A GOAL PROGRAMMING MODEL OF THE STOCHASTIC VEHICLE ROUTING PROBLEM

By

YAHIA ZARE-MEHJERDI

Bachelor of Science in Mathematics
Iranian National University
Tehran, Iran
1976

Bachelor of Science in Civil Engineering
Texas A & I University
Kingsville, Texas
1980

Master of Science in Operations Research
St. Mary's University
San Antonio, Texas
1983

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
December 1986
Thesis
1986D
Z36g
cap.2
A GOAL PROGRAMMING MODEL OF THE STOCHASTIC

VEHICLE ROUTING PROBLEM

Thesis Approved:

[Signatures]

Marian Palmer Lavel
Thesis Adviser

Allen C. Schumann

Roger Holbert

James E. Shanklin

Norman V. Durham
Dean of the Graduate College
This research incorporates the concept of chance-constrained pro-
gramming and multiple objective goal programming in the area of vehicle
routing problems. The research led to the development of the model of
the Goal Programming (GP) Stochastic Vehicle Routing Problem (SVRP) that
allows the decision makers involvement in the solution process of prob-
lem to obtain satisfactory vehicle routes for the SVRP. It is shown
that, mathematically, a new set of deterministic linear time constraints
are equivalent to the nonlinear set of time constraints of the problem
for distributions such as poisson and chi-square. Additionally, the
effects of the route failing probabilities on the total elapsed time of
the whole delivery system, and the existence of the optimum solution for
the "F" type problem are proven mathematically.

A modification of the Clarke and Wright algorithm is developed to
determine the most favorable vehicle routes of the SVRP for the "E" type
problem. Additionally, two heuristic algorithms which are the modifica-
tion of the Clarke and Wright "savings" approach are developed for solv-
ing the "F" type problem. Computational experiments are performed on
three test problems to justify the proposed algorithms. Two interactive
computer programs are developed for the SVRP and goal programming tech-
nique which allows the decision maker to provide satisfactory vehicle
routes.

I wish to express my sincere appreciation and respect to my com-
mittee chairman, Dr. M. Palmer Terrell, for his guidance and support
throughout this research and my doctoral program. I also appreciate Dr. Keolling, who first brought this problem to my attention.

Appreciation is also extended to my committee members, Drs. Allen C. Schuermann, James Shamblin, Wayne C. Turner, and J. Leroy Folks, for their interest and assistance during this study.

I also want to thank Shirley Motsinger and Kenna Long for their virtually faultless typing and valuable suggestions concerning this dissertation.

I wish to convey my special appreciation to my parents, particularly my mother, and to my brothers, sisters, and other relatives for their encouragement and support during my study in the United States.

Finally, to my wife, Mary, and my son, Omid, I wish to express my heartfelt thanks for the many sacrifices they made to allow me to pursue my goals. Without their understanding, support, and love, the days and nights spent in school preparing this study would have been impossible.
TABLE OF CONTENTS

Chapter | Page
--- | ---
I. INTRODUCTION | 1
  1.1 Background | 1
  1.2 Research Objectives | 4
    1.2.1 Primary Objectives | 4
    1.2.2 Secondary Objectives | 5
  1.3 Outline of Succeeding Chapters | 6
II. LITERATURE REVIEW | 7
  2.1 Vehicle Routing Problem | 7
  2.2 Solution Techniques for the VRP | 11
    2.2.1 Background | 11
    2.2.2 Heuristic Algorithms | 11
  2.3 Exact Solutions to the VRP | 25
  2.4 Stochastic Vehicle Routing Problem (SVRP) | 26
    2.4.1 An Overview | 26
  2.5 Interactive Heuristic Approach | 30
  2.6 Summary | 32
III. CHANCE-CONSTRAINED PROGRAMMING (CCP) | 34
  3.1 Introduction | 34
  3.2 Development of Deterministic Equivalents | 36
    3.2.1 Constraint Requirements Random Variable, \( b_f \) | 37
    3.2.2 Input-Output Coefficients Random Variable, \( a_{ij} \) | 38
  3.3 Summary | 39
IV. MULTIPLE OBJECTIVE GOAL PROGRAMMING MODELS | 41
  4.1 Introduction | 41
  4.2 General Model | 42
    4.2.1 Preemptive Goal Programming | 43
    4.2.2 Archimedean Goal Programming | 44
    4.2.3 Multigoal Programming | 44
  4.3 Solution Methods for Linear Goal Programming Problems | 44
  4.4 Summary | 46
Chapter V. DEVELOPMENT OF THE STOCHASTIC VEHICLE ROUTING PROBLEM ................................................. 48

