer to Stage 1	g_2		r_2
1	1	1	0	0	0
2	1	2	6	3	2
3	1	3	8	4	3
4	1	4	9	5	5
5	1	5	11	7	8
6	2	1	7	4	3
7	2	2	13	7	5
8	2	3	15	8	6
9	2	4	16	9	8
10	2	5	18	11	11
11	3	1	10	6	4
12	3	2	16	9	6
13	3	3	18	10	7
14	3	4	19	11	9
15	3	5	21	13	12
16	4	1	12	8	5
17	4	2	18	11	7
18	4	3	20	12	8
19	4	4	21	13	10
20	4	5	23	15	13
21	5	1	14	10	6
22	5	2	20	13	8
23	5	3	22	12	9
24	5	4	23	15	11
25	5	5	25	17	14

None of the elements of F_2 violates the feasibility conditions, and hence

$$F_2^{\,f} = F_2 \quad .$$

However, some of the solutions can be eliminated by the dominancy test. Consider the solutions presented in rows 3 and 6. Both solutions result in the same return (3 units). The solution in row 3 requires 8 units of resource 1 and 4 units of resource 2. Solution in row 6 requires 7 units of resource 1 and 4 units of resource 2. Solution (7,4) dominates solution (8,4). Therefore, solution (8,4) is eliminated from the set. Consider solutions which generate a return of 5 units. They are: (9,5), (13,7), and (12,8). Obviously, (9,5) dominates (13,7) and (12,8), and hence (13,7) and (12,8) are eliminated from the solution set. Compare solutions (9,5) and (10,6) resulting in returns of 5 and 4 units respectively. Solution (10,6) consumes a higher amount of both the resources, but generates a lower return only. Solution (10,6) is eliminated from the set. Conducting the dominancy test on all the elements of $F_2^{\,f}$, a total of 16 solutions are eliminated and a nine element vector F_2^e is generated.

$$F_2^{\,e} = \Big\{ (0,0),\ (6,3),\ (7,4),\ (9,5),\ (11,7),\ (18,11),\ (21,13),\ (23,15),\ (25,17) \Big\} .$$

Thus the updated TS_2 and T_2 matrices are as follows

	T_2		TS_2		
Row	X_2	Pointer to Stage 1	g_2		r_2
1	1	1	0	0	0
2	1	2	6	3	2
3	2	1	7	4	3
4	1	4	9	5	5
5	1	5	11	7	8
6	2	5	18	11	11
7	3	5	21	13	12
8	4	5	23	15	13
9	5	5	25	17	14

Stage 3 Calculations

$$J = 3, \quad K = K_3 = 5$$

$$G_3 = \left\{(0,0),\ (8,6),\ (10,8),\ (12,9),\ (15,12)\right\}$$

and

$$F_3 = G_3 \text{ o } F_2^{\ e}$$

Since G_3 and $F_2^{\ e}$ have 5 and 9 elements respectively, F_3 will have 45 elements. The elements of F_3 with the associated returns are listed in the TS_3 matrix.

T_3 TS_3

Row	X_3	Pointer to Stage 2	g_3		r_3
1	1	1	0	0	0
2	1	2	6	3	2
3	1	3	7	4	3
4	1	4	9	5	5
5	1	5	11	7	8
6	1	6	18	11	11
7	1	7	21	13	12
8	1	8	23	15	13
9	1	9	25	17	14
10	2	1	8	6	6
11	2	2	14	9	8
12	2	3	15	10	9
13	2	4	17	11	11
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
33	4	6	30	20	22
34	4	7	33	22	23
35	4	8	35	24	24
36	4	9	37	26	25
37	5	1	15	12	13
38	5	2	21	15	15
39	5	3	22	16	16
40	5	4	24	17	18
41	5	5	26	19	21
42	5	6	33	23	24
43	5	7	36	25	25
44	5	8	38	27	26
45	5	9	40	29	27

In F_3, all the solutions with resource requirements higher than (28,28) are eliminated. For example, solution (30,20) in row 33 will be eliminated since the resource 1 requirement, 30, exceeds the availibility, 28. For solution (40,29) in row 45, the requirements exceed the availabilities for both the resources. Hence, it will be eliminated. A total of 14 infeasible solutions are eliminated. Further, the dominancy test will eliminate 17 more solutions. The resulting F_3^e with 14 elements and r_3 are shown in the updated T_3 and TS_3 matrices.

T_3 TS_3

Row i	X_3	Pointer to Stage 2	g_3		r_3
1	1	1	0	0	0
2	1	2	6	3	2
3	1	3	7	4	3
4	1	4	9	5	5
5	2	1	8	6	6
6	1	5	11	7	8
7	3	1	10	8	9
8	4	1	12	9	11
9	5	1	15	12	13
10	3	4	19	13	14
11	4	4	21	14	16
12	3	5	21	15	17
13	4	5	23	16	19
14	5	5	26	19	21

Stage 4 Calculations

$$J = 4,\ K = K_4 = 5$$

$$G_4 = \left\{(0,0),\ (5,4),\ (6,8),\ (9,12),\ (10,15)\right\}$$

and

$F_4 = G_4 \circ\ F_3^{\ell}$ contains 70 elements of which 37 elements are eliminated by feasibility and dominancy tests. The resulting F_4^e will have 33 solutions and are shown in the following T_4 and TS_4 matrices.

T_4			TS_4		
Row i	X_4	Points to Stage 3	g_4		r_3
1	1	1	0	0	0
2	1	2	6	3	2
3	2	1	5	4	4
4	1	4	9	5	5
5	1	5	8	6	6
6	3	1	6	8	7
7	2	3	12	6	7
8	1	6	11	7	8
9	1	7	10	8	9
10	4	1	9	12	10
11	5	1	10	15	11
12	1	8	12	9	11
13	2	6	10	11	12
14	1	9	15	12	13
15	3	5	14	14	13
16	1	10	19	13	14
17	2	8	17	13	15
18	1	11	21	14	16
19	2	7	16	16	16
20	1	12	21	15	17
21	2	9	20	16	17
22	3	8	18	17	18
23	1	13	23	16	19

T_4 TS_4

Row i	X_4	Pointer to Step 3	g_4		r_4
24	4	7	19	20	19
25	3	9	21	18	20
26	5	7	20	23	20
27	1	14	26	19	21
28	4	8	21	21	21
29	5	8	22	24	22
30	2	13	28	20	23
31	3	11	27	22	23
32	4	9	24	24	23
33	3	12	27	23	24

At the end of stage 4 calculations, from column 3 of TS_4 matrix, it is found that the maximum (optimum) return is 24.

$$R_4^* = 24$$

The optimal decision values at the various stages are determined by back-tracking through the T-matrices. The optimum value 24 is generated by the solution (27,23), in row 33. The optimal value of X_4 is 3 and the pointer to stage 3 indicates that optimal value of X_3^* is in the 12th row of T_3 matrix. $X_3^* = 3$ and the pointer to stage 2 indicates that optimal value of X_2 is in the 5 th row of T_2 matrix . $X_2^* = 1$ and the pointer to stage 1 indicates that optimal value of X_1 is in row 5 of T_1 matrix $X_1^* = 5$. The backtracking operation is summarized in the following table.

Stage j	X_j^*	Pointed to Stage j-1	$A_{1j}(X_j)$ $A_{2j}(X_j)$		$r_j(X_j)$
4	3	12	6	8	7
3	3	5	10	8	9
2	1	5	0	0	0
1	5	-	11	7	8
			$g_4 = (27,23)$		$R_4^* = 24$

4.2 The Hybrid Algorithm

The solution of the same example problem will now be obtained using the hybrid algorithm. The algorithm requires an initial lower and upper bounds to the objective which is obtained by the surrogate constraint methodology.

The surrogate problem of Problem A is as follows:

Problem E

$$\text{Max} \sum_{j=1}^{N} r_j\,(X_j)$$

subject to

$$\sum_{j=1}^{N} \sum_{i=1}^{M} \alpha_i\, A_{ij}\,(X_j) \leq \sum_{i=1}^{M} \alpha_i\, b_i$$

$$\sum_{i=1}^{n} \alpha_i = 1$$

$$\alpha_i > 0,\ i = 1, 2, \ldots, M$$

$$X_j \in S_j = \left[1, 2, \ldots, k_j\right]$$

$$j = 1, 2, \ldots, N$$

Let

$$\alpha_i = \frac{1}{M}$$

$$A_j(X_j) = \sum_{i=1}^{M} \alpha_i \, A_{ij} \, (X_j)$$

and $$B = \sum_{i=1}^{M} \alpha_i \, b_i$$

where

$\left[I\right]$ defines the largest integer value less than or equal to I.

Then, the surrogate problem is:

$$\max \sum_{j=1}^{N} A_j(X_j) \leq B$$

$$X_j \in S_j = \left[1, 2, \ldots, k_j\right]$$

$$j = 1, 2, \ldots, N.$$

For the example problem A_j and r_j are as follows:

j	$A_j(X_j)$	$r_j(X_j)$
1	(0, 4, 6, 7, 9)	(0, 2, 3, 5, 8)
2	(0, 5, 8, 10, 12)	(0, 2, 4, 5, 6)
3	(0, 7, 9, 10, 13)	(0, 6, 9, 11, 13)
4	(0, 4, 7, 10, 12)	(0, 4, 7, 10, 11)

The solution to the surrogate problem is obtained using the dynamic programming techniques in tabular form (7).

Stage 1 Calculations

Table of Returns for Stage 1

X_1 / A_1 / r_1 / B	1 / 0 / 0	2 / 4 / 2	3 / 6 / 3	4 / 7 / 5	5 / 9 / 8	R_1^*	X_1^*
0-3	0	-	-	-	-	0	1
4-5	0	2	-	-	-	2	2
6	0	2	3	-	-	3	3
7-8	0	2	3	5	-	5	4
9-28	0	2	3	5	8	8	5

Stage 2 Calculations

Table of Cumulative Returns for
Stages 2 and 1

B \ X_2	1	2	3	4	5	R_2*	X_2*
A_2	0	5	8	10	12		
r_2	0	3	4	5	6		
0 - 3	0	-	-	-	-	0	1
4	2	-	-	-	-	2	1
5	2	3	-	-	-	3	2
6	3	3	-	-	-	3	1,2
7	5	3	-	-	-	5	1
8	5	3	4	-	-	5	1
9	8	5	4	-	-	8	1
10	8	5	4	5	-	8	1
11	6	6	4	5	-	8	1
12 - 13	8	8	6	5	6	8	1,2
14	8	11	7	7	6	11	2
15	8	11	9	7	6	11	2
16	8	11	9	8	8	11	2
17	8	11	12	10	8	12	3
18	8	11	12	10	9	12	3
19 - 20	8	11	12	13	11	13	4
21 - 28	8	11	12	13	11	13	4

Stage 3 Calculations

Table of Cumulative Returns for Stages 3, 2, 1

B \ X_3 / A_3 / r_3	1 / 0 / 0	2 / 7 / 6	3 / 9 / 9	4 / 10 / 11	5 / 13 / 13	R_3*	X_3*
0 - 3	0	-	-	-	-	0	1
4	2	-	-	-	-	2	1
5	3	-	-	-	-	3	1
6	3	-	-	-	-	3	1
7	5	6	-	-	-	6	2
8	5	6	-	-	-	6	2
9	8	6	9	-	-	9	3
10	8	6	9	11	-	11	4
11	8	8	9	11	-	11	4
12	8	9	9	11	-	11	4
13	8	9	11	11	13	13	5
14	11	11	12	13	13	13	4,5
15	11	11	12	14	13	14	4
16	11	14	14	14	13	14	2,3,4
17	12	14	14	16	15	16	4
18	12	14	17	16	16	17	3
19	13	14	17	19	16	19	4
20	13	14	17	19	18	19	4
21	13	17	17	19	18	19	4
22	13	17	17	19	21	21	5
23	13	17	20	19	21	21	5
24	13	18	20	22	21	22	4
25	13	18	20	22	21	22	4
26	13	19	21	22	21	22	4
27	13	19	21	23	24	24	5
28	13	19	21	23	24	24	5

Stage 4 Calculations

Table of Cumulative Returns for Stages 4, 3, 2, and 1

B / X_4 / A_4 / r_4	1 / 0 / 0	2 / 4 / 4	3 / 7 / 7	4 / 10 / 10	5 / 12 / 11	R_4*	X_4*
0 - 3	0	-	-	-	-	0	1
4	2	4	-	-	-	4	2
5	3	4	-	-	-	4	2
6	3	4	-	-	-	4	2
7	6	4	7	-	-	7	3
8	6	4	7	-	-	7	3
9	9	7	7	-	-	9	1
10	11	7	7	10	-	11	1
11	11	10	9	10	-	11	1
12	11	10	10	10	11	11	1,5
13	13	13	10	10	11	13	1,2
14	13	15	13	12	11	15	2
15	14	15	13	13	11	15	2
16	14	15	16	13	13	16	3
17	16	17	18	16	14	18	3
18	17	17	18	16	14	18	3
19	19	18	18	19	17	19	1,4
20	19	18	20	21	17	21	4
21	19	20	20	21	20	21	4
22	21	21	21	21	22	22	5
23	21	23	21	23	22	23	2,4
24	22	23	23	23	22	23	2,3,4
25	22	23	24	24	24	24	3,4,5
26	22	25	26	24	24	26	3
27	24	25	26	26	25	26	3,4
28	24	26	26	27	25	27	4

The maximum value of the returns is 27 when B equals 28. The corresponding optimal decisions for the various stages are obtained by tracing back through the stage calculations and are illustrated below:

Stage j	B	X_j^*	A_j	r_j^*
4	28	4	10	10
3	18	3	9	9
2	9	1	0	0
1	9	5	9	8

$$\text{Therefore } X^* = (X_1^*, X_2^*, X_3^*, X_4^*)$$

$$= (5, 1, 3, 4)$$

Similarly, the optimal decisions for various values of B can be determined by tracing back through the stage calculations. The solution of the surrogate problem for various values of B are shown in Table IV. In Table IV, the solution for B equal to 27 is eliminated since B = 26 and B = 27 generate the same objective function value, R_{sp} (27) = R_{sp} (26) = 26.

In addition, it should be noted that certain values of B generate alternate optimum solutions; e.g. when B = 25, there are three alternate optimal decisions.

The initial lower and upper bounds to be used in the hybrid algorithm are determined as follows:

$$UB_o = R_{sp}\ (28) = 27.$$

TABLE IV. SOLUTION OF THE SURROGATE PROBLEM AS A FUNCTION OF RIGHT HAND SIDE

State (B)	Return R(B)	Optimal decisions X
0	0	1 1 1 1
4	4	1 1 1 2
7	7	1 1 1 3
9	9	1 1 3 1
10	11	1 1 4 1
13	13	1 1 5 1
13	13	1 1 3 2
14	15	1 1 4 2
16	16	1 1 3 3
17	18	1 1 4 3
19	19	5 1 4 1
19	19	1 1 3 4
20	21	1 1 4 4
22	22	1 1 4 5
23	23	5 1 4 2
25	24	5 1 3 3
25	24	1 1 5 5
25	24	1 2 4 4
26	26	5 1 4 3
28	27	5 1 3 4

The lower bound LB_0 is the largest optimal return value of the surrogate problem with the corresponding optimal decisions being feasible to the original problem.

Consider $B = 28$; $X^* = (5,1,3,4)$

Since

$$A_{11}(5) + A_{12}(1) + A_{13}(3) + A_{14}(4)$$

$$= 11 + 0 + 10 + 9 = 30 > 28 = b_1,$$

the first constraint is violated and $X^* = (5,1,3,4)$ is an infeasible solution.

Let $B = 26$; $X^* = (5,1,4,3)$

$$A_{11}(5) + A_{12}(1) + A_{13}(4) + A_{14}(3)$$

$$= 11 + 0 + 12 + 6 = 29 > 28 = b_1$$

implies that $X^* = (5,1,4,3)$ is also infeasible.

Consider $B = 25$; $X^* = (1,2,4,4)$

Since $A_{11}(1) + A_{12}(2) + A_{13}(4) + A_{14}(4)$

$$= 0 + 7 + 12 + 9 = 28 = b_1,$$

X^* satisfies first constraint.

$$A_{21}(1) + A_{22}(2) + A_{23}(4) + A_{24}(4)$$

$$= 0 + 4 + 9 + 12 = 25 < 28 = b_1$$

implies second constraint also is satisfied. $X^* = (1,2,4,4)$ is a feasible solution to the original problem.

Therefore

$$LB_0 = R_{sp}\ (25) = 24.$$

Solution of the Example Problem:

Stage 1 Calculations

$G_1 = \quad (0, 0), (6,3), (8,4), (9,5), (11,7)$

$F_1{}^e = F_1{}^f = G_1$

$R_1 = (0, 2, 3, 5, 8)$

No points are eliminated by the feasibility and dominancy tests. The T_1 and TS_1 matrices are as follows.

T_1 $\qquad\qquad\qquad\qquad$ TS_1

Row	X_1	Pointer to Stage 0	$g^i{}_1$		$R_1(g^i{}_1)$
1	1	--	0	0	0
2	2	--	6	3	2
3	3	--	8	4	3
4	4	--	9	5	5
5	5	--	11	7	8

Let $i = 1$ $\qquad\qquad$ $g^i{}_1 = (0,0)$

$$UB1_1{}^1 = R_1\ (g_1{}^1) + \sum_{t=2}^{4} r_t\ (k_t)$$

$$= 0 + (6 + 13 + 11) = 30$$

$$UB2_1{}^1 = R_1\,(g_1{}^1) + R_{sp}\,(B - B^1(g_1{}^1)\,)$$

$$= 0 + R_{sp}\,(28 - \frac{0 + 0}{2}\,)$$

$$= 0 + R_{sp}\,(28)$$

$$= 0 + 27 = 27$$

$$UB1_1{}^1 = 30 > 24 = LB_o$$

$$UB2_2{}^1 = 27 > 24 = LB_o$$

$g_1{}^1 = (0,0)$ is not discarded.

Let $i = 2$ $\qquad\qquad g_1{}^2 = (6,3)$

$$UB1_1{}^2 = R_1\,(g_1{}^2) + \sum_{t=2}^{4} r_t\,(k_t)$$

$$= 2 + (6 + 13 + 11)$$

$$= 32$$

$$UB2_1{}^2 = 2 + R_{sp}\,(28 - \frac{6+3}{2}\,)$$

$$= 2 + R_{sp}\,(24)$$

$$= 2 + 23 = 25$$

$$UB1_1{}^2 = 32 > 24 = LB_o$$

$$UB2_1{}^2 = 25 > 24 = LB_o$$

$g_1{}^2 = (6,3)$ is not eliminated.

Let $i = 3$ $\quad g_1^{\,3} = (8,4)$

$UB1_1^{\,3} = 3 + 30 = 30 > LB_o.$

$UB2_1^{\,3} = 3 + R_{sp}\ (28 - \frac{8 + 4}{2})$

$= 3 + R_{sp}\ (22) = 3 + 22$

$= 25 > LB_o.$

$g_1^{\,3} = (8,4)$ is not discarded.

Let $i = 4$ $\quad g_1^{\,4} = (9,5)$

$UB1_1^{\,4} = 5 + 30 = 35 > LB_o.$

$UB2_1^{\,4} = 5 + R_{sp}\ (28 - \frac{9 + 5}{2}) = 5 + R_{sp}\ (21)$

$= 5 + 21 = 26 > LB_o.$

$g_1^{\,4} = (9,5)$ is not eliminated.

Let $i = 5$ $\quad g_1^{\,5} = (11,7)$

$UB1_1^{\,5} = 8 + 30 = 38 > LB_o.$

$UB2_1^{\,5} = 8 + R_{sp}\ (28 - \frac{11 + 7}{2})$

$= 8 + R_{sp}\ (19)$

$= 8 + 19 = 27 > LB_o.$

$g_1^{\,5} = (11,7)$ is not eliminated.

T_1 and TS_1 matrices are as shown before.

The upper and lower bounds are updated as follows:

$$UB_1 = \text{Min}\left\{\underset{i}{\text{Min}}\left\{UB2_1^{\,i}\right\},\ UB_o\right\}$$

$$= \text{Min}\left\{\underset{i}{\text{Min}}\left\{27,25,25,26,27\right\},\ 27\right\}$$

$$= \text{Min}\left\{25,27\right\} = 25.$$

$$LB_1 = \text{Max}\left\{\underset{i}{\text{Max}}\left\{R(g_1^{\,i})\right\},\ LB_o\right\}$$

$$= \text{Max}\left\{\underset{i}{\text{Max}}\left\{0,2,3,5,8\right\},\ 24\right\}$$

$$= \text{Max}\left\{8,24\right\}$$

$$= 24.$$

$$UB_1 = 25 \neq 24 = LB_1.$$

Stage 2 Calculations

First, T_2 and TS_2 matrices are generated similar to the imbedded-state approach. T_2 and TS_2 are given below:

T_2 TS_2

Row i	X_2	Points to stage 1	$g_2^{\,i}$		$R_2(g_2^{\,i})$
1	1	1	0	0	0
2	1	2	6	3	2
3	2	1	7	4	3
4	1	4	9	5	5
5	1	5	11	7	8
6	2	5	18	11	11
7	3	5	21	13	12
8	4	5	23	15	13
9	5	5	25	17	14

The following can be easily observed.

$UB1_2^1 = 0 + 24 = 24 \nless LB_1$

$UB2_2^1 = 0 + R_{sp}(28) = 27 \nless LB_1$

$g_2^1 = (0,0)$ is not eliminated.

Similarly,

$g_2^2 = (6,3)$, $g_2^3 = (7,4)$, $g_2^4 = (9,5)$,

$g_2^5 = (11,7)$, and $g_2^6 = (18,11)$ are not to be eliminated.

Let $i = 7$ $\qquad g_2^7 = (21,13)$

$UB1_2^7 = 12 + 24 \nless LB_1$

$$UB2_2^7 = 12 + R_{sp}\left(28 - \frac{21 + 13}{2}\right)$$

$$= 12 + R_{sp}(11).$$

$$= 12 + 11 = 23 < LB_1.$$

$g_2^7 = (21,13)$ is eliminated.

Similarly $g_2^8 = (23,15)$ and $g_2^9 = (25,17)$ are also eliminated. The updated T_2 and TS_2 matrices are as follows.

	T_2		TS_2		
Row i	X_2	Pointer to stage 1	g_2^i		$R_2(g_2^i)$
1	1	1	0	0	0
2	1	2	6	3	2
3	2	1	7	4	3
4	1	4	9	5	5
5	1	5	11	7	8
6	2	5	18	11	11

The bounds are updated as follows:

$$UB_2 = \text{Min}\left\{\underset{i}{\text{Min}}\left\{27,25,26,26,27,26\right\}, 25\right\}$$

$$= \text{Min}\left\{25,25\right\} = 25.$$

$$LB_2 = \text{Max}\left\{\text{Max}\left\{0,2,3,5,8,11\right\}, 24\right\}$$

$$= \text{Max}\left\{11,24\right\} = 24.$$

$LB_2 \neq UB_2$.

Stage 3 Calculations

After eliminating the solutions which are infeasible and are dominated, T_3 and TS_3 matrices are obtained as follows:

T_3 TS_3

Row i	X_3	Pointer to stage 2	g_3^i		$R_3(g_3^i)$
1	1	1	0	0	0
2	1	2	6	3	2
3	1	3	7	4	3
4	1	4	9	5	5
5	2	1	8	6	6
6	1	5	11	7	8
7	3	1	10	8	9
8	4	1	12	9	11
9	5	1	15	12	13
10	3	4	19	13	14
11	4	4	21	14	16
12	3	5	21	15	17
13	4	5	23	16	19
14	5	5	26	19	21

Let i = 1

$UB1_3^1 = 0 + 11 = 11 < LB_1$. Eliminate g_3^1. Similarly g_3^2, g_3^3,

g_3^4, g_3^5, g_3^6, g_3^7, and g_3^8 are eliminated.

For i = 9-14 the computations are as follows:

i	g_3^i		$UB1_3^i$	$UB2_3^i$
9	15	12	24	28
10	19	13	25	25
11	21	14	27	27
12	21	15	28	28
13	23	16	29	28
14	26	19	32	25

$UB1_3{}^i$ and $UB2_3{}^i$, for i = 9,14, are not less than LB_2. Therefore $g_3{}^i$ (i = 9,14) are not eliminated. The updated T_3 and TS_3 matrices are as follows:

T_3 TS_3

Row i	X_3	Pointer to stage 2	$g_3{}^i$		$R_3(g_3{}^i)$
1	5	1	15	12	13
2	3	4	19	13	14
3	4	4	21	14	16
4	3	5	21	15	17
5	4	5	23	16	19
6	5	5	26	19	21

The bounds are updated as follows:

$$UB_3 = \text{Min}\left\{\text{Min}\left\{28,25,27,28,28,25\right\}, 25\right\}$$

$$= \text{Min}\left\{25,25\right\} = 25.$$

$$LB_3 = \text{Max}\left\{\text{Max}\left\{13,14,16,17,19,21\right\}, 24\right\}$$

$$= \text{Max}\left\{21,24\right\} = 24.$$

Stage 4 Calculations

After eliminating the solutions which are infeasible and dominated, the T_4 and TS_4 matrices are generated as follows:

T_4 TS_4

Row i	X_4	Pointer to stage 3	g_4^i	$R_4(g_4^i)$
1	1	1	15 12	13
2	1	2	19 13	14
3	1	3	21 14	16
4	1	4	21 15	17
5	2	1	20 16	17
6	1	5	23 16	19
7	3	1	21 18	20
8	1	14	26 19	21
9	2	5	28 20	23
10	3	3	27 22	23
11	4	1	24 24	23
12	3	4	27 23	24

Since $UB1_4^i$ (i = 1,11) are less than 24 = LB_3, g_4^i (i = 1,11) are eliminated from the set. The updated T_4 and TS_4 matrices are as follows:

T_4 TS_4

Row i	X_4	Pointer to stage 3	g_4^i	$R_4(g_4^i)$
1	3	4	27 23	24

$UB1_4^1 = 24$ $UB2_4^1 = 24$

$$UB_4 = \text{Min}\left\{24,25\right\} = 24$$

$$LB_4 = \text{Max}\left\{24,24\right\} = 24.$$

$$LB_4 = UB_4.$$

The optimal solution is found with $R^* = 24$. The optimal decision variable are determined by tracing back through the T-matrices.

$$X^* = (5,1,3,3).$$

4.3: A Modified Hydrid Algorithm

The example problem will again be solved using the modified hybrid algorithm. After calculation of lower and upper bounds, three points with the objective function value of 24 are obtained, as shown in Table III. Any of these points can be considered as X', ie.

Case 1: $\underset{\sim}{X}' = (5, 1, 3, 3)$ or

Case 2: $\underset{\sim}{X}' = (1, 1, 5, 5)$ or

Case 3: $\underset{\sim}{X}' = (1, 2, 4, 4)$

In this example we consider the first two cases individually to demonstrate the performance of the modified algorithm:

Case 1:

$\underline{X}^1 = (5, 1, 3, 3);$ $\qquad R_{sp}(\underline{X}^1) = 24$

$\underline{X}^2 = (5, 1, 3, 4);$ $\qquad R_{sp}(\underline{X}^2) = 27$

Stage 1

$$X_1^{\,1} \geq X_2^{\,1} \Longrightarrow S_1 = \{X_1^{\,1}\} = \{5\}.$$

T_1 $\qquad$ TS_1

Row i	X_1	Pointer to stage 0	$g_1^{\,i}$	$R_1(g_1^{\,i})$
1	5	-	11 7	8

Stage 2

$$X_1^2 \geq X_2^2 \Rightarrow S_2 = \left\{X_1^2\right\} = \left\{1\right\}$$

T_2 TS_2

Row i	X_2	Pointer to stage 1	g_2^i	$R_2(g_2^i)$
1	1	1	11 7	8

Stage 3

$$X_1^3 \geq X_2^3 \Rightarrow S_3 = \left\{X_1^3\right\} = \left\{3\right\}$$

T_3 TS_3

Row i	X_3	Pointer to stage 2	g_3^i	$R_3(g_3^i)$
1	3	1	21 15	17

Stage 4

$$X_1^4 < X^4 \Rightarrow S_4 = \left\{X_1^4,\ X_1^4 + 1, \ldots, X_2^4\right\}$$

$$= \left\{3,4\right\}$$

T_4 TS_4

Row i	X_4	Pointer to stage 3	g_4^i	$R_4(g_4^i)$
1	3	1	27 23	24
2	4	1	30 27	27

Feasibility test will eliminate g_4^2.

Thus optimal objective function value,

$$R(\underline{X}^*) = 24 \quad \text{and} \quad X^* = (5, 1, 3, 3).$$

Case 2:

$X^1 = (1, 1, 5, 5)$, $\qquad R_{sp}\,(\underline{X}^1) = 24$

$X^2 = (5, 1, 3, 4)$, $\qquad R_{sp}\,(\underline{X}^2) = 27$

Stage 1

$$X_1^{\,1} < X_1^{\,2} \Rightarrow S_1 = \{1, 2, 3, 4, 5\}$$

T_1 $\qquad$ TS_1

Row i	X_1	Pointer to stage 0	$g_1^{\,i}$		$R_1(g_1^{\,i})$
1	1	-	0	0	0
2	2	-	6	3	2
3	3	-	8	4	3
4	4	-	9	5	5
5	5	-	11	7	8

None of $g_1^{\,i}$ is eliminated by the tests.

$$UB_1 = 25 \neq 24 = LB_1.$$

Stage 2

$$X_2^{\,1} \geq X_2^{\,2} \Rightarrow S_2 = \{X_2^{\,1}\} = \{1\}$$

T_2 $\qquad$ TS_2

Row i	X_2	Pointer to stage 1	$g_2^{\,i}$		$R_2(g_2^{\,i})$
1	1	1	0	0	0
2	1	2	6	3	2
3	1	3	8	4	3
4	1	4	9	5	5
5	1	5	11	17	8

$$UB_2 = 25 \neq 24 = LB_2$$

Stage 3

$$X_3^1 > X_3^2 \Rightarrow S_3 = \{X_3^1\} = \{5\}$$

T_3 TS_2

Row i	X_4	Pointer to stage 2	g_4^i		$R_3(g_3^i)$
1	5	1	15	12	13
2	5	2	21	15	15
3	5	3	23	16	16
4	5	4	24	17	18
5	5	5	26	29	21

g_3^5 is eliminated (infeasibility).

$UB_3 = 25 \neq 24 = LB_3$.

Stage 4

$$X_4^1 > X_4^2 \Rightarrow 54 = \{X_4^1\} = \{5\}.$$

T_4 TS_4

Row i	X_4	Pointer to stage 3	g_4^i		$R_4(g_4^i)$
1	5	1	25	27	24
2	5	2	31	30	26
3	5	3	33	31	27
4	5	4	34	32	29

Infeasibility eliminates g_4^2, g_4^3 and g_4^4. Therefore the optimal solution is

$$X^* = (1, 1, 5, 5), \quad \text{with} \quad R(X^*) = 24.$$

4.4: A Multiple District Optimization Example

The State of Texas is divided into 25 highway Districts. In each District there are more than 200 highway segments which are considered annually for rehabilitation and maintenance activities. In each budget cycle it is necessary to observe or estimate each highway's pavement condition and also to estimate the condition of the segment in succeeding years. If any segment does not satisfy the minimum serviceability or maximum distress requirements, a maintenance strategy should only be considered if its use results in this segment exceeding the minimum requirements. The model requires the following data information.

1. A description of the highway segments used in each district.
2. Pavement condition ratings for each segment.
3. The gain-of-rating matrices.
4. The pavement survivor matrices.
5. Resource information in each district.
6. Available state budget.

Since most of the data needed are not readily available at present for most of the Districts in the state of Texas, a hypothetical example problem was generated.

The example problem has 3 highway Districts, each with 10 highway segments to be considered for maintenance in each district. A total of six maintenance strategies are adopted. They are: (1) seal coat, (2) 1.0" overlay, (3) 2.0" overlay, (4) 3.0" overlay, (5) 5.0" overlay, and (6) 7.0" overlay. Six types of pavement distress are used to measure the segment deterioration. They are: (1) rutting, (2) alligator cracking, (3) longitudinal cracking, (4) transverse cracking, (5) failures per mile, and (6) the serviceability index. The manpower resources are:

(1) asphalt cement, (2) grader, (3) loader, (4) truck, (5) grader operator, (6) loader operator and (7) truck operator. The budget is the last resource to be considered here.

In each District, highway segments are divided into 2 classes. The first class consists of 'U.S.' and 'State highways' and the second type consists of 'Farm-to-Market' highways.

Highway Segment Information. The following information is needed for each highway segment within all Districts; (a) Highway type, (b) length (miles) and (c) width (feet) of each highway segment. The traffic index and environmental factors are assumed to be unity for this example. The data required is as shown in Table V.

The current rating of highway segments by distress types are shown in Table VI. This information is needed for each District.

The enhancement in pavement quality level attained through the application of a maintenance strategy for various distress types are shown in Table VII. The quality level cannot be greater than the maximum possible rating. If an application of any one strategy causes this to occur, the highway rating is fixed at this maximum level.

Pavement survivor matrices are developed for each distress type and maintenance strategy combination. All highway segments within each District are assumed to have identical pavement deterioration curves. Road deterioration curve fractions for each type of maintenance strategy are listed in Tables IX, X, XI, XII, and XIII by distress type. The road deterioration curves are determined by multiplying the road deterioration fractions by the maximum quality levels.

The resources constraints need two major inputs: requirements and availability. The first indicates how much of a given resource will be used by maintenance strategy (per mile-ft. of the pavement) and the second indicates how much of the resources are available. These are shown in Table XIV.

The optimum solution to the three District example is given in Tables XV, and XVI.

Table XV shows the resulting optimal maintenance strategy schedule for each highway segment in the three Districts. The optimal budget utilization is shown in Table XVI. As a result of solving each branch (District) of the dynamic programming model, the minimum required and the maximum needed funds for a District to maintain the pavement quality are obtained. Columns 2 and 3 of the Table XVI, indicate the maximum and minimum budget levels. The sum of maximum budget levels for the three Districts is 256,000 dollars; but the available budget for the three Districts is only 250,000 dollars. (If the sum of minimum required budgets exceeds the available budget, an infeasible solution will result). The optimum budget levels for the three Districts are shown in Column 4 and the corresponding utilities are shown in Column 5.

TABLE V

HIGHWAY SEGMENT DATA

No.	Highway Type	Length (Mile)	Width (Feet)	Traffic Index	Environmental Index
1	1	3.309	36.000	1.000	1.000
2	1	2.266	12.000	1.000	1.000
3	2	3.818	12.000	1.000	1.000
4	1	2.512	12.000	1.000	1.000
5	1	4.712	36.000	1.000	1.000
6	2	1.663	12.000	1.000	1.000
7	1	3.572	24.000	1.000	1.000
8	1	2.462	12.000	1.000	1.000
9	2	2.625	12.000	1.000	1.000
10	1	1.590	12.000	1.000	1.000

TABLE VI

CURRENT RATING OF SEGMENTS

Segment No.	DISTRESS TYPE 1	2	3	4	5	6
1	15.000	25.000	20.000	25.000	30.000	40.000
2	5.000	20.000	20.000	25.000	30.000	0.000
3	15.000	10.000	5.000	25.000	30.000	10.000
4	5.000	25.000	25.000	25.000	30.000	40.000
5	5.000	10.000	15.000	5.000	30.000	0.000
6	15.000	25.000	15.000	25.000	30.000	40.000
7	15.000	25.000	25.000	25.000	30.000	40.000
8	15.000	20.000	20.000	25.000	30.000	40.000
9	5.000	25.000	20.000	5.000	30.000	10.00
10	5.000	5.000	5.000	5.000	30.000	10.000

TABLE VII

GAIN-OF-RATING MATRIX

Strategy No.	DISTRESS TYPE 1	2	3	4	5	6
1	0.000	15.000	15.000	15.000	10.000	2.000
2	13.000	19.000	19.000	19.000	24.000	45.000
3	13.000	20.000	20.000	20.000	25.000	45.000
4	15.000	25.000	25.000	20.000	30.000	50.000
5	15.000	25.000	25.000	20.000	35.000	50.000
6	15.000	25.000	25.000	20.000	40.000	50.000

TABLE VIII

PAVEMENT DETERIORATION FACTORS FOR R&M STRATEGY 1

Year	DISTRESS TYPE					
	1	2	3	4	5	6
1	1.000	1.000	1.000	1.000	1.000	1.000
2	0.930	0.940	0.930	0.920	1.000	0.900
3	0.910	0.890	0.880	0.860	0.910	0.700
4	0.880	0.890	0.870	0.850	0.780	0.500
5	0.780	0.650	0.670	0.670	0.470	0.400
6	0.310	0.280	0.370	0.380	0.220	0.300
7	0.220	0.240	0.320	0.330	0.200	0.200
8	0.150	0.150	0.180	0.180	0.100	0.100
9	0.070	0.090	0.090	0.090	0.040	0.100
10	0.050	0.070	0.070	0.060	0.010	0.000
11	0.020	0.020	0.020	0.010	0.000	0.000
12	0.020	0.010	0.010	0.010	0.000	0.000
13	0.020	0.010	0.010	0.010	0.000	0.000
14	0.020	0.010	0.010	0.000	0.000	0.000
15	0.010	0.000	0.000	0.000	0.000	0.000
16	0.010	0.000	0.000	0.000	0.000	0.000
17	0.010	0.000	0.000	0.000	0.000	0.000
18	0.010	0.000	0.000	0.000	0.000	0.000
19	0.010	0.000	0.000	0.000	0.000	0.000
20	0.010	0.000	0.000	0.000	0.000	0.000