5.1 Development ............................................................................. 48
  5.1.1 The Route Construction Stage ........................................ 49
  5.1.2 The Route Improvement Stage ......................................... 50
5.2 Notations .................................................................................. 51
5.3 Assumptions ............................................................................ 53
5.4 Route Improvement Stage: Problem Formulation ............... 54
  5.4.1 Time Constraints ............................................................. 54
  5.4.2 Demand Constraints ......................................................... 56
5.5 Development of the Deterministic Forms for the
  Set of Constraints of Problem B ............................................. 58
  5.5.1 Deterministic Form for Unload Time
        Constraint ........................................................................ 58
  5.5.2 Deterministic Form for Travel Time
        Constraint ......................................................................... 60
  5.5.3 Evaluation of the Total Time TT ..................................... 62
  5.5.4 Deterministic Form for the Demand
        Constraint ......................................................................... 62
  5.5.5 Safety Stock and Surplus in Terms of
        the Decision Variables Xij ............................................. 63
5.6 Route Construction Stage: Problem Formulation ............. 66
  5.6.1 Using Cost as a Criterion ................................................ 67
  5.6.2 Using Total Time as a Criterion .................................... 67
  5.6.3 Equivalent Deterministic Forms of Problems C and D of
        the RCS of the Problem .................................................... 68
5.7 Distributions Other Than Normal ....................................... 76
5.8 Goal Programming Formulation ........................................ 80
5.9 Sensitivity of Time and Truck Capacity upon the
  Probability of Route Failures .............................................. 83
5.10 Summary ................................................................................. 90

Chapter VI. LINEAR INTEGER GOAL PROGRAMMING TECHNIQUE ......................................................... 92

6.1 Introduction ................................................................. 92
6.2 Cutting Plane Method for Integer Goal Programming .. 93
  6.2.1 Development of All-Integer Cut .................................... 94
  6.2.2 Development of the Mixed-integer Cut ......................... 96
6.3 Algorithm for LIGP using Cutting Plane Method .......... 97
6.4 Branch and Bound Method for Integer Goal
    Programming .......................................................................... 99
6.5 Summary ................................................................................. 108

Chapter VII. DEVELOPMENT OF THE HEURISTIC ALGORITHM FOR STOCHASTIC
            VRP ................................................................. 110

7.1 Introduction ............................................................................ 110
7.2 Clarke and Wright Algorithm .............................................. 110
7.3 Development of the Heuristic Approach for "E"
        Type Problem ..................................................................... 111
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.4</td>
<td>113</td>
</tr>
<tr>
<td>Development of Heuristic Approach for &quot;F&quot; Type Problem</td>
<td>113</td>
</tr>
<tr>
<td>7.4.1 Algorithm (I)</td>
<td>115</td>
</tr>
<tr>
<td>7.4.2 Algorithm (II)</td>
<td>116</td>
</tr>
<tr>
<td>7.5</td>
<td>119</td>
</tr>
<tr>
<td>Example Problem</td>
<td>119</td>
</tr>
<tr>
<td>7.5.1 Route Construction Stage</td>
<td>126</td>
</tr>
<tr>
<td>7.5.2 Route Improvement Stage</td>
<td>129</td>
</tr>
<tr>
<td>7.6</td>
<td>136</td>
</tr>
<tr>
<td>Sensitivity of Elapsed Time Upon the Probability of Route Failures</td>
<td>136</td>
</tr>
<tr>
<td>7.7</td>
<td>145</td>
</tr>
<tr>
<td>Summary</td>
<td>145</td>
</tr>
<tr>
<td>VIII. ANALYSIS OF RESULTS</td>
<td>147</td>
</tr>
<tr>
<td>8.1</td>
<td>147</td>
</tr>
<tr>
<td>Introduction</td>
<td>147</td>
</tr>
<tr>
<td>8.2</td>
<td>148</td>
</tr>
<tr>
<td>Validity of the New Model</td>
<td>148</td>
</tr>
<tr>
<td>8.3</td>
<td>149</td>
</tr>
<tr>
<td>Comparison with the Stewart Model</td>
<td>149</td>
</tr>
<tr>
<td>8.4</td>
<td>150</td>
</tr>
<tr>
<td>Validity of the Developed Heuristic Approaches</td>
<td>150</td>
</tr>
<tr>
<td>8.4.1 Results of Category 1</td>
<td>154</td>
</tr>
<tr>
<td>8.4.2 Results of Category 2</td>
<td>155</td>
</tr>
<tr>
<td>8.4.3 Results of Category 3</td>
<td>155</td>
</tr>
<tr>
<td>8.4.4 Results of Category 4</td>
<td>155</td>
</tr>
<tr>
<td>8.4.5 Results of Category 5</td>
<td>158</td>
</tr>
<tr>
<td>8.5</td>
<td>158</td>
</tr>
<tr>
<td>Summary</td>
<td>158</td>
</tr>
<tr>
<td>IX. USING INTERACTIVE COMPUTER PROGRAMS</td>
<td>161</td>
</tr>
<tr>
<td>9.1</td>
<td>161</td>
</tr>
<tr>
<td>Introduction</td>
<td>161</td>
</tr>
<tr>
<td>9.2</td>
<td>162</td>
</tr>
<tr>
<td>Interactive Linear Integer Goal Programming</td>
<td>162</td>
</tr>
<tr>
<td>9.2.1 The Data Input Procedure</td>
<td>168</td>
</tr>
<tr>
<td>9.3</td>
<td>169</td>
</tr>
<tr>
<td>Interactive Stochastic Vehicle Routing Problem</td>
<td>169</td>
</tr>
<tr>
<td>9.3.1 The Data Input Procedure</td>
<td>170</td>
</tr>
<tr>
<td>9.4</td>
<td>176</td>
</tr>
<tr>
<td>Summary</td>
<td>176</td>
</tr>
<tr>
<td>X. CONCLUSIONS AND RECOMMENDATIONS</td>
<td>177</td>
</tr>
<tr>
<td>10.1</td>
<td>177</td>
</tr>
<tr>
<td>Conclusions</td>
<td>177</td>
</tr>
<tr>
<td>10.2</td>
<td>179</td>
</tr>
<tr>
<td>Recommendations</td>
<td>179</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>180</td>
</tr>
<tr>
<td>APPENDIX A - INTERACTIVE COMPUTER PROGRAM FOR LINEAR INTEGER GOAL PROGRAMMING</td>
<td>185</td>
</tr>
<tr>
<td>APPENDIX B - INTERACTIVE COMPUTER PROGRAM FOR STOCHASTIC VEHICLE ROUTING PROBLEM</td>
<td>264</td>
</tr>
<tr>
<td>APPENDIX C - DATA FOR TEST PROBLEMS 1, 2, AND 3</td>
<td>334</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>I. A 10-Variable Test Problem for Example 3.</td>
<td>107</td>
</tr>
<tr>
<td>II. Summary of Distances Between Location in Miles.</td>
<td>122</td>
</tr>
<tr>
<td>III. Mean Travel Time Between Locations in Minutes</td>
<td>123</td>
</tr>
<tr>
<td>IV. Summary of Demand and Unload Time for Each Demand Point</td>
<td>124</td>
</tr>
<tr>