TABLE IX

PAVEMENT DETERIORATION FACTORS FOR R&M STRATEGY 2

Year	DISTRESS TYPE					
	1	2	3	4	5	6
1	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	0.890	1.000	1.000	1.000	1.000
4	1.000	0.820	1.000	1.000	1.000	0.900
5	0.880	0.730	1.000	1.000	1.000	0.800
6	0.780	0.670	0.750	0.830	1.000	0.700
7	0.460	0.670	0.500	0.670	1.000	0.600
8	0.250	0.670	0.500	0.670	0.330	0.500
9	0.250	0.670	0.250	0.330	0.330	0.400
10	0.250	0.360	0.000	0.000	0.330	0.300
11	0.000	0.110	0.000	0.000	0.000	0.000
12	0.000	0.090	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000

TABLE X

PAVEMENT DETERIORATION FACTORS FOR R&M STRATEGY 3

Year	DISTRESS TYPE					
	1	2	3	4	5	6
1	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	0.950	0.930	0.940	1.000	1.000
4	1.000	0.910	0.930	0.940	0.890	0.900
5	0.790	0.900	0.400	0.430	0.530	0.800
6	0.750	0.610	0.140	0.180	0.230	0.700
7	0.750	0.560	0.140	0.180	0.160	0.600
8	0.750	0.550	0.120	0.140	0.150	0.500
9	0.750	0.510	0.070	0.060	0.130	0.400
10	0.750	0.280	0.020	0.010	0.080	0.300
11	0.330	0.170	0.000	0.000	0.020	0.000
12	0.250	0.140	0.000	0.000	0.000	0.000
13	0.250	0.140	0.000	0.000	0.000	0.000
14	0.170	0.140	0.000	0.000	0.000	0.000
15	0.080	0.080	0.000	0.000	0.000	0.000
16	0.000	0.010	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000

TABLE XI

PAVEMENT DETERIORATION FACTORS FOR R&M STRATEGY 4

Year	DISTRESS TYPE					
	1	2	3	4	5	6
1	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	0.770	1.000	1.000	0.770	0.900
6	0.830	0.640	0.330	0.630	0.510	0.800
7	0.710	0.580	0.110	0.260	0.480	0.700
8	0.660	0.530	0.000	0.220	0.360	0.600
9	0.620	0.510	0.000	0.110	0.330	0.500
10	0.380	0.380	0.000	0.040	0.240	0.500
11	0.300	0.210	0.000	0.000	0.170	0.000
12	0.300	0.190	0.000	0.000	0.170	0.000
13	0.300	0.190	0.000	0.000	0.170	0.000
14	0.280	0.170	0.000	0.000	0.170	0.000
15	0.220	0.150	0.000	0.000	0.170	0.000
16	0.170	0.100	0.000	0.000	0.170	0.000
17	0.120	0.070	0.000	0.000	0.070	0.000
18	0.040	0.060	0.000	0.000	0.000	0.000
19	0.040	0.060	0.000	0.000	0.000	0.000
20	0.040	0.030	0.000	0.000	0.000	0.000

TABLE XII

PAVEMENT DETERIORATION FACTORS FOR R&M STRATEGY 5

Year	DISTRESS TYPE					
	1	2	3	4	5	6
1	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000
6	1.000	0.710	0.330	0.330	0.750	0.900
7	1.000	0.620	0.330	0.330	0.590	0.900
8	1.000	0.440	0.280	0.280	0.500	0.800
9	1.000	0.290	0.170	0.170	0.480	0.700
10	1.000	0.290	0.170	0.170	0.250	0.600
11	0.670	0.290	0.170	0.170	0.250	0.000
12	0.670	0.170	0.170	0.170	0.250	0.000
13	0.670	0.140	0.170	0.170	0.250	0.000
14	0.670	0.140	0.170	0.170	0.250	0.000
15	0.220	0.120	0.170	0.170	0.200	0.000
16	0.000	0.000	0.170	0.170	0.000	0.000
17	0.000	0.000	0.170	0.170	0.000	0.000
18	0.000	0.000	0.170	0.170	0.000	0.000
19	0.000	0.000	0.170	0.170	0.000	0.000
20	0.000	0.000	0.170	0.170	0.000	0.000

TABLE XIII

PAVEMENT DETERIORATION FACTORS FOR R&M STRATEGY 6

Year	DISTRESS TYPE					
	1	2	3	4	5	6
1	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	0.900
5	1.000	1.000	1.000	1.000	1.000	0.800
6	0.720	0.490	1.000	1.000	0.470	0.700
7	0.670	0.360	1.000	1.000	0.360	0.600
8	0.580	0.360	1.000	1.000	0.320	0.500
9	0.500	0.360	0.650	0.650	0.270	0.400
10	0.500	0.290	0.600	0.600	0.270	0.300
11	0.360	0.270	0.600	0.600	0.270	0.000
12	0.330	0.270	0.600	0.600	0.200	0.000
13	0.330	0.270	0.530	0.510	0.180	0.000
14	0.280	0.270	0.400	0.400	0.180	0.000
15	0.170	0.210	0.380	0.380	0.150	0.000
16	0.170	0.190	0.210	0.200	0.090	0.000
17	0.170	0.190	0.200	0.000	0.090	0.000
18	0.170	0.180	0.200	0.000	0.090	0.000
19	0.170	0.110	0.200	0.000	0.090	0.000
20	0.170	0.090	0.200	0.000	0.090	0.000

TABLE XIV

RESOURCE REQUIREMENTS*

Strategy	RESOURCES						
	1	2	3	4	5	6	7
1	0.800	0.000	0.012	0.060	0.000	0.012	0.060
2	3.000	0.000	0.000	0.278	0.000	0.000	0.278
3	1.500	0.000	0.000	0.278	0.000	0.000	0.278
4	4.100	0.000	0.000	0.556	0.000	0.000	0.556
5	8.100	0.000	0.000	0.834	0.000	0.000	0.834
6	1.500	01.000	0.333	3.611	1.000	0.333	3.611

*Resources 1 and 2 are materials. UNIT is TON/MILE-FT

Resources 3 and 4 are equipment. UNIT is EQUIPMENT-DAYS/MILE-FT

Resources 5, 6 and 7 are manpower. UNIT is MANPOWER-DAYS/MILE-FT

TABLE XV

OPTIMAL MAINTENANCE DECISIONS

DISTRICT	SEGMENTS 1	2	3	4	5	6	7	8	9	10
1	5	4	2	2	6	4	4	2	1	1
2	5	5	2	2	2	5	2	2	1	1
3	5	4	2	3	2	6	4	4	1	1

TABLE XVI

OPTIMAL BUDGET UTILIZATION

District	Budget			Utility
	Maximum	Mininum	Optimum	
1	71,000	66,000	71,000	2171
2	84,000	68,000	78,000	1772
3	101,000	92,000	101,000	1742
TOTAL		250,000		5685

CHAPTER 5

SUMMARY

The major purpose of this report was to describe a technique which can be used in determining the optimal allocation of resources and budget for rehabilitation and maintenance of the highway network system in the State of Texas. In TTI Research Report No. 207-3, the highway maintenance problem at the district level was represented as a 0-1 integer linear programming problem, and an efficient optimization technique presented. In this report, solution of the statewide maintenance problem is presented.

The allocation problem at the state level is modeled as a dynamic programming problem. As shown in Chapter 3, the problem cannot be efficiently modeled as a 0-1 problem, since it will be too large to solve. A solution technique, based on dynamic programming was developed to solve this large, discrete, nonlinear knapsack problem.

A FORTRAN based computer program was written using the algorithm. A hypothetical example was formulated and solved by this program. The compilation time on the FORTRAN H - Extended compiler was 2.43 seconds and the example problem presented took 8.4 seconds of execution time.

REFERENCES

1. Ahmed, N.V., "A Code for 0-1 Integer Linear Programming Problems with Multiple Choice Constraints," an unpublished Dissertation, Industrial Engineering, Texas A&M University, College Station, Texas (1978).

2. Ahmed, N.V., Lytton, R.L., Mahoney, J.P., and Phillips, D.T., "Texas Rehabilitation and Maintenance District Optimization Systems," Research Report No. 207-3, Texas Transportation Institute, Texas A&M University, College Station, Texas (1978).

3. Brooks, R. and Geoffrion, A., "Finding Everett's Lagrangian Multipliers by Linear Programming," Operations Research, Vol. 14, No. 6 (1966), pp. 1149-1152.

4. Epps, J.A., & Meyer, A.H., et al., "Roadway Maintenance Evaluation User's Manual," Maintenance Quality, Methods and Rating, Research Report 151-2, Texas Transportation Institute, Texas A&M University, Texas, 1974.

5. Everett, H.M., "Generalized Lagrangian Multiplier Method for Solving Problems of Optimal Allocation of Resources," Operations Research, Vol 11 (1963), pp. 399-417.

6. Marsten, R.E. and Morin, T.L., "MMDP a Computer Code for Solving Multi-Constrained Knapsack Problems in Integer Variables: User's Guide, Purdue University, West Lafayette, IN 1978.

7. Nemhauser, G.L., Introduction to Dynamic Programming, John Wiley & Sons, Inc., New York, London, Sydney, 1966.

8. Phillips, D.D., Shanumgham, C.V., Ghasemi-Tari, F. and Lytton, R.L., "Rehabilitation And Maintenance System: State Optimal Fund Allocation - Program I," Research Report No. 239-4, Texas Transportation Institute, Texas A&M University, College Station, Texas (1980)

9. Phillips, D.T. and Lytton, R.L., "A Resource Constrained Capital Budgeting Model for State Highway Rehabilitation and Maintenance," Proc. AIIE Spring Annual Conference, Dallas, Texas (1977), pp. 279-304.

10. Phillips, D.T., Shanmughan, C.V., Sathaye, S., and Lytton, R.L., "Rehabilitation And Maintenance System: District Time Optimization," Research Report No. 239-3, Texas Transportation Institute, Texas A&M University, College Station, Texas (1980).

11. Taha, H.A., Integer Programming Theory, "Applications and Computations," Academic Press, New York, San Francisco, London, 1977.

APPENDIX A

DEVELOPMENT OF ALGORITHMS TO SOLVE THE STATE OPTIMIZATION PROBLEM

APPENDIX A

A.1 Scope of the Hybrid Algorithm

The hybrid algorithm was developed to solve problems in the general form of a separable nonlinear knapsack problem with non-negativity assumptions on all the problem coefficients and decision variables. The mathematical representation of such a problem is as follows:

$$R = \text{Max.} \sum_{j=1}^{N} r_j(X_j) \tag{A.1}$$

Subject to

$$\sum_{j=1}^{N} a_{ij}(X_j) \leq b_i \qquad i = 1, 2, \ldots, M$$

$$X_j \text{ is contained in } S_j \qquad j = 1, 2, \ldots, N$$

$$S_j = \{1, 2, \ldots, K_j\} \qquad \forall\, K_j,\ j = 1, 2, \ldots, N.$$

K_j is a finite positive integer, while $r_j(X_j)$ and $a_{ij}(X_j)$ are non-decreasing positive valued functions, and $b_i \geq 0$ for all $i = 1, 2, \ldots, M$. It is assumed that $r_j(0) = 0$ for all $j = 1, 2, \ldots, N$ and $a_{ij}(0) = 0$ for all values of $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$.

This mathematical model is the general form of knapsack-type resource allocation problems in which there are N different sections, each section involving K_j different projects, and the selection of project X_j in section j consumes $a_{ij}(X_j)$ of resource type i, and provides the return of $r_j(X_j)$. Each section is assumed to have a "do nothing" alternative action which consumes zero amount of resources with an associated return value of zero. The objective is to maximize the summation of the returns obtained in each section. The amount of resource i available is b_i, for i $i = 1, 2, \ldots, M$. A special case of the model is where the return and resource consumption functions are presented in a linear form. In that case, the proposed algorithm is a dynamic programming solution procedure for the general linear integer programming problem. For the case of $K_j = 2$ for all j's, the model is a 0-1 integer linear program and can be solved by the hybrid algorithm.

The algorithmic procedure is based upon a combinatorial enumeration scheme. This concept provides the capability of solving problems with non-integral state spaces, i.e., the $a_{ij}(X_j)$ can be a real-valued step function, and in the case of linear integer programs, a_{ij} can be a real-valued coefficient. Note that the integrality assumption on a_{ij} in the tabular dynamic programming problem has limited the application of this method.

A.2 The Imbedded-State Approach

The imbedded-state approach for state reduction in dynamic programming problems is a methodology which converts an M-dimensional state variable (vector) to a single-state variable. This is accomplished utilizing the points of discontinuity of the return function as a possible solution space. It is assumed that the return function remains constant in the consecutive points of discontinuity. This is a realistic assumption since in the case of integer programming, the function's value between two integer points is of no concern to decision makers. To illustrate the concept of imbedded-state, consider the model presented in Section A.1. Let G_j be the state space at the j^{th} stage of the model, defined as follows:

$$G_j = \{\, g_j \mid g_j = (a_{1j}(X_j),\ a_{2j}(X_j),\ \ldots,\ a_{Mj}(X_j)),\ \text{for } X_j \varepsilon S_j \,\} \quad . \ . \ . \ (A.2)$$

The imbedded-state space for the j^{th} stage is defined as follows:

$$F_1 = G_1$$

$$F_j = G_j \circ G_{j-1} \qquad \text{for} \qquad j = 2, 3, \ldots, N$$

where the operator "o" is defined as the sums of each element of G_j with each element of G_{j-1}, for all the elements contained in set G_j and G_{j-1}. The resulting set F_j defines the state space for stages 1, 2, ..., j, which can then be modified by feasibility and dominancy tests. The feasibility of the elements in F_j is checked and those elements which provide an infeasible solution are eliminated. As a result, a set of feasible points is obtained at each stage. Let $F_j^f \subseteq F_j$ define the set of feasible points at stage j. The mathematical expression for F_j^f is:

$$F_j^f = \{g_j' \mid g_j' \varepsilon G_j \circ G_{j-1}, \text{ and } g_j' \leq (b_1, b_2, \ldots, b_M)\}$$

The dominancy test is performed to eliminate those solution points with lower return and higher resource consumption. As as result, a set of feasible and efficient points are obtained. Let $F_j^e \subseteq F_j^f \subseteq F_j$ represent this set. This iterative procedure is continued until F_N^e is reached. The point(s) in F_N^e with the highest return value comprise the optimal solution to the problem. The imbedded-state algorithmic procedure can now be prescribed as follows:

Algorithm 1

Step 1 - Set $j = 1$, $K = K_1$, $F_1 = G_1 = \{a_{11}(\underset{\sim}{x}_1), a_{12}(\underset{\sim}{x}_1), \ldots,$
$a_{1M}(\underset{\sim}{x}_1) \mid \underset{\sim}{x}_1 \in S_1\}$, and $r_1 = \{r_1(x_1) \mid \underset{\sim}{x}_1 \in S_1\}$

Step 2 - $F_1^e = F_1^f = \{g_1 \mid g_1 \leq (b_1, b_2, \ldots, b_M)\}$

Step 3 - Set $j = j+1$, $K = K_j$.

Step 4 - Set $G_j = \{a_{j1}(\underset{\sim}{x}_j), a_{j2}(\underset{\sim}{x}_j), \ldots, a_{jM}(x_j) \mid \underset{\sim}{x}_j \in S_j\}$

Step 5 - Set $F_j = G_j \circ F_{j-1}^e$

Step 6 - Set $F_j^f = \{g_j' \mid g_j' \in F_j$ and $g_j \leq (b_1, b_2, \ldots, b_M)\}$

Step 7 - Set $r_j = \{r_j(g_j') + r_j(\underset{\sim}{x}_j)$ and $\underset{\sim}{x}_j \in S_j\}$

Step 8 - Set $F_j^e = \{F^f$ - all points dominated by better points$\}$

Step 9 - If $j = N$ got to 10, otherwise go to 3.

Step 10- Find the maximum r_N^*.

Step 11 - Stop.

For computational purposes, the information regarding the state space and return values at each stage are stored in a temporary matrix called TS. This matrix consists of M+1 columns and a number of rows which varies for each stage. As an example, the TS matrix at the first stage consists of K_1 rows while in the j^{th} stage it consists of $\prod_{i=1}^{j} k_i$ rows, less rows eliminated either because they are infeasible or dominated. In addition another matrix T_j is employed to store the information required to recover the optimal values of the decision variables through the backtracking procedure. The T_j matrix is composed of two columns and the same number of rows as the TS matrix (in the j^{th} stage), where the rows of T_j possess a one-to-one correspondence to TS. The first column stores the value of the decision variables, X_j's, corresponding to the point in the state space for this row, and the second column stores an index showing the row number of the previous stage from which the current state is obtained.

The algorithm is implemented by first setting TS_1 and T_1 to the following initial values:

T_1			TS_1	
Row i	X_i	Pointer to stage 0	g_1^i	$R_1(g_1^i)$
1	X_1	-	$a_{11}(X_1), \ldots, a_{1M}(X_1)$	$r(X_1)$
2	X_2	-	$a_{11}(X_2), \ldots, a_{1M}(X_2)$	$r(X_2)$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
K_1	X_{K_1}	-	$a_{11}(X_{K_1}), \ldots, a_{1M}(X_{K_1})$	$r(X_{K_1})$

In the following stages, the matrix TS will be updated and will contain the number of rows equal to $\prod_{i=1}^{j} K_i$, less those eliminated by feasibility and dominancy considerations. The matrix T_j is constructed accordingly. At any stage, the matrix TS is sorted according to increasing return values and T_j is arranged in a similar fashion.

When the last stage is enumerated, the optimal value of r_N^* is obtained from the last row of the matrix TS. Referring to the corresponding row of matrix T_N, the value of X_N^* is obtained from the first column. The second column of T_N indicates the row of T_{N-1} from which X_{N-1}^* can be obtained, and so forth.

It is apparent that this method creates a large number of points in the state-space, even for small problems, particularly at intermediate stages, because in each stage almost all combinations of the previous state must be considered. That is, feasibility will not remove many points until the later stages. However, the dominancy properties may provide significant reduction. Therefore, one must find a way to eliminate those states which do not provide a good solution from the outset. An analytical method will subsequently be developed and presented which will eliminate a considerable number of points in the state space which do not lead to a good solution. Since the proposed method is based on the imbedded-state approach, a simple example problem to better illustrate the imbedded state concept was presented in Chapter 4.

In the following sections a methodology will be presented which reduces the state-space solutions by elimination through partial enumeration using a branch and bound (B&B) approach. First, a technique will be presented for calculating the initial lower and upper bounds, and second, and algorithm will be developed by combining the imbedded-state approach and B&B

methodology.

A.3 Calculation of Initial Lower and Upper Bounds

Fathoming of a partial solution by branch and bound (B&B) effectively eliminates nonpromising points from the state space and hence provides extensive savings in computational time and storage. Since the potential state-space grows in a combinatorial fashion with each stage, the use of implicit enumeration approaches such as B&B can be used to reduce the burden of this growth to a reasonable degree. Moreover, the degree of state reduction through the bounds is highly dependent upon how near-optimal the initial bounds are. As an example, if one considers a lower bound of zero, (LB=0) none of the state-space is eliminated in the first stage. However, the use of a better lower bound might eliminate most of the nonpromising state-space in the first stage. Therefore, the efficiency of the method is dependent upon the calculation of a good lower and upper bound.

There are several ways of calculating a lower and upper bound for the optimal solution(s). The use of a surrogate constraint, which converts the problem to a single-constraint problem, provides a narrow bound on the optimal solution. Moreover the surrogate might provide a solution which satisfies the feasibility condition of the original problem, and hence is an optimal solution to the original problem. Subsequent discussion will show that the surrogate problem is merely a relaxation of the constraints and thereby should provide good bounds. A mathematical presentation of the problem with its associated surrogate problem is shown as P and SP, respectively.

Problem P

$$\text{Max.} \sum_{j=1}^{N} r_j(X_j)$$

Subject to:

$$\sum_{j=1}^{N} a_{ij}(X_j) \leq b_i \qquad i = 1, 2, \ldots, M$$

$$X_j \varepsilon S_j = \{1, 2, \ldots, K_j\}$$

Problem SP

$$\text{Max.} \sum_{j=1}^{N} r_j(X_j)$$

Subject to:

$$\sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_i a_{ij}(X_j) \leq \sum_{i=1}^{M} \alpha_i b_i$$

$$\sum_{i=1}^{M} \alpha_i = 1$$

$$\alpha_i > 0 \qquad \forall i = 1, 2, \ldots, M$$

Before further discussion regarding the calculation of the lower and upper bounds, it will be useful to present some of the properties of the surrogate problem. Let $H^1(\underset{\sim}{X})$ and $H^2(\underset{\sim}{X})$ be two set-functions defining the feasible region of Problem P and Problem SP respectively.

1. If $\underset{\sim}{X}$ is a feasible solution to Problem P, it is also feasible to Problem SP, i.e.

If $\underset{\sim}{X} \varepsilon H^1(\underset{\sim}{X}) = \{h_i^1(X) \mid h_4^1(\underset{\sim}{X}) = \sum_{j=1}^{N} a_{ij} (X_j) \leq b_i$ for $i = 1, 2, \ldots, M\}$

$$\underset{\sim}{X} \varepsilon H^2(\underset{\sim}{X}) = \{h^2(\underset{\sim}{X}) \mid h^2(\underset{\sim}{X}) = \sum_{i=1}^{M} \sum_{i=1}^{N} \alpha_i a_{ij}(X_j) \leq \sum_{i=1}^{M} \alpha_i b_i\}$$

2. If $\underset{\sim}{X}^*$ is an optimal solution to problem SP with objective function value $R_{SP}^* (\underset{\sim}{X}^*)$, the objective funtion value of problem P for any $\underset{\sim}{X}$ is always less than or equal to the $R_{SP}^* (\underset{\sim}{X}^*)$, i.e.

$$R_P(\underset{\sim}{X}) \leq R_{SP}^* (\underset{\sim}{X}^*) \qquad\qquad \forall\ \underset{\sim}{X} \varepsilon H^2 (\underset{\sim}{X})$$

3. If $\underset{\sim}{X}^*$ is an optimal solution to Problem SP, and $\underset{\sim}{X}^*$ satisfies the feasibility conditions of Problem P, then $\underset{\sim}{X}^*$ is an optimal solution to Problem P.

The above three properties are the consequence of Theorem 3-5 page 101 of Taha (6), where $a_{ij}(X_j)$ is a linear function defined by $a_{ij}X_j$. For the general case where $a_{ij}(X_j)$ is not a linear function, it can be demonstrated that the above properties still hold. This demonstration is aided by the extension of the above theorems to the case of discrete nonlinear constraints. The extension is stated in the following theorem and proved as follows:

Theorem: Any feasible solution point in Problem P is a feasible solution to the problem SP, i.e.,

$$\forall\ \underset{\sim}{X} \varepsilon H^1(\underset{\sim}{X}) \text{ then } \underset{\sim}{X} \varepsilon H^2(\underset{\sim}{X})$$

Proof: Let $\underset{\sim}{X}$ be any point such that $\underset{\sim}{X} \varepsilon H^1(\underset{\sim}{X})$. This means that

$$\underset{\sim}{X} \varepsilon \{h_i^1 (\underset{\sim}{X}) = \sum_{j=1}^{h} a_{ij}(X_j) \leq b_i \quad \text{for } i = 1, 2, \ldots, M\}$$

$$\underset{\sim}{X} \in \{h_i^1 \ (\underset{\sim}{X}) = \sum_{j=1}^{h} a_{ij}(X_j) \leq b_i \quad \text{for } i = 1, 2, \ldots, M\}$$

On the other hand, the expanded form of $H^2(\underset{\sim}{X})$ is

$$H^2(\underset{\sim}{X}) = \{h^2(\underset{\sim}{X}) \mid \alpha_1 \sum_{j=1}^{N} a_{ij}(X_j) + \overset{\alpha}{X}_2 \sum_{j=1}^{N} a_{2j}(X_j) + \ldots$$

$$\ldots + \alpha_M \sum_{j=1}^{N} a_{Mj}(X_j) \leq \alpha_1 b_1 + \alpha_2 b_2 + \ldots + \alpha_M b_M\}$$

$$= \{h^2(\underset{\sim}{X}) \mid \alpha_1 h_1^1(\underset{\sim}{X}) + \alpha_2 h_2^1(X) + \ldots + \alpha_M h_M^1(\underset{\sim}{X}) \leq \alpha_1 b_1 +$$

$$\alpha_2 b_2 \ldots + \alpha_M b_M\} \tag{A.3}$$

but $\underset{\sim}{X} \in h_i^1(\underset{\sim}{X}) \leq b_i \quad \underset{\sim}{X} \in \alpha_i h_i^1(\underset{\sim}{X}) \leq \alpha_i b_i$ for $\alpha_i \geq 0$

Thus, each term of the left-hand side of the inequality is less than each term of the right-hand side in Equation (A.3). Therefore, any $\underset{\sim}{X} \in H^1(\underset{\sim}{X})$ also belongs to $H^2(\underset{\sim}{X})$. It should be noted, however, that any point feasible in Problem SP is not necessarily a feasible solution to Problem P. Therefore, SP is a relaxation of Problem P, and hence the above three properties also hold for the non-linear case.

The results obtained by this theorem reveal that to obtain a good initial lower and upper bound, one may solve a single-constraint D.P. problem SP, and if the optimal solution of SP is feasible with respect to Problem P, the optimal solution has been found. On the other hand, if the solution is not feasible with respect to Problem P, the right-hand side of the single-constraint problem could be reduced until a feasible solution to Problem P is obtained. This solution could be used as an initial lower bound for the optimal solution of Problem P, since it is feasible, and the optimal solution of SP could be used as an upper bound.

The algorithmic procedure for obtaining lower and upper bounds is similar to a tabular D.P. algorithm. For this reason, all the data of the surrogate problem is rounded to integer-valued data. A conservative way to obtain integer-valued data is to round down all the coefficients of the surrogated right-hand side. This calculation is expressed mathematically as follows:

Let [I] define the largest integer value less than or equal to I.

The surrogate constraint coefficients are obtained by:

$$A_j(X_j) = \left[\sum_{i=1}^{M} \alpha_i a_{ij}(X_j) \right]$$

and the right hand side of the surrogated constraint is:

$$B = \begin{cases} \displaystyle\sum_{i=1}^{M} \alpha_i b_i & \text{if } \displaystyle\sum_{i=1}^{M} \alpha_i b_i = \left[\sum_{i=1}^{M} \alpha_i b_i \right] \\ \displaystyle\sum_{i=1}^{M} \alpha_i b_i \quad +1 & \text{otherwise} \end{cases}$$

The following algorithm provides a procedure for obtaining the initial lower and upper bounds. Notice that the inherent sensitivity analysis in D.P. enables one to find the optimal solution for different values of the right-hand side (RHS).

<u>Algorithm 2 - Calculation of Initial Bounds</u>

Step 1 - Set k=1

Step 2 - Solve the following single constraint D.P. problem

$$\text{Max. } R(B) = \sum_{j=1}^{N} r_j(X_j)$$

Subject to:

$$\sum_{j=1}^{N} A_j(X_j) \leq B$$

$$X_j \text{ is contained in } S_j$$

Step 3 - Check whether the solution obtained in Step 2 is feasible to Problem P, i.e., if the optimal solution from Step 2, $\underset{\sim}{X}^*$, satisfies

$$\sum_{j=1}^{N} a_{ij}(X_j^*) \leq b_i \qquad \text{for } i = 1, 2, \ldots, M$$

go to Step 5, otherwise go to Step 4.

Step 4 - Find the optimal solution of SP for the RHS of B-k, i.e., designate this solution as $\underset{\sim}{X}^*$, and set k = k+1. Go to 3.

Step 5 - If k > 1 got to Step 6, otherwise set LB = UB = R(B) and stop.

Step 6 - Set UB = R(B) and LB = R(B-k) and stop.

Depending on the result of this algorithm, it will be determined whether further calculations are needed, or the optimal solution has already been obtained (LB=UB). If LB < UB, further iterations must be performed in order to push the bounds to equality. The procedure for further computations is discussed in the next section.

A.4 Development of the Hybrid Algorithm

As previously mentioned, the computational effort required to solve a NKP grows rapidly as the problem size increases. This is due to the combinatorial nature of the problem , which demands the evaluation of a large state space, which increases substantially at each stage. However, implementation of the algorithm in Section A.3 fathoms partial solutions in the state-space which would result in a solution worse than the predetermined lower bound. Although this will reduce the solutions to be examined by a significant amount, it is still important to consider the computational demands of the remaining problem. Therefore, additional reduction techniques are important for increasing the computational capability. This additional reduction can be accomplished by updating the lower and upper bound as the computation progresses. These updates are performed as the size of the gap between the lower and upper bound becomes smaller, until either the size of the gap reaches zero or becomes very small.

The upper and lower bounds can easily be calculated for incorporation in the hybrid algorithm. At each stage, two sets of upper bounds for the objective function value is calculated by two different methods. The first method is to ignore all the constraints in succeeding stages. This set of upper bounds can be calculated by:

$$UB1_j^i = R(g_j^i) + \sum_{t=j+1}^{N} r_t\,(K_t) \tag{A.4}$$

where $R(g_j^i)$ is the return of the i^{th} element of the state space g_j at stage j, and $\sum_{t=j+1}^{N} r_t\,(K_t)$ is the sum of maximum possible returns from the succeeding stages.

The second method for obtaining a set of upper bounds is based on the surrogate approach. At any stage j, the bound can be estimated by combining the return of the i^{th} point in g_j with the returns of the succeeding stages calculated from the surrogate problem. Note that returns from the succeeding stages of the surrogate problem con easily be obtained by sensitivity analysis on the RHS of the surrogate solution. The set of upper bounds assigned by the second method can be calculated as follows:

$$UB_j^i = R(g_j^i) + R_{SP}\ (B- \ B'(g_j^i)\ ) \tag{A.5}$$

where R_{SP} is the return of the surrogate problem as a function of the RHS, and $B - \ B'(g_j^i)$ is the initial RHS value (B) less the integer value of the surrogated resource consumed through stage j. These two sets of upper bounds will then be utilized as one of the criteria for discarding inefficient points from g_j. The discarded points are those that satisfy the following relations.

$$UB1_j^i < LB_{\ j-1} \text{ or } UB2_j^i < LB_{j-1} \tag{A.6}$$

Further, at each stage the upper and lower bounds are updated and used as a termination criterion; i.e., the algorithm terminates whenever the lower bound reaches the upper bound. The updated upper bound is determined as follows:

$$UB_j = \text{Min.} \quad \{\underset{i}{\text{Min.}}(UB2_j^i), UB_{j-1}\} \tag{A.7}$$

Based on the concept of the imbedded state and surrogate bounding along with the above updating considerations, a hybrid algorithm has been developed to find the optimal solution to the NKP. In summary, the first algorithm gives a procedure for generating a reduced state space, the second algorithm facilitates the computation of an initial bounding criteria to aid further reduction of state space, and the combination of these two algorithms incorporating updating of the lower and upper bounds provides an efficient algorithmic procedure for solving NKP's in a reduced state space.

A general description of the hybrid algorithm is given below and also appears in flowchart form as Figure A.1. Steps 1-8 of the algorithm comprise the solution of the original surrogate problem to obtain an initial bound. If the optimal solution of the surrogate problem satisfies the feasibility condition of the main problem, steps 9-29 are discarded and the final solution will be obtained by step 30; otherwise, further computation will take place starting from step 9. The computations regarding the first stage are performed through steps 10-11. Steps 11 and 12 comprise the construction of the imbedded state space. The reductions in the imbedded state space through feasibility and dominancy tests are performed by steps 13-15, and the further reductions by bounding are achieved through steps 18-22. The lower and upper bounds are updated by steps 23-24 and the remaining steps comprise the tests for termination of the algorithm.

<u>Algorithm 3 - The Hybrid Algorithm</u>

Step 1 - Set R=0 ℓ=0 and $\alpha_i = \frac{1}{M}$ for 1 = 1, 2, ..., M,

Step 2 - Set $A_j(X_j) = \left[\sum_{i=1}^{M} \alpha_i \; a_{ij}(X_j)\right]$ for j = 1, 2, ..., N

Step 3 - Set $B = \begin{cases} \sum_{i=1}^{M} \alpha_i b_i & \text{if } \sum_{i=1}^{M} \alpha_i b_i = \left[\sum_{i=1}^{M} \alpha_i b_i \right] \\ \left[\sum_{i=1}^{M} \alpha_i b_i \right] + 1 & \text{otherwise} \end{cases}$

Step 4 - Find the optimal solution to the following problem:

$$\text{Max. } R_{SP}(B) = \sum_{j=1}^{N} r_j(X_j)$$

Subject to:

$$\sum_{j=1}^{N} A_j(X_j) \leq B$$

call the solution $\underset{\sim}{X}^*$ and $R_{SP}(B)$, and let $UB_o = R^*_{SP}(B)$.

Step 5 - If $\sum_{j=1}^{N} a_{ij}(X^*_j) \leq b_i$ for $i = 1, 2, \ldots, M$ go to step 8,

otherwise go to step 6.

Step 6 - Set $\ell = \ell + 1$

Step 7 - Let $\hat{B} = B - \ell$ and find the optimal solution of $R_{SP}(\hat{B})$, call this solution X^*, $R^*_{SP}(\hat{B})$, and go to step 5.

Step 8 - If $\ell > 0$, set $LB_o = R^*_{SP}(\hat{B})$, go to step 9, otherwise set $LB_o(B) = UB_o(B)$, go to step 30.

Step 9 - Let $j=1$, $S=S_j$, $F^e_j = G_j = \{a_{j1}(X_j)$, and $a_{j2}(X_j), \ldots, a_{jM}(X_j)$; for $X_j \in S\}$ and $r(g_j) = \{r_j(X_j) \mid X_j \in S\}$

Step 10 - Go to step 16.

Step 11 - Let $j = j+1$, $S=S_j$, $K_j=K$ and $G_j = \{a_{j1}(X_j), a_{j2}(X_j), \ldots, a_{jM}(X_j); \text{ for } X_j \in S\}$

Step 12 - $F_j = G_j \circ F^e_{j-1}$

Step 13 - $F^f_j = \{\underset{\sim}{g}^i_j \mid \underset{\sim}{g}^i_j \in F_j \text{ and } \underset{\sim}{g}^i_j \leq (b_1, b_2, \ldots, B_M)\}$

Step 14 - $R_j = \{r^i_j \mid r^i_j = R_j(g^i_j) + r_j(\underset{\sim}{X}_j) \text{ and } \underset{\sim}{X}_j \in S_j\}$

Step 15 - Let $F^e_j = \{F^f -$ all the points dominated by better points$\}$

Step 16 - Let the number of points in F^e_j be KK

Step 17 - $i=1$

Step 18 - Let $UB1^i_j = R(\underset{\sim}{g}^i_j) + \sum_{t=J+1}^{N} r_t(K)$

Step 19 - Let $UB2^i_j = R(\underset{\sim}{g}^i_j) + R_{SP}\ (B - B'(\underset{\sim}{g}^i_j)\ )$

Step 20 - Let $R(g^i_j) = -1$ for those i which satisfy $UB1^i_j < LB_{j-1}$ or $UB2^i_j < LB_{j-1}$

Step 21 - If i=KK go to step 22 otherwise set i=i+1 and go to step 18

Step 22 - Redefine $R(g_j)$ by elimination of all negative $R(g_j)$

Step 23 - Let $UB_j = \text{Min.}\ \{\underset{i}{\text{Min.}}\ UB2^i_j, UB_{j-1}\}$

Step 24 - Set $LB_j = \text{Max.}\ \{\underset{i}{\text{Max.}}\ R(\underset{\sim}{g}^i_j), LB_{j-1}\}$

Step 25 - If $LB_j = UB_j$ go to step 27, otherwise go to step 26

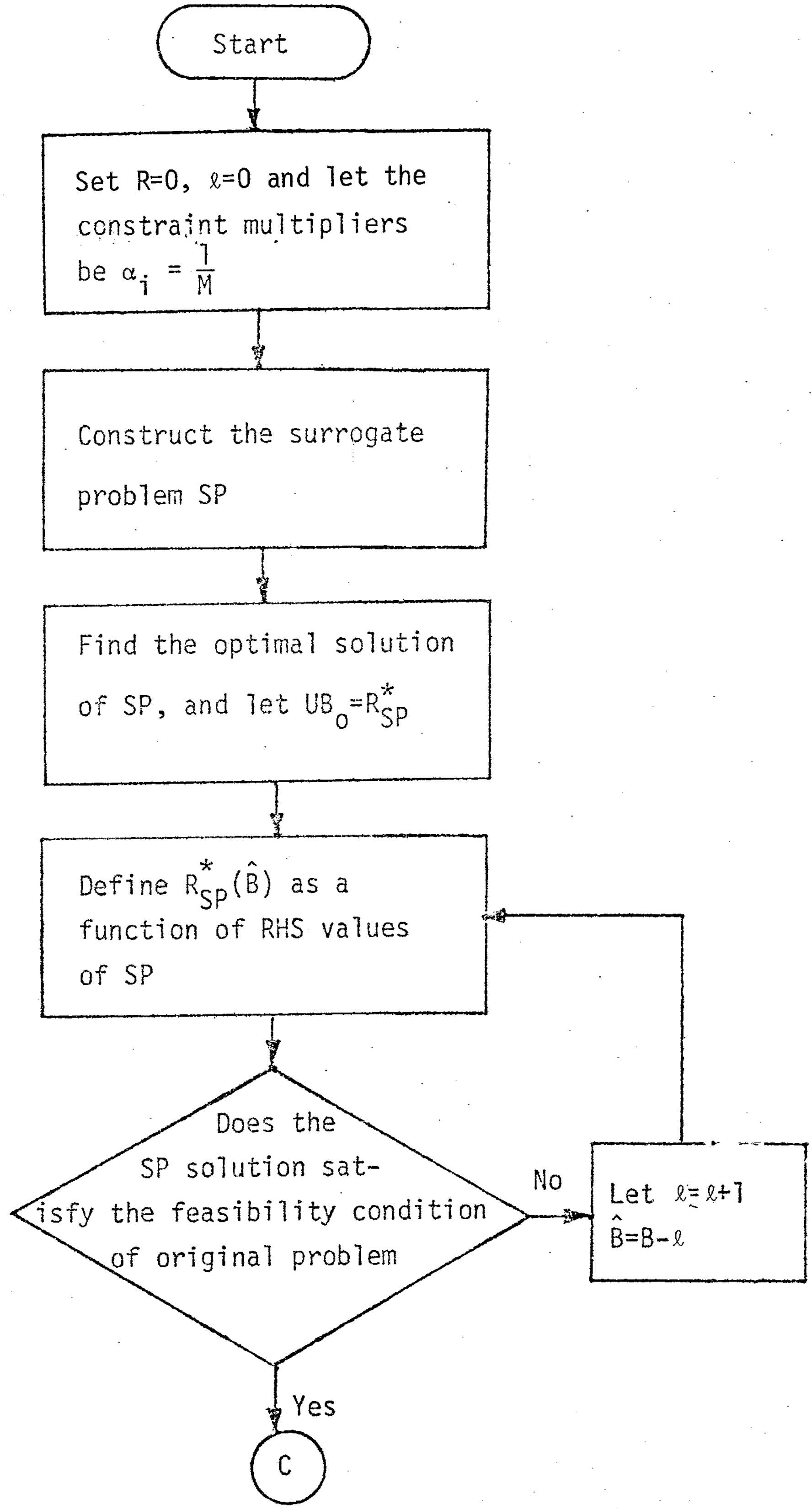

Figure A.1. Flow Chart for Hybrid Algorithm

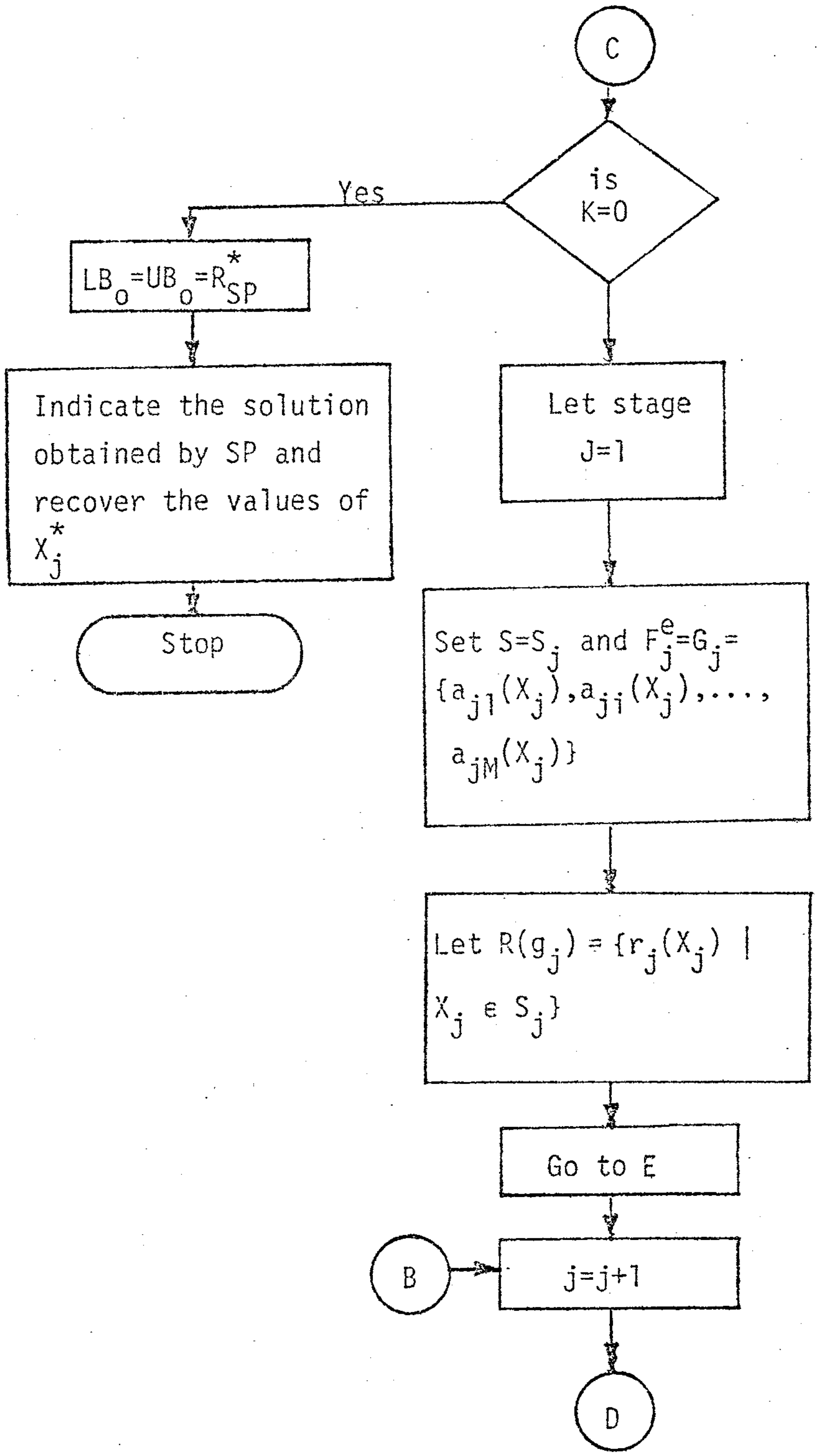

Figure A.1. (continued)

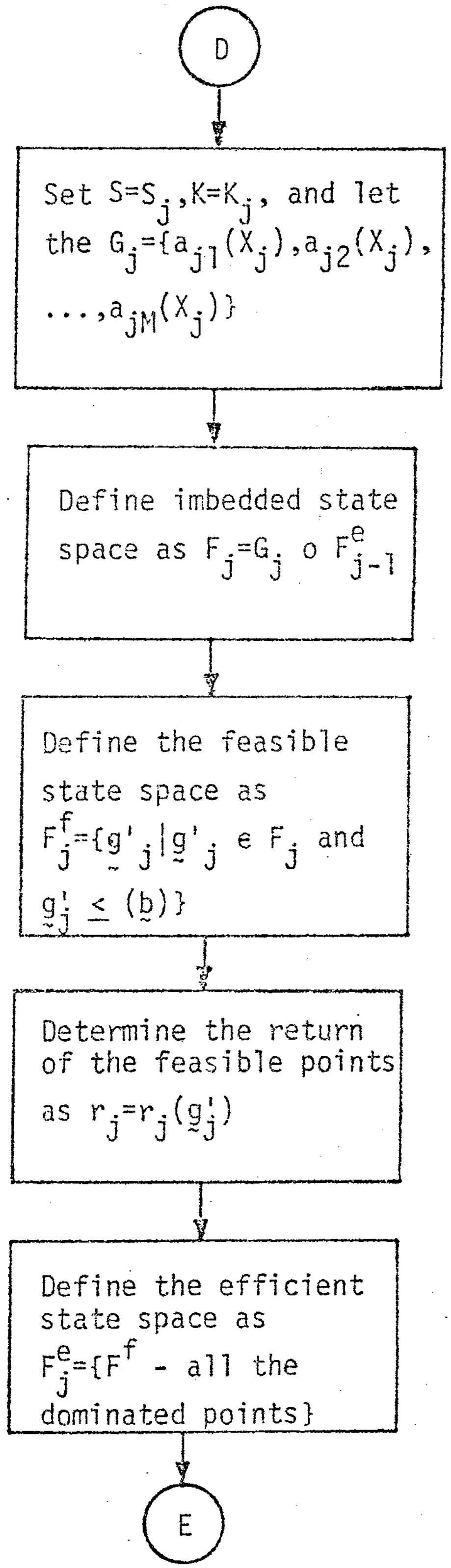

Figure A.1. (continued)

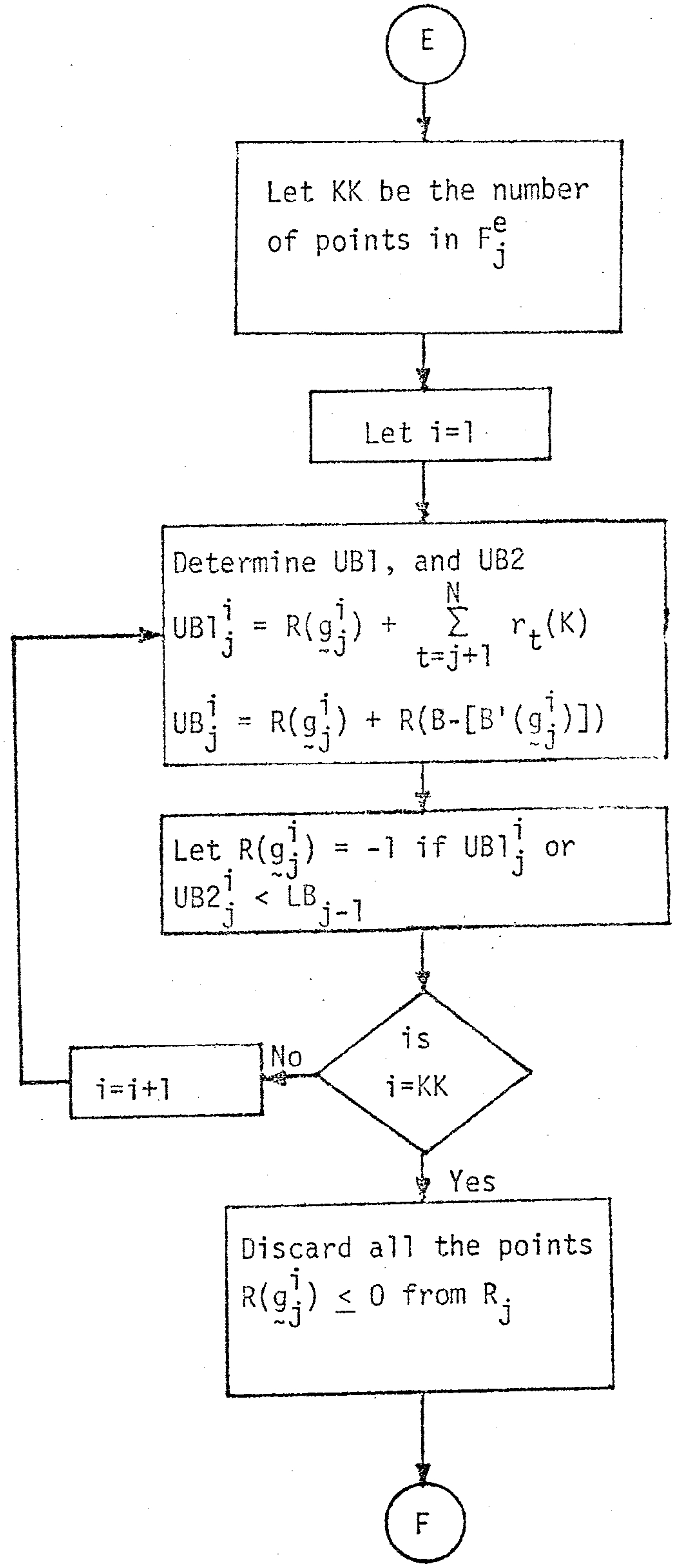

Figure A.1. (continued)

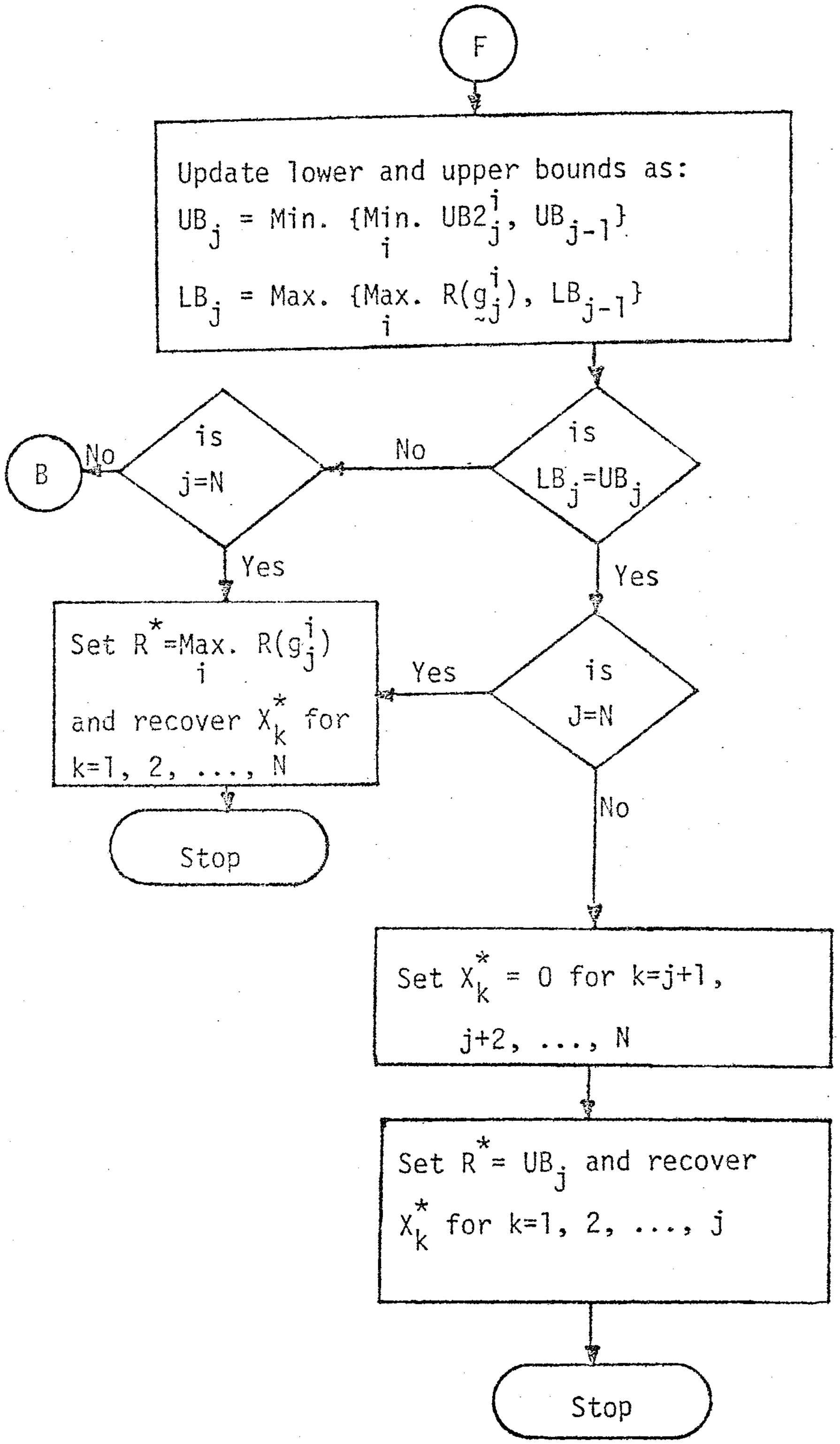

Figure A.1. (continued)

Step 26 - If j=N go to step 28, otherwise go to step 11

Step 27 - If j<N, set X_{j+1}, X_{j+2}, ..., $X_N = 0$ go to step 29, otherwise go to step 28.

Step 28 - Set $R^* = \text{Max } R(g_j^i)$ and recover X_k^* for k = 1, 2, ..., N and stop.

Step 29 - Set $R^* = UB_j$ and recover X_k^* for k = 1, 2, ..., j and stop.

Step 30 - Indicate that the solution is obtained by surrogate problem, set $R^* = UB_1$ and recover X_k^* for k = 1, 2, ..., N, and stop.

The information regarding the returns at each stage, and the corresponding state space solutions, is stored in a matrix similar to the matrix TS presented in Section A.2. Similarly, the T_j vectors are used for backtracking to obtain the optimal solution $\underset{\sim}{X}^*$.

A.5 Modifications of the Hybrid Algorithm for Large Scale NKP

In order to find a solution to a large NKP, a computationally efficient routine must be developed. The hybrid algorithm developed in Section A.4 provides optimal solutions to medium-size problems in a reasonable amount of computer time. However, the combinatorial increases in the state space solution will result in the enumeration of a very large state variable vector at each stage for large problems even with the reductions provided by the hybrid algorithm. This causes the algorithm to require a very large computational time. In this section, a heuristic procedure is developed and is incorporated in the hybrid algorithm in order to limit the growth of the state variable vector. This modification will increase the computational speed and convergence rate, and reduce the storage required by the algorithm. A mathematical support for the heuristic procedure is demonstrated in the form of a proposition. This modification suffers from the fact that the solution obtained by the modified method may not be an optimum. However, it should be a good solution to a previously intractable problem.

Preliminary investigation has shown that the increase in the size of the state variable vector is the primary cause of the increase in computation time. The size of this vector is a monotonically increasing function of K_j (number of alternatives in each stage) and N (number of stages). Therefore, to resolve the problem of slow computational speed, either N and/or K_j must be kept at reasonably low value(s). In general, the number of stages N cannot be reduced. However, the value of K_j can be limited by a systematic approach.

The modification of the hybrid algorithm is based primarily on the limitation of K_j values. In order to construct a good heuristic procedure, consider the monotonicity and separation properties of the objective function. Let $\underset{\sim}{X}^1 = (X_1^1, X_2^1, \ldots, X_N^1)$ be any feasible solution to the following problem:

Problem A.1

$$\text{Max. } R(\underset{\sim}{X}) = \sum_{j=1}^{N} r_j(X_j)$$

Subject to:

$$\sum_{j=1}^{N} a_{ij}(X_j) \leq b_i \qquad i = 1, 2, \ldots, M$$

X_j is contained in S_j

$S_j = \{1, 2, \ldots, K_j\}$

Let $X^2 = (X_1^2, X_2^2, \ldots, X_N^2)$ be an optimal solution obtained from the surrogated version of the above problem. As discussed in Section A.3, it is known that the optimal solution to problem A.1 satisfies the following relationship:

$$R(\underset{\sim}{X}^1) \leq R(\underset{\sim}{X}^*) \leq R(\underset{\sim}{X}^2) \tag{A.9}$$

Thus, a good heuristic procedure would be to find a solution which falls within the limits defined by A.9. Let S_j^1 be a set defined by

$$S_j^1 = \begin{cases} (X_j^1,\ X_j^1+1,\ X_j^1+2,\ \ldots\ X_j^2) & \text{for } X_j^1 < X_j^2 \\ X_j^1 & \text{for } X_j^1 \geq X_j^2 \end{cases}$$

Then the following proposition can be stated:

Proposition: Any feasible point $\underset{\sim}{X}^3$ belonging to the set S_j will have an objective funtion value within the limits defined by $F(\underset{\sim}{X}^1)$ as a satisfactory solution to the original problem, and $F(\underset{\sim}{X}^2)$ as the optimal solution of the surrogate Problem A.1. That is,

$$F(\underset{\sim}{X}^1) \leq F(\underset{\sim}{X}^3) \leq F(X^2) \qquad \forall\ \underset{\sim}{X}^3 \in S_j^1 \tag{A.10}$$

Proof - The following holds true based on the monotonicity and separation properties of the objective function:

(1) By the separation property:

$$R(\underset{\sim}{X}) = r_1(X_1) + r_2(X_2) + \ldots + r_N(X_N)$$

(2) by the monotonicity property:

$$r_j(X_j^m) \geq r_j(X_j^n) \qquad \forall\ X_j^m \geq X_j^n$$

therefore, the following relation holds:

$$\sum_{j=1}^{N} r_j\,(X_j^m) \geq \sum_{j=1}^{N} r_j(X_j^n) \qquad \forall\ j \text{ where } X_j^m \geq X_j^n$$

Since any point belonging to S_j^1 is greater than or equal to that of $\underset{\sim}{X}^1$.

$$R(\underset{\sim}{X}^1) \leq R(\underset{\sim}{X}^k) \qquad X_j^k \,\varepsilon\, S_j^1 \text{ and } j = 1, 2, \ldots, N$$

Further, since X^2 is the optimal point obtained from a relaxation (surrogate) of the constraints, any feasible point in S_j^1 has the property that;

$$R(\underset{\sim}{X}^k) \leq R(\underset{\sim}{X}^2) \quad \text{for } X_j^k \,\varepsilon\, S_j^1 \text{ and } \sum_{j=1}^{N} a_{ij}(X_j^k) \leq b_i$$

The above proposition has indicated that the choice of S_j^1 will limit the variation of the alternative solution and provide a good solution point if $\underset{\sim}{X}^1$ is calculated properly. Algorithm 2 presented in Section A.3 seems to have the ability of providing such a point. Thus, the only modification to the hybrid algorithm is to substitute S_j^1 for S_j.

APPENDIX B

PROGRAM DOCUMENTATION
OF
RAMS
STATE PLANNING PROGRAM

DESCRIPTION OF INPUT DATA

Card	Column	Variable	Description
A	6-65	Title	Name of the problem
B	6-10	NODS	No. of districts
	11-15	N	No. of segments in each districts
	16-20	M	No. of constraints
	21-25	KJ	No. of maintenance strategies
	26-35	CAPT	The capital available in (10,000) dollars
C	The names of maintenance strategies along with the overhead budget requirements for the strategies. KJ cards are read in.		
	6-25	STRT	Strategy Name
	26-40	CR_i	Overhead budget requirement for mile-foot
D	Names of distress type and maximum possible rating for the distress type. ND (number of distress type) cards are read in		
	6-25	DSTR	Distress type
	26-40	$RMAX_i$	Maximum possible rating for the distress i
E	Names of resources types and amounts of resources available per mile-ft. (M-1) cards to be read in.		
	6-25	RSRC	Resource type
	26-40	RS_i	Amount of i^{th} resource available per mile-foot
F	Resource requirements per mile-foot unit for each strategy.		
	6-10	L	Strategy number
	11-17	$RSRL_1$	1^{st} resource requirements for L^{th} strategy

18-24	$RSRL_2$	2^{nd} resource requirement for L^{th} strategy
74-80	$RSRL_{10}$	10^{th} resource requirement for L^{th} strategy.

The number cards to be read in are KJ. Currently the assumption is that there are at most 10 resources, excluding money.

G Potential gains of pavement rating for each distress type. When a certain maintenance strategy is applied. KJ cards are read in. Currently the assumption is that there are at most 10 distress types.

6-10	L	Strategy number
11-17	$DISTL_1$	Gain of rating for the 1st distress when L^{th} strategy is applied
18-24	$DIST_{L,2}$	Gain of rating for the 2nd distress for the L^{th} strategy is applied
74-80	$DIST_{L,10}$	Gain of rating for the 10th distress when Lth strategy is applied

H Probability of survival for different strategies and distress types for the length of the planning horizon. Length of planning horizon is assumed to be 20 years. (KJ x 20) cards are to be read in. The maximum number of distress types are assumed to be less than 10

6-7	L	Strategy number
8-10	M	Year in the planning horizon
11-17	$P_{L,M,1}$	Probability of survival for the Lth strategy in Mth year when 1st distress type is present
74-80	$P_{L,M,10}$	Probability of survival for the Lth strategy in Mth year when 10 th distress type is present.

Cards type 9, 10, and 11 are associated with each district.

I Highway segment information H cards are to be read in.

6-8	L	Segment number
9-10	HTYP (L)	Highway type of Lth segment
11-18	PAR1	Segment Name
19-30	PAR2	County Identification
31-38	PAR3	District Name
39-45	L1(L)	Length of Lth segment (miles)
46-52	L2(L)	Width of Lth segment (feet)
67-73	TRAF(L)	Traffic index of the Lth segment
74-80	ENVR(L)	Environmental index of the Lth segment

J Overhead budget available for the district

26-35	CC	Overhead budget in dollars

K Current pavement rating. H cards to be read in. It is assumed that there are at most 10 distress types

11-17	$RC_{I,1}$	Current pavement rating for the Ith highway if 1st distress type is present.
74 -80	$RC_{I,10}$	Current pavement rating for the Ith highway segment if 10th distress type is present.

APPENDIX C

INPUT AND OUTPUT
OF
THE EXAMPLE PROBLEM
FOR

STATE OPTIMAL FUND ALLOCATION PROGRAM II

INPUT DATA

FOR

EXAMPLE PROBLEM

```
000000000111111111122222222223333333333444444444455555555556666666666667
123456789012345678901234567890123456789012345678901234567890123456789890

A    STATE OPTIMAL FUND ALLOCATION - PROGRAM II
B        3  10    8    6          250
C    R1 SEAL COAT                214.000
C    R2 1.0 INCH OVERLAY         950.000
C    R3 2.5 INCH OVERLAY         925.000
C    R4 4.0 INCH OVERLAY        2000.000
C    R5 7.5 INCH OVERLAY        2600.000
C    R6 10. INCH OVERLAY        3549.000
D    RUTTING                      15.000
D    ALLIGATOR CRACKING           25.000
D    LONGTUD. CRACKING            25.000
D    TRANSVERSE CRACKING          20.000
D    FAILURES/MILE                40.000
D    SERVICEABILITY INDEX         50.000
E    ASPHALT CEMENT                4.600
E    GRADER                        0.700
E    LOADER                        0.340
E    TRUCK                         0.840
E    GRADER OPERATOR               0.700
E    LOADER OPERATOR               0.340
E    TRUCK OPERATOR                0.840
F        1  0.800  0.000  0.012  0.060  0.000  0.012  0.060
F        2  3.000  0.000  0.000  0.278  0.000  0.000  0.278
F        3  1.500  0.000  0.000  0.278  0.000  0.000  0.278
F        4  4.100  0.000  0.000  0.556  0.000  0.000  0.556
F        5  8.100  0.000  0.000  0.834  0.000  0.000  0.834
F        6  1.500 01.000  0.333  3.611  1.000  0.333  3.611
G        1  0.000 15.000 15.000 15.000 10.000  2.000
G        2 13.000 19.000 19.000 19.000 24.000 45.000
G        3 13.000 20.000 20.000 20.000 25.000 45.000
G        4 15.000 25.000 25.000 20.000 30.000 50.000
G        5 15.000 25.000 25.000 20.000 35.000 50.000
G        6 15.000 25.000 25.000 20.000 40.000 50.000
H     1  1  1.000  1.000  1.000  1.000  1.000  1.000
H     1  2  0.930  0.940  0.930  0.920  1.000  0.900
H     1  3  0.910  0.890  0.880  0.860  0.910  0.700
H     1  4  0.880  0.890  0.870  0.850  0.780  0.500
H     1  5  0.780  0.650  0.670  0.670  0.470  0.400
H     1  6  0.310  0.280  0.370  0.380  0.220  0.300
H     1  7  0.220  0.240  0.320  0.330  0.200  0.200
H     1  8  0.150  0.150  0.180  0.180  0.100  0.100
H     1  9  0.070  0.090  0.090  0.090  0.040  0.100
H     1 10  0.050  0.070  0.070  0.060  0.010  0.000
H     1 11  0.020  0.020  0.020  0.010  0.000  0.000
H     1 12  0.020  0.010  0.010  0.010  0.000  0.000
H     1 13  0.020  0.010  0.010  0.010  0.000  0.000
H     1 14  0.020  0.010  0.010  0.000  0.000  0.000
H     1 15  0.010  0.000  0.000  0.000  0.000  0.000
H     1 16  0.010  0.000  0.000  0.000  0.000  0.000
H     1 17  0.010  0.000  0.000  0.000  0.000  0.000
```

0000000001111111111222222222233333333334444444444555555555566666666667
1234567890123456789012345678901234567890123456789012345678901234567890

H	1	18	0.010	0.000	0.000	0.000	0.000	0.000
H	1	19	0.010	0.000	0.000	0.000	0.000	0.000
H	1	20	0.010	0.000	0.000	0.000	0.000	0.000
H	2	1	1.000	1.000	1.000	1.000	1.000	1.000
H	2	2	1.000	1.000	1.000	1.000	1.000	1.000
H	2	3	1.000	0.890	1.000	1.000	1.000	1.000
H	2	4	1.000	0.820	1.000	1.000	1.000	0.900
H	2	5	0.880	0.730	1.000	1.000	1.000	0.800
H	2	6	0.780	0.670	0.750	0.830	1.000	0.700
H	2	7	0.460	0.670	0.500	0.670	1.000	0.600
H	2	8	0.250	0.670	0.500	0.670	0.330	0.500
H	2	9	0.250	0.670	0.250	0.330	0.330	0.400
H	2	10	0.250	0.360	0.000	0.000	0.330	0.300
H	2	11	0.000	0.110	0.000	0.000	0.000	0.000
H	2	12	0.000	0.090	0.000	0.000	0.000	0.000
H	2	13	0.000	0.000	0.000	0.000	0.000	0.000
H	2	14	0.000	0.000	0.000	0.000	0.000	0.000
H	2	15	0.000	0.000	0.000	0.000	0.000	0.000
H	2	16	0.000	0.000	0.000	0.000	0.000	0.000
H	2	17	0.000	0.000	0.000	0.000	0.000	0.000
H	2	18	0.000	0.000	0.000	0.000	0.000	0.000
H	2	19	0.000	0.000	0.000	0.000	0.000	0.000
H	2	20	0.000	0.000	0.000	0.000	0.000	0.000
H	3	1	1.000	1.000	1.000	1.000	1.000	1.000
H	3	2	1.000	1.000	1.000	1.000	1.000	1.000
H	3	3	1.000	0.950	0.930	0.940	1.000	1.000
H	3	4	1.000	0.910	0.930	0.940	0.890	0.900
H	3	5	0.790	0.900	0.400	0.430	0.530	0.800
H	3	6	0.750	0.610	0.140	0.180	0.230	0.700
H	3	7	0.750	0.560	0.140	0.180	0.160	0.600
H	3	8	0.750	0.550	0.120	0.140	0.150	0.500
H	3	9	0.750	0.510	0.070	0.060	0.130	0.400
H	3	10	0.750	0.280	0.020	0.010	0.080	0.300
H	3	11	0.330	0.170	0.000	0.000	0.020	0.000
H	3	12	0.250	0.140	0.000	0.000	0.000	0.000
H	3	13	0.250	0.140	0.000	0.000	0.000	0.000
H	3	14	0.170	0.140	0.000	0.000	0.000	0.000
H	3	15	0.080	0.080	0.000	0.000	0.000	0.000
H	3	16	0.000	0.010	0.000	0.000	0.000	0.000
H	3	17	0.000	0.000	0.000	0.000	0.000	0.000
H	3	18	0.000	0.000	0.000	0.000	0.000	0.000
H	3	19	0.000	0.000	0.000	0.000	0.000	0.000
H	3	20	0.000	0.000	0.000	0.000	0.000	0.000
H	4	1	1.000	1.000	1.000	1.000	1.000	1.000
H	4	2	1.000	1.000	1.000	1.000	1.000	1.000
H	4	3	1.000	1.000	1.000	1.000	1.000	1.000
H	4	4	1.000	1.000	1.000	1.000	1.000	1.000
H	4	5	1.000	0.770	1.000	1.000	0.770	0.900
H	4	6	0.830	0.640	0.330	0.630	0.510	0.800
H	4	7	0.710	0.580	0.110	0.260	0.480	0.700

```
0000000001111111111222222222233333333334444444444555555555566666666667
1234567890123456789012345678901234567890123456789012345678901234567890
```

H	4	8	0.660	0.530	0.000	0.220	0.360	0.600
H	4	9	0.620	0.510	0.000	0.110	0.330	0.500
H	4	10	0.380	0.380	0.000	0.040	0.240	0.500
H	4	11	0.300	0.210	0.000	0.000	0.170	0.000
H	4	12	0.300	0.190	0.000	0.000	0.170	0.000
H	4	13	0.300	0.190	0.000	0.000	0.170	0.000
H	4	14	0.280	0.170	0.000	0.000	0.170	0.000
H	4	15	0.220	0.150	0.000	0.000	0.170	0.000
H	4	16	0.170	0.100	0.000	0.000	0.170	0.000
H	4	17	0.120	0.070	0.000	0.000	0.070	0.000
H	4	18	0.040	0.060	0.000	0.000	0.000	0.000
H	4	19	0.040	0.060	0.000	0.000	0.000	0.000
H	4	20	0.040	0.030	0.000	0.000	0.000	0.000
H	5	1	1.000	1.000	1.000	1.000	1.000	1.000
H	5	2	1.000	1.000	1.000	1.000	1.000	1.000
H	5	3	1.000	1.000	1.000	1.000	1.000	1.000
H	5	4	1.000	1.000	1.000	1.000	1.000	1.000
H	5	5	1.000	1.000	1.000	1.000	1.000	1.000
H	5	6	1.000	0.710	0.330	0.330	0.750	0.900
H	5	7	1.000	0.620	0.330	0.330	0.590	0.900
H	5	8	1.000	0.440	0.280	0.280	0.500	0.800
H	5	9	1.000	0.290	0.170	0.170	0.500	0.700
H	5	10	1.000	0.290	0.170	0.170	0.480	0.600
H	5	11	0.670	0.290	0.170	0.170	0.250	0.000
H	5	12	0.670	0.170	0.170	0.170	0.250	0.000
H	5	13	0.670	0.140	0.170	0.170	0.250	0.000
H	5	14	0.670	0.140	0.170	0.170	0.250	0.000
H	5	15	0.220	0.120	0.170	0.170	0.200	0.000
H	5	16	0.000	0.000	0.170	0.170	0.000	0.000
H	5	17	0.000	0.000	0.170	0.170	0.000	0.000
H	5	18	0.000	0.000	0.170	0.170	0.000	0.000
H	5	19	0.000	0.000	0.170	0.170	0.000	0.000
H	5	20	0.000	0.000	0.170	0.170	0.000	0.000
H	6	1	1.000	1.000	1.000	1.000	1.000	1.000
H	6	2	1.000	1.000	1.000	1.000	1.000	1.000
H	6	3	1.000	1.000	1.000	1.000	1.000	1.000
H	6	4	1.000	1.000	1.000	1.000	1.000	0.900
H	6	5	1.000	1.000	1.000	1.000	1.000	0.800
H	6	6	0.720	0.490	1.000	1.000	0.470	0.700
H	6	7	0.670	0.360	1.000	1.000	0.360	0.600
H	6	8	0.580	0.360	1.000	1.000	0.320	0.500
H	6	9	0.500	0.360	0.650	0.650	0.270	0.400
H	6	10	0.500	0.290	0.600	0.600	0.270	0.300
H	6	11	0.360	0.270	0.600	0.600	0.270	0.000
H	6	12	0.330	0.270	0.600	0.600	0.200	0.000
H	6	13	0.330	0.270	0.530	0.510	0.180	0.000
H	6	14	0.280	0.270	0.400	0.400	0.180	0.000
H	6	15	0.170	0.210	0.380	0.380	0.150	0.000
H	6	16	0.170	0.190	0.210	0.200	0.090	0.000
H	6	17	0.170	0.190	0.200	0.000	0.090	0.000

```
0000000001111111111222222222233333333334444444444555555555566666666667
1234567890123456789012345678901234567890123456789012345678901234567890


H     6 18  0.170  0.180  0.200  0.000  0.090  0.000
H     6 19  0.170  0.110  0.200  0.000  0.090  0.000
H     6 20  0.170  0.090  0.200  0.000  0.090  0.000
I      1 1DIST                          3.309 36.000
I      2 1DIST                          2.266 12.000
I      3 2DIST                          3.818 12.000
I      4 1DIST                          2.512 12.000
I      5 1DIST                          4.712 36.000
I      6 2DIST                          1.663 12.000
I      7 1DIST                          3.572 24.000
I      8 1DIST                          2.462 12.000
I      9 2DIST                          2.625 12.000
I     10 1DIST                          1.590 12.000
J    OVERHEAD BUDGET          705800
K        1 15.000 25.000 20.000 25.000 30.000 40.000
K        2  5.000 20.000 20.000 25.000 30.000  0.000
K        3 15.000 10.000  5.000 25.000 30.000 10.000
K        4  5.000 25.000 25.000 25.000 30.000 40.000
K        5  5.000 10.000 15.000  5.000 30.000  0.000
K        6 15.000 25.000 15.000 25.000 30.000 40.000
K        7 15.000 25.000 25.000 25.000 30.000 40.000
K        8 15.000 20.000 20.000 25.000 30.000 40.000
K        9  5.000 25.000 20.000  5.000 30.000 10.000
K       10  5.000  5.000  5.000  5.000 30.000 10.000
I      1 1DIST                          3.317 36.000
I      2 1DIST                          2.313 12.000
I      3 2DIST                          4.029 12.000
I      4 1DIST                          3.356 12.000
I      5 1DIST                          3.876 24.000
I      6 1DIST                          1.054 12.000
I      7 1DIST                          3.439 24.000
I      8 2DIST                          3.151 12.000
I      9 2DIST                          3.948 24.000
I     10 1DIST                          3.331 12.000
J    OVERHEAD BUDGET          837700
K        1  5.000 10.000 15.000 25.000 30.000 40.000
K        2 15.000 25.000  5.000 25.000 30.000  0.000
K        3  5.000 20.000 25.000 25.000 30.000 40.000
K        4  5.000 25.000 20.000 25.000 30.000  0.000
K        5 15.000 10.000 25.000 25.000 30.000 10.000
K        6 15.000 25.000  5.000 25.000 30.000 40.000
K        7 15.000 25.000 25.000 25.000 30.000 10.000
K        8 15.000 10.000 25.000 25.000 30.000  0.000
K        9 15.000 20.000 15.000 25.000 30.000 40.000
K       10 15.000 25.000 25.000 25.000 30.000 40.000
I      1 1DIST                          3.324 36.000
I      2 1DIST                          2.360 12.000
I      3 2DIST                          4.240 12.000
I      4 1DIST                          4.200 12.000
I      5 1DIST                          3.040 12.000
```

```
0000000001111111111222222222233333333334444444444555555555566666666667
1234567890123456789012345678901234567890123456789012345678901234567890


I      6 1DIST                          4.445 12.000
I      7 1DIST                          3.307 24.000
I      8 1DIST                          3.839 12.000
I      9 2DIST                          1.270 12.000
I     10 1DIST                          1.073 36.000
J    OVERHEAD BUDGET        1002300
K        1 15.000 25.000 25.000 25.000 30.000  0.000
K        2 15.000 10.000 20.000 25.000 30.000 10.000
K        3 15.000 25.000 25.000 25.000 30.000 40.000
K        4  5.000 25.000 20.000 25.000 30.000 40.000
K        5 15.000 20.000 20.000  5.000 30.000 40.000
K        6 15.000 25.000  5.000 25.000 30.000 40.000
K        7  5.000 25.000 20.000 25.000 30.000 10.000
K        8 15.000 25.000 25.000  5.000 30.000 40.000
K        9 15.000 10.000 20.000 25.000 30.000  0.000
K       10 15.000  5.000  5.000 25.000 30.000 40.000
```

OUTPUT OF

EXAMPLE PROBLEM

TEXAS TRANSPORTATION INSTITUTE
TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77843

REHABILITATION AND MAINTENANCE SYSTEMS
STATE OPTIMAL FUND ALLOCATION - PROGRAM II

	BUDGET LEVELS			
DISTRICT	MAXIMUM	MINIMUM	OPTIMUM	BENEFIT
1	710000	690000	710000	2187.
2	840000	770000	770000	1896.
3	1010000	930000	1010000	1803.
			2490000	5886.

TEXAS TRANSPORTATION INSTITUTE
TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77843

REHABILITATION AND MAINTENANCE SYSTEMS
STATE OPTIMAL FUND ALLOCATION - PROGRAM II

DISTRICT	SEGMENT	STRATEGY
1	1	5
1	2	5
1	3	2
1	4	5
1	5	3
1	6	4
1	7	5
1	8	2
1	9	1
1	10	1
2	1	5
2	2	5
2	3	2
2	4	2
2	5	2
2	6	4
2	7	5
2	8	4
2	9	1
2	10	1
3	1	5
3	2	5
3	3	2
3	4	3
3	5	3
3	6	6
3	7	5
3	8	4
3	9	1
3	10	1

APPENDIX D

LISTING OF COMPUTER PROGRAM

```
C
C      ****************************************************************
C
C      REHABILATATION AND MAINTENANCE SYSTEM
C
C      STATE OPTIMAL FUND ALLOCATION - PROGRAM II
C
C      TEXAS TRANSPORTATION INSTITUTE
C      TEXAS A&M UNIVERSITY
C      COLLEGE STATION, TEXAS 77843
C
C
C      AUTHORS                 : GHASHEMI F. TARI
C                                CHIYYARATH V. SHANMUGHAM
C
C      INSTALLATION            : AMDAHL 470V/6
C                                DATA PROCESSING CENTER
C                                TEXAS A&M UNIVERSITY
C
C
C      LANGUAGE                : FORTRAN IV
C
C      DATE                    : SEPTEMBER 1980
C
C      ****************************************************************
C
      COMMON /A1/ NODS, N, M, KJ, CC, CAPT
      COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
      COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
      COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
      COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
      COMMON /C1/ IKL(10), BAR(10), X(10)
      COMMON /C2/ IKU(10), IOD(10), FM(10)
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
      COMMON /E1/ HTYP(10), L1(10), L2(10)
      COMMON /E2/ PAR1(10,2), PAR2(10,3), PAR3(10,2)
      COMMON /E3/ PAR4(10), PAR5(10), TRAF(10), ENVR(10)
      COMMON /E4/ RC(10,6)
      INTEGER      CAPT, UNIT, TOTC, X, TRACE
      INTEGER      TRC, TRC1, TRC2, TRC3, TLV
      INTEGER      HTYP, PAR1, PAR2, PAR3, D
      REAL         LB, IMS, IMS1, L1, L2
  500 FORMAT ( 5X, 15A4 )
  510 FORMAT ( 5X, 4I5, I10 )
  520 FORMAT ( 5X, I3, I2, 7A4, 6F7.3 )
  530 FORMAT ( 25X, F10.0 )
  540 FORMAT ( 10X, 10F7.3 )
      UNIT     = 10000
      TLV      = 100
      READ  (5,500) TITL
      WRITE (6,500) TITL
      READ  (5,510) NODS, N, M, KJ, CAPT
```

```
      WRITE (6,510) NODS, N, M, KJ, CAPT
      CALL DATAIN
      DO 2000 D = 1, NODS
      DO 1100 I = 1, N
      READ  (5,520) L, HTYP(L), ( PAR1(L,J), J = 1, 2 ),
     1            ( PAR2(L,J), J = 1, 3 ), ( PAR3(L,J), J = 1, 2 ),
     2            L1(L), L2(L), PAR4(L), PAR5(L), TRAF(L), ENVR(L)
      WRITE (6,520) L, HTYP(L), ( PAR1(L,J), J = 1, 2 ),
     1            ( PAR2(L,J), J = 1, 3 ), ( PAR3(L,J), J = 1, 2 ),
     2            L1(L), L2(L), PAR4(L), PAR5(L), TRAF(L), ENVR(L)
 1100 CONTINUE
      READ  (5,530) CC
      DO 1200 I = 1, N
      READ  (5,540) ( RC(I,J), J = 1, KJ )
      WRITE (6,540) ( RC(I,J), J = 1, KJ )
 1200 CONTINUE
      CALL OBJFCN
      DO 2345 I = 1, N
      WRITE (6,666)
      WRITE (6,667) ( R(I,J), J = 1, KJ )
      WRITE (6,666)
      DO 1234 J = 1, KJ
      WRITE (6,667) ( C(K,I,J), K = 1, M )
 1234 CONTINUE
 2345 CONTINUE
  666 FORMAT ( 1X )
  667 FORMAT ( 10F12.5 )
      ND        = D
      LB=0.
      UB=0.
      ZIBR(ND)=0.
C
C     TO CALCULATE THE INITIAL LOWER AND UPPER BCUNDS
C
      DO 10 J=2,N
      UB=UB+R(J,KJ)
   10 CONTINUE
      DO 11 J=1,N
      DO 11 K=1,KJ
      S=0.
      DO 12 I=1,M
      S=S+C(I,J,K)
   12 CONTINUE
      ICS(J,K)=S/M
   11 CONTINUE
      TIB=0.
      DO 13 I=1,M
      TIB=TIB+B(I)
   13 CONTINUE
C     DO 3456 J = 1, N
C     WRITE (6,669) ( ICS(J,K), K = 1, KJ )
C3456 CONTINUE
C 669 FORMAT ( '    ICS ', 10I10 )
      IB=TIB/M+1
C     WRITE (6,670) TIB, IB
```

```
C 670 FORMAT ( '  TIB', F12.3, '    IB ', I5 )
      CALL RETRN1
      INFS      = 0
      CALL SERCH1 ( INFS )
      IF ( INFS .EQ. 0 ) GO TO 4738
      WRITE (6,689) ND
  689 FORMAT ( '  DISTRICT', I3, ' IS INFEASIBLE ' )
      GO TO 2000
 4738 CONTINUE
C     PRINT,LB,UB
      IF(ICHEK.EQ.0) GO TO 15
      CALL IMBEDD
      IOLP=IR
      IF(JON.GT.0) GO TO 1000
      Z=IMS1(IR,M1)
      DO 90 JJ=1,N
      J=N-JJ+1
      X(J)=TRACE(IR,J,1)
      IR=TRACE(IR,J,2)
   90 CONTINUE
      GO TO 16
 1000 Z=IMS1(IR,M1)
      DO 1001 JJ=1,JON
      J=JON-JJ+1
      X(J)=TRACE(IR,J,1)
      IR=TRACE(IR,J,2)
 1001 CONTINUE
      JON=JON+1
      DO 1002 JJ=JON,N
      X(JJ)=1
 1002 CONTINUE
      GO TO 16
   15 Z=FM(N)
      DO 60 J=1,N
      X(J)=IOD(J)
   60 CONTINUE
C     PRINT 202
C 202 FORMAT(20X,'SUROGATE SOLUTION',//)
C     PRINT 203,Z
C     PRINT 206
      DO 17 J=1,N
C     PRINT 205,J,X(J)
   17 CONTINUE
      MAXA(ND)=AICH+1.
      MINA(ND)=AICH+1.
      ZIBR(ND)=-10.
      BII(ND,1)=Z
      NALT(ND)=1
      ICZ(ND,1)=1
      DO 30 J=1,N
      ICY(ND,J)=X(J)
   30 CONTINUE
      GO TO 2000
C 16  PRINT 203,Z
   16 CONTINUE
```

```
C 203 FORMAT(20X,'OBJECTIVE FUNCTION VALUE=',F12.5,//)
C     PRINT 206
C 206 FORMAT(12X,'OPT. DEC. VALUE',/)
      DO 18 J=1,N
C     PRINT 205,J,X(J)
C 205 FORMAT(12X,I3,12X,I3)
  18  CONTINUE
      IR=IOLP
      LOC=M1-1
      MINA(ND)=IMS1(1,LOC)+1
      KOK=MINA(ND)
      ICZ(ND,1)=KOK-MINA(ND)+1
      ICY(ND,1)=1
      BII(ND,1)=IMS(1,M1)
      K=1
      DO 21 J=1,IR
      SUB=IMS1(J,LOC)
      IF(KOK-SUB) 22,23,23
  23  BII(ND,K)=IMS1(J,M1)
      GO TO 21
  22  KOK=IMS1(J,LOC)+1
      K=K+1
      ICZ(ND,K)=KOK-MINA(ND)+1
      ICY(ND,K)=J
      BII(ND,K)=IMS1(J,M1)
  21  CONTINUE
      NALT(ND)=K
      MAXA(ND)=ICZ(ND,K)+MINA(ND)-1
 2000 CONTINUE
      UNIT=10000
      CALL RESULT
      STOP
      END




      SUBROUTINE  RESULT
      COMMON /A1/ NODS, N, M, KJ, CC, CAPT
      COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
      COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
      COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
      COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
      COMMON /C1/ IKL(10), BAR(10), X(10)
      COMMON /C2/ IKU(10), IOD(10), FM(10)
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
      DIMENSION   IY( 5,10)
      INTEGER     CAPT, UNIT, TOTC, X, TRACE
      INTEGER     TRC, TRC1, TRC2, TRC3, TLV
      REAL        LB, IMS, IMS1
  600 FORMAT ( 1H1, //// )
  610 FORMAT ( 51X, 30HTEXAS TRANSPORTATION INSTITUTE, /,
```

```
     1              56X, 20HTEXAS A&M UNIVERSITY, /,
     2              52X, 28HCOLLEGE STATION, TEXAS 77843, //,
     3              47X, 38HREHABILITATION AND MAINTENANCE SYSTEMS, /,
     4              45X, 42HSTATE OPTIMAL FUND ALLOCATION - PROGRAM II, // )
  620 FORMAT ( 59X, 13HBUDGET LEVELS, /,
     1              44X, 44HDISTRICT  MAXIMUM  MINIMUM  OPTIMUM  BENEFIT, /
  630 FORMAT ( 47X, I2, 3X, I9, I9, I9, F9.0, / )
  631 FORMAT ( /, 70X, I9, F9.0)
  635 FORMAT ( /, 44X, 43('-'), / )
  640 FORMAT ( 53X, 27HDISTRICT  SEGMENT  STRATEGY, /,
     1              53X, 26('-'), /  )
  650 FORMAT ( 55X, I2, 6X, I2, 8X, I2 )
  660 FORMAT ( /, 53X, 26('-'), /  )
      WRITE (6,600)
      WRITE (6,610)
      WRITE (6,620)
      CAPTN=0.
      DO 826 I=1,NODS
      CAPTN=CAPTN+MINA(I)
  826 CONTINUE
      CAPT=CAPT-CAPTN+NODS
      DO 806 I=1,NODS
      NOAL=NALT(I)
      DO 806 J=1,NOAL
C     PRINT 506,I,ICZ(I,J),BII(I,J)
 806  CONTINUE
C506  FORMAT(32X,I3,8X,I8,5X,F10.2)
      INFIS=-99999
      DO 4 IN=1,NODS
      MAX(IN)=MAXA(IN)-MINA(IN)+1
      MIN(IN)=1
      CALL RETRN2(IN)
    4 CONTINUE
      CALL SERCH2
C
C                               OUT PUT
C
      TOTC=0
      DO 30 I=1,NODS
      MIN(I)=MINA(I)*UNIT
      MAX(I)=MAXA(I)*UNIT
      J=IOD(I)
      ICZ(I,J)=(ICZ(I,J)+MINA(I)-1)*UNIT
      TOTC=TOTC+ICZ(I,J)
      IF(I.EQ.1) GO TO 31
      II=I-1
      RIGH=FM(II)
      BI(I,J)=FM(I)-RIGH
      GO TO 32
   31 BI(I,J)=FM(I)
   32 CONTINUE
      WRITE (6,630) I,MAX(I),MIN(I),ICZ(I,J),BI(I,J)
      IF(ZIBR(I).EQ.-10.) GO TO 855
      IR=ICY(I,J)
      DO 850 JJ=1,N
```

```
      NIN=N-JJ+1
      IY(I,NIN)=TRACE(IR,NIN,1)
      IR=TRACE(IR,NIN,2)
  850 CONTINUE
      GO TO 30
 855  DO 856 J=1,N
      IY(I,J)=ICY(I,J)
  856 CONTINUE
  30  CONTINUE
      TOTR=FM(NODS)
      WRITE (6,635)
      WRITE (6,631) TOTC, TOTR
      WRITE (6,600)
      WRITE (6,610)
      WRITE (6,640)
      DO 1236 I = 1, NODS
      DO 1235 J = 1, N
      WRITE (6,650) I, J, IY(I,J)
 1235 CONTINUE
 1236 CONTINUE
      WRITE (6,660)
      WRITE (6,600)
      RETURN
      END
```

```
      SUBROUTINE  RETRN2(I)
      COMMON /A1/ NODS, N, M, KJ, CC, CAPT
      COMMON /A2/ NO, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
      COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
      COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
      COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
      COMMON /C1/ IKL(10), BAR(10), X(10)
      COMMON /C2/ IKU(10), IOD(10), FM(10)
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
      INTEGER     CAPT, UNIT, TOTC, X, TRACE
      INTEGER     TRC, TRC1, TRC2, TRC3, TLV
      REAL        LB, IMS, IMS1
      IF(I.GT.1) GO TO 3
      KK=MAX(I)
      ISS=MIN(I)-1
      NOAL=NALT(I)
      K=MIN(I)-1
      IF(NOAL.EQ.1) GO TO 40
      NIL=NOAL-1
      GO TO 41
  40  NIL=1
  41  DO 2 JOZ=1,NIL
      KO1=ICZ(I,JOZ)
      JOZY=JOZ+1
```

```
      KO2=ICZ(I,JOZY)-1
      DO 2 J=KO1,KO2
      IS(J)=K+1
      K=IS(J)
      ISK=K
      F(I,ISK)=BII(I,JOZ)
      ID(I,ISK)=JOZ
    2 CONTINUE
      J=ICZ(I,NOAL)
      IS(J)=K+1
      K=IS(J)
      ISK=K
      F(I,ISK)=BII(I,NOAL)
      ID(I,ISK)=NOAL
      ISS=MIN(I)
      GO TO 16
    3 IF(I.EQ.NODS) GO TO 15
      KK=0
      ISS=0
      DO 4 J=1,I
      KK=MAX(J)+KK
      ISS=MIN(J)+ISS
    4 CONTINUE
      IF(KK.LT.CAPT) GO TO 50
      KK=CAPT
   50 ISK=ISS-1
      DO 5 K=ISS,KK
      IS(K)=K
      ISK=IS(K)
      F(I,ISK)=INFIS
      ID(I,ISK)=-1
      NOAL=NALT(I)
      DO 6 J=1,NOAL
      ICIJ=ICZ(I,J)
      II=I-1
      ICH=ISK-ICIJ
      MAXCH=0
      MINCH=0
      DO 7 L=1,II
      MINCH=MIN(L)+MINCH
      MAXCH=MAX(L)+MAXCH
    7 CONTINUE
      IF(MAXCH.LE.CAPT) GO TO 51
      MAXCH=CAPT
   51 IF(ICH.LT.MINCH) GO TO 5
      IF(ICH.GT.MAXCH) GO TO 8
      FIF=BII(I,J)+F(II,ICH)
      IF(FIF.LT.F(I,ISK)) GO TO 6
      F(I,ISK)=FIF
      ID(I,ISK)=J
      GO TO 6
    8 KA=MAXCH
      F(I,ISK)=BII(I,J)+F(II,KA)
      ID(I,ISK)=J
    6 CONTINUE
```

```
    5   CONTINUE
        GO TO 16
   15   CONTINUE
        NOAL=NALT(I)
        DO 9 J=1,NOAL
        IF(CAPT.GT.300) GO TO 61
        ISK=CAPT
        GO TO 62
   61   ISK=249
        CAPT=ISK
   62   F(I,ISK)=INFIS
        ID(I,ISK)=-1
        II=I-1
        ICIJ=ICZ(I,J)
        ICH=ISK-ICIJ
        MAXCH=0
        MINCH=0
        DO 11 L=1,II
        MINCH=MIN(L)+MINCH
        MAXCH=MAX(L)+MAXCH
   11   CONTINUE
        IF(MAXCH.LE.CAPT) GO TO 52
        MAXCH=CAPT
   52   IF(ICH.LT.MINCH) GO TO 9
        IF(ICH.GT.MAXCH) GO TO 12
        FIF=BII(I,J)+F(II,ICH)
        IF(FIF.LT.F(I,ISK)) GO TO 9
        F(I,ISK)=FIF
        ID(I,ISK)=J
        GO TO 9
   12   KA=MAXCH
        F(I,ISK)=BII(I,J)+F(II,KA)
        ID(I,ISK)=J
    9   CONTINUE
   16   ILOW(I)=ISS
        IHIG(I)=KK
        RETURN
        END
```

```
        SUBROUTINE  SERCH2
        COMMON /A1/ NODS, N, M, KJ, CC, CAPT
        COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
        COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
        COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
        COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
        COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
        COMMON /C1/ IKL(10), BAR(10), X(10)
        COMMON /C2/ IKU(10), IOD(10), FM(10)
        COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
        COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
        COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
        INTEGER     CAPT, UNIT, TOTC, X, TRACE
```

```
      INTEGER      TRC, TRC1, TRC2, TRC3, TLV
      REAL         LB, IMS, IMS1
      I=NODS
      FM(I)=F(I,CAPT)
      IOD(I)=ID(I,CAPT)
      J=IOD(I)
      KEK=NODS-1
      ISIS=CAPT-ICZ(I,J)
      DO 2 I=1,KEK
      II=NODS-I
      IF(ISIS.GT.IHIG(II)) GO TO 3
      K=ISIS
      FM(II)=F(II,K)
      IOD(II)=ID(II,K)
      J=IOD(II)
      ISIS=ISIS-ICZ(II,J)
      GO TO 2
    3 K=IHIG(II)
      FM(II)=F(II,K)
      IOD(II)=ID(II,K)
      J=IOD(II)
      ISIS=ISIS-ICZ(II,J)
    2 CONTINUE
      RETURN
      END




      SUBROUTINE  RETRN1
      COMMON /A1/ NODS, N, M, KJ, CC, CAPT
      COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
      COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
      COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
      COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
      COMMON /C1/ IKL(10), BAR(10), X(10)
      COMMON /C2/ IKU(10), IOD(10), FM(10)
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
      INTEGER      CAPT, UNIT, TOTC, X, TRACE
      INTEGER      TRC, TRC1, TRC2, TRC3, TLV
      REAL         LB, IMS, IMS1
      FMAX=0.
      LM         = 0
      IF ( IB .LE. 300 ) GO TO 1200
      WRITE (6,667)
      IB         = 300
 1200 CONTINUE
  667 FORMAT ( ' **** IB GT 300. SET IB = 300' )
      DO 11 J=1,N
      DO 11 L=1,IB
      ID(J,L)=1
      F(J,L)=-999.
```

```
      DO 12 K=1,KJ
      ICSI=ICS(J,K)
      ISL=L-1
      IF(ISL.LT.ICSI) GO TO 11
      IF(J.GT.1) GO TO 13
      FI=R(J,K)
      GO TO 14
   13 JJ=ISL-ICSI+1
      JK=J-1
      FI=R(J,K)+F(JK,JJ)
   14 IF(FI.LT.F(J,L)) GO TO 12
      F(J,L)=FI
      ID(J,L)=K
   12 CONTINUE
      IF(J.LT.N) GO TO 11
      IF(FMAX.GE.F(J,L)) GO TO 11
      FMAX=F(J,L)
      IDMAX=ID(J,L)
      LM=L
   11 CONTINUE
C     DO 1234 L = 1, IB
C     WRITE (6,777) ( ID(J,L), J = 1, N ), ( F(J,L), J = 1, N )
C1234 CONTINUE
C 777 FORMAT ( 10I5, 10F7.1 )
C     WRITE (6,778) LM
C 778 FORMAT ( '  LM  ', I5 )
      IF ( LM .GT. 0 ) RETURN
      WRITE (6,666) LM
  666 FORMAT ( 1X, '?????????????? LM IS ', I3 )
      STOP
      END




      SUBROUTINE  SERCH1 ( INFS )
      COMMON /A1/ NODS, N, M, KJ, CC, CAPT
      COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
      COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
      COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
      COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
      COMMON /C1/ IKL(10), BAR(10), X(10)
      COMMON /C2/ IKU(10), IOD(10), FM(10)
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
      INTEGER     CAPT, UNIT, TOTC, X, TRACE
      INTEGER     TRC, TRC1, TRC2, TRC3, TLV
      REAL        LB, IMS, IMS1
      IND=0
   16 LM=LM-IND
      KK=ID(N,LM)
      FM(N)=F(N,LM)
      IOD(N)=ID(N,LM)
```

```
      ICC=LM-ICS(N,KK)
      J = 0
C     WRITE (6,666) J, N, KK, ICC, FM(N)
      NN = N - 1
      DO 10 J = 1, NN
      JJ=N-J
C 666 FORMAT ( ' J = ', I5, '  JJ = ', I5, '  KK = ', I5,
C    1          '  ICC = ', I5, '  FM ', F10.1 )
      FM(JJ)=F(JJ,ICC)
      IOD(JJ)=ID(JJ,ICC)
      KK=IOD(JJ)
      ICC=ICC-ICS(JJ,KK)
C     WRITE (6,666) J, JJ, KK, ICC, FM(JJ)
   10 CONTINUE
      IF(IND.GT.0) GO TO 20
      DO 21 J=1,N
      IKU(J)=IOD(J)
   21 CONTINUE
   20 CONTINUE
      DO 11 I=1,M
      AICH=0.
      DO 12 J=1,N
      IK=IOD(J)
      AICH=AICH+C(I,J,IK)
   12 CONTINUE
      IF(AICH.GT.B(I)) GO TO 13
   11 CONTINUE
      IF(IND.GT.0) GO TO 14
      ICHEK=0
      DO 22 J=1,N
      IKL(J)=IKU(J)
   22 CONTINUE
      LB=FM(N)
      UB=LB
      GO TO 17
   14 ICHEK=1
      DO 23 J=1,N
      IKL(J)=IOD(J)
   23 CONTINUE
      LB=FM(N)
      GO TO 17
   13 IF(IND.GT.0) GO TO 15
      UB=FM(N)
   15 IND=IND+1
      IF ( LM .GT. IND ) GO TO 16
      INFS      = 1
      ICHEK = 1
   17 RETURN
      END



      SUBROUTINE  IMBEDD
      COMMON /A1/ NODS, N, M, KJ, CC, CAPT
```

```
      COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /A3/ M1, JON, FMAX, IDMAX, LM, LB, UB
      COMMON /B1/ MIN( 5), MINA( 5), ILOW( 5), NALT( 5)
      COMMON /B2/ MAX( 5), MAXA( 5), IHIG( 5), ZIBR( 5)
      COMMON /B3/ BI( 5,110), BII( 5,110), ICY( 5,110), ICZ( 5,110)
      COMMON /C1/ IKL(10), BAR(10), X(10)
      COMMON /C2/ IKU(10), IOD(10), FM(10)
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /D1/ IS(500), TRC(500,2), IMS(500,9), IMS1(500,9)
      INTEGER      CAPT, UNIT, TOTC, X, TRACE
      INTEGER      TRC, TRC1, TRC2, TRC3, TLV
      REAL         LB, IMS, IMS1
      J=1
      JON=0
      UCB=UB
      DO 10 K=1,KJ
      IR=K
      DO 11 I=1,M
      IMS1(K,I)=C(I,J,K)
      IMS(K,I)=C(I,J,K)
   11 CONTINUE
      M1=M+1
      IMS1(K,M1)=R(J,K)
      IMS(K,M1)=R(J,K)
      TRC(IR,1)=K
      TRC(IR,2)=1
      TRACE(IR,1,1)=K
      TRACE(IR,1,2)=1
   10 CONTINUE
      DO 12 J=2,N
      DO 18 IRR=1,IR
      TRC(IRR,1)=1
      TRC(IRR,2)=IRR
   18 CONTINUE
      IRB=IR
      TLILY=IMS1(IR,M1)
      TLILI=IMS1(IR,M)
      IRM=IR
      KJU=IKU(J)
      KJL=IKL(J)
      IF(KJL.EQ.1) GO TO 3
      IF(KJU-KJL) 2,2,4
    2 KJU=KJL
      GO TO 4
    3 KJL=2
      IF(KJU.GE.2) GO TO 4
      GO TO 76
    4 DO 13 K=KJL,KJU
      DO 14 L=1,IRM
C
C     FEASIBILITY TEST
C
      DO 99 I=1,M
      FISIB=IMS1(L,I)+C(I,J,K)
```

```
      IF(FISIB.GT.B(I)) GO TO 14
   99 CONTINUE
      IF(J.EQ.N) GO TO 85
      RT1=IMS1(L,M1)+R(J,K)
      JK=J+1
      UB1=RT1
      DO 15 JJ=JK,N
      UB1=UB1+R(JJ,KJ)
   15 CONTINUE
      TMSR=0.
      DO 16 I=1,M
      TMSR=TMSR+IMS1(L,I)+C(I,J,K)
   16 CONTINUE
      IMSR=TMSR/M
      LL=IB-IMSR
      UB2=F(N,LL)+RT1
C
C     THE LOWER BOUND TEST
C
      IF(UB1.LT.LB) GO TO 14
      IF(UB2.LT.LB) GO TO 14
C
C     TO UPDATE THE UPPER BOUND
C
      IF(UB2.GT.UB) GO TO 85
      UB=UB2
   85 IRB=IRB+1
C
C     TO CONSTRUCT THE IMBEDDED STATE VECTOR
C
      DO 17 I=1,M
      IMS(IRB,I)=IMS1(L,I)+C(I,J,K)
   17 CONTINUE
      IMS(IRB,M1)=IMS1(L,M1)+R(J,K)
      TRC(IRB,1)=K
      TRC(IRB,2)=L
   14 CONTINUE
   13 CONTINUE
      IR=IRB
      IF(J.EQ.N) GO TO 50
      IF(IR.LT.TLV) GO TO 76
   50 DO 19 IRR=1,IR
      NI=IR-IRR+1
      BOZ=IMS(1,M1)
      JAK=1
      IF(NI.EQ.1) GO TO 19
      DO 21 IRC=2,NI
      IF(IMS(IRC,M1).LT.BOZ) GO TO 21
      BOZ=IMS(IRC,M1)
      TRC1=TRC(IRC,1)
      TRC2=TRC(IRC,2)
      DO 40 I=1,M1
      BAR(I)=IMS(IRC,I)
   40 CONTINUE
      JAK=IRC
```

```
   21 CONTINUE
      DO 22 I=1,M1
      IMS(JAK,I)=IMS(NI,I)
      IMS(NI,I)=BAR(I)
   22 CONTINUE
      DO 23 I=1,2
      TRC(JAK,I)=TRC(NI,I)
   23 CONTINUE
      TRC(NI,1)=TRC1
      TRC(NI,2)=TRC2
   19 CONTINUE
      IF(J.EQ.N) GO TO 76
C
C     DOMINANCY TEST
C
      DO 24 IRR=2,IR
      IF(IMS(IRR,M1).LT.-1.) GO TO 24
      IR1=IRR-1
   39 IF(IMS(IR1,M1).GT.0) GO TO 45
      IF(IR1.EQ.1) GO TO 24
      IR1=IR1-1
      GO TO 39
   45 IF(IMS(IR1,M1)-IMS(IRR,M1)) 25,26,24
   25 DO 28 I=1,M
      IF(IMS(IRR,I)-IMS(IR1,I)) 28,28,24
   28 CONTINUE
      IMS(IR1,M1)=-999.
      IF(IR1.EQ.1) GO TO 24
      IR1=IR1-1
      GO TO 39
   26 ICH1=0
      ICH2=0
      DO 31 I=1,M
      IF(IMS(IRR,I)-IMS(IR1,I)) 32,33,34
   32 ICH1=1
      GO TO 31
   34 ICH2=1
      GO TO 31
   33 ICH3=ICH1*ICH2
      IF(ICH3.EQ.0) GO TO 31
      GO TO 24
   31 CONTINUE
      ICH3=ICH1*ICH2
      IF(ICH3.EQ.1) GO TO 24
      IF(ICH1.EQ.0) GO TO 35
      IMS(IR1,M1)=-999.
      IF(IR1.EQ.1) GO TO 24
      IR1=IR1-1
      GO TO 39
   35 IMS(IRR,M1)=-999.
   24 CONTINUE
      IF(J.EQ.N) GO TO 76
      JAHL=J+1
      IRM=IR
      IRM1=IR
```

```
      DO 88 L=1,IRM
      IF(IMS(L,M1).LT.0.) GO TO 88
      TOKH=0.
      DO 77 JJ=JAHL,N
      TOKH=IMS(L,M1)+R(JJ,KJ)+TOKH
   77 CONTINUE
      IF(TOKH.GE.LB) GO TO 76
      IMS(L,M1)=-999.
   88 CONTINUE
  76  III=0
      IF(IR.GT.0) GO TO 200
      GO TO 300
  200 IRIM=1
      IF(IR.LT.TLV) GO TO 201
      IRIM=IR-TLV+1
  201 DO 36 IRR=IRIM,IR
      IF(IMS(IRR,M1).LT.0) GO TO 36
      III=III+1
      DO 37 I=1,M1
      IMS1(III,I)=IMS(IRR,I)
      IMS(III,I)=IMS1(III,I)
   37 CONTINUE
      DO 38 I=1,2
      TRACE(III,J,I)=TRC(IRR,I)
   38 CONTINUE
   36 CONTINUE
   6  IR=III
      IF(III.EQ.0) GO TO 1002
C     PRINT,ND,J,IR,IMS1(III,M1)
      LIIB=IMS1(III,M1)
C
C     TO UPDATE THE LOWER BOUND
C
      IF(LIIB.LE.LB) GO TO 12
      LB=LIIB
C
C     TERMINATION TEST
C
      IF(LB.GE.UB) GO TO 1000
   12 CONTINUE
      GO TO 1001
 1002 IR=IRM
      IMS1(IR,M1)=TLILY
      LILA=M1-1
      IMS1(IR,LILA)=TLILI
  300 JON=J-1
      GO TO 1001
 1000 IF(J.EQ.N) GO TO 1001
      JON=J
C1001 PRINT,UB
 1001 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  CATAIN
      COMMON /A1/ NODS, N, NC, NS, CC, CAPT
      COMMON /F1/ STRT(6,5), DSTR(6,5), RSRC(10,5)
      COMMON /F2/ CR(6), RMAX(6), RS(10), RSR(6,10)
      COMMON /F3/ DIST(6,6), P(6,20,6)
      COMMON /G1/ T, NPAN, NTYP
      INTEGER     T
      M1          = NC - 1
  500 FORMAT ( 5X, 5A4, F15.3 )
  510 FORMAT ( 5X, I5, 10F7.3 )
  520 FORMAT ( 5X, I2, I3, 10F7.3 )
      ND          = 6
      NTYP        = 2
      T           = 10
      NPAN        = 20
      DO 1200 I = 1, NS
      READ  (5,500) ( STRT(I,J), J = 1, 5 ), CR(I)
      WRITE (6,500) ( STRT(I,J), J = 1, 5 ), CR(I)
 1200 CONTINUE
      DO 1300 I = 1, ND
      READ  (5,500) ( DSTR(I,J), J = 1, 5 ), RMAX(I)
      WRITE (6,500) ( DSTR(I,J), J = 1, 5 ), RMAX(I)
 1300 CONTINUE
      DO 1400 I = 1, M1
      READ  (5,500) ( RSRC(I,J), J = 1, 5 ), RS(I)
      WRITE (6,500) ( RSRC(I,J), J = 1, 5 ), RS(I)
 1400 CONTINUE
      DO 1500 I = 1, NS
      READ  (5,510) L, ( RSR(I,J), J = 1, M1 )
      WRITE (6,510) L
 1500 CONTINUE
      DO 1600 I = 1, NS
      READ  (5,510) L, ( DIST(I,J), J = 1, ND )
      WRITE (6,510) L
 1600 CONTINUE
      DO 1800 I = 1, NS
      DO 1800 J = 1, NPAN
      READ  (5,520) L, M, ( P(L,M,K), K = 1, ND )
      WRITE (6,520) L, M
 1800 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  OBJFCN
      COMMON /A1/ NODS, N, M, NS, CC, CAPT
      COMMON /A2/ ND, INFIS, UNIT, AICH, TLV, ICHEK, IB, IR
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /C4/ C(8,10,6), TRACE(500,10,2), B(8)
      COMMON /E1/ HTYP(10), L1(10), L2(10)
      COMMON /F2/ CR(6), RMAX(6), RS(10), RSR(6,10)
```

```
      COMMON /G1/ T, NPAN, NTYP
      DIMENSION    XL(10)
      INTEGER      T, UNIT
      REAL         L1, L2
      M1          = M - 1
      XXL         = 0.0
      DO 2000 I = 1, N
      XL(I)       = L1(I) * L2(I)
      XXL         = XXL + XL(I)
 2000 CONTINUE
      DO 2100 I = 1, M1
      RS(I)       = RS(I) * XXL
      B(I)        = 70.000
 2100 CONTINUE
      B(M)        = CC / UNIT
      DO 4000 I = 1, N
      INDX        = 0
      DO 3700 J = 1, NS
      CALL BENFIT ( I, J, SUM )
      R(I,J)      = SUM * XL(I) / 100.0
      XSGR        = 0.0
      DO 3100 K = 1, M1
      C(K,I,J)    = 100.0 * XL(I) * RSR(J,K) / RS(K)
 3100 CONTINUE
      C(M,I,J)    = ( XL(I) * CR(J) ) / UNIT
 3700 CONTINUE
 4000 CONTINUE
      RETURN
      END




      SUBROUTINE  EENFIT ( I, J, SUM )
      COMMON /A1/ NCDS, N, M, NS, CC, CAPT
      COMMON /C3/ R(10,6), ICS(10,8), F(10,300), ID(10,300)
      COMMON /E3/ PAR4(10), PAR5(10), TRAF(10), ENVR(10)
      COMMON /E4/ RC(10,6)
      COMMON /F2/ CR(6), RMAX(6), RS(10), RSR(6,10)
      COMMON /F3/ DIST(6,6), P(6,20,6)
      COMMON /G1/ T, NPAN, NTYP
      INTEGER      T
      REAL         L1, L2
      SUM         = 0.0
      DO 3000 K = 1, NS
      L           = 0
      IT          = T
      DTJK        = AMIN1(DIST(J,K),RMAX(K)-RC(I,K)+0.1)
      RR          = RC(I,K) + DTJK
 2500 CONTINUE
      L           = L + 1
      PP          = 1.0 - TRAF(I) * ENVR(I) * ( 1.0 - P(J,L,K) )
      PP          = AMAX1(0.0,PP*RMAX(K))
      IF ( RR .LT. PP ) GO TO 2600
      IF ( PP .LT. RC(I,K) ) GO TO 3000
```

```
      SUM          = SUM + PP - RC(I,K)
      IF ( SUM .LE. 0.0 ) SUM = 0.0
      GO TO 2700
 2600 IT           = IT + 1
 2700 IF ( IT .GE. NPAN ) GO TO 3000
      IF ( L .LT. IT ) GO TO 2500
 3000 CONTINUE
      RETURN
      END
```