<td>V. Summary of Results for &quot;E&quot; Type Problem Using ( \alpha_k = 0.1, \beta_k = \eta_k = 0.05 )</td>
<td>127</td>
</tr>
<tr>
<td>VI. Summary of Results for the &quot;F&quot; Type Problem Using Algorithm (I) Where ( \alpha_k = 0.1, \beta_k = \eta_k = 0.05 ) (( \gamma = 0.9 ))</td>
<td>128</td>
</tr>
<tr>
<td>VII. Summary of Results for the &quot;F&quot; Type Problem Using Algorithm (II) Where ( \alpha_k = 0.10, \beta_k = \eta_k = 0.05 ) (( \delta = 0.5 ))</td>
<td>128</td>
</tr>
<tr>
<td>VIII. Summary of Results for Route 1 of &quot;E&quot; Type Problem After Employment of the RIS of the Problem</td>
<td>133</td>
</tr>
<tr>
<td>IX. The Summary of Final Results of Example Problem Based on the &quot;E&quot; Type Problem Analysis</td>
<td>137</td>
</tr>
<tr>
<td>X. Summary of Results for the &quot;E&quot; Type Problem For ( \beta_k = \eta_k = 0.05 )</td>
<td>139</td>
</tr>
<tr>
<td>XI. Summary of Results for the &quot;E&quot; Type Problem For ( \eta_k = 0.05 )</td>
<td>140</td>
</tr>
<tr>
<td>XII. Summary of Results for the &quot;F&quot; Type Problem Using Algorithm (I) Where ( \beta_k = \eta_k = 0.05 ) and ( \gamma = 0.90 )</td>
<td>141</td>
</tr>
<tr>
<td>XIII. Summary of Results for the &quot;F&quot; Type Problem Using Algorithm (II) Where ( \eta_k = 0.05 ) and ( \gamma = 0.90 )</td>
<td>142</td>
</tr>
<tr>
<td>XIV. Summary of Results for the &quot;F&quot; Type Problem Using Algorithm (II) Where ( \beta_k = \eta_k = 0.05 ) and ( \delta = 0.50 )</td>
<td>143</td>
</tr>
<tr>
<td>XV. Summary of Results for the &quot;F&quot; Type Problem Using Algorithm (II) Where ( \eta_k = 0.05 ) and ( \delta = 0.5 )</td>
<td>144</td>
</tr>
</tbody>
</table>
Table | Page
--- | ---
XVI. Comparison of Results of the 50 Demand Points with the Stewart Algorithm for the Chance-Constrained VRP with Normally Distributed Customer Demand | 151
XVII. Comparison of Results for 75 Demand Points with the Stewart Algorithm for Chance-Constrained VRP with Normally Distributed Customer Demand | 151
XVIII. Summary of Data for Test Problem Number 3 | 153
XIX. Summary of Results of "E" Type Problem (Category 1) | 156
XX. Summary of Results for the "F" Type Problem Using Algorithm (I) Where $\gamma = 0.90$ (Category 2) | 156
XXI. Summary of Results for the "F" Type Problem Using Algorithm (II) Where $\delta = 0.50$ (Category 3) | 157
XXII. Solution of Subproblem 9 Using Algorithm (I) with Various Values of $\gamma$ (Category 4) | 157
XXIII. Solution of Problem 9 Using Algorithm (I) with Various Values of $\delta$ (Category 5) | 159
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Flowchart for the Branch and Bound Procedure for LIGP</td>
<td>104</td>
</tr>
<tr>
<td>2. A Tree Diagram Presentation of the Solution of Example 2</td>
<td>105</td>
</tr>
<tr>
<td>3. Initial Set-up</td>
<td>110</td>
</tr>
<tr>
<td>4. NODES i and j linked by using the Savings Approach Concept</td>
<td>111</td>
</tr>
<tr>
<td>5. Algorithmic Flowchart of Procedural Steps for &quot;E&quot; Type Problem</td>
<td>114</td>
</tr>
<tr>
<td>6. Algorithmic Flowchart of Procedural Steps for &quot;F&quot; Type Problem</td>
<td>118</td>
</tr>
<tr>
<td>7. General Structure of the Analysis and Solution of SVRP</td>
<td>120</td>
</tr>
<tr>
<td>8. Algorithmic Flowchart of the Computer Program for LIGP</td>
<td>164</td>
</tr>
<tr>
<td>9. Algorithmic Flowchart of the Computer Program for SVRP</td>
<td>171</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

1.1 Background

Managers of private firms involved in the distribution of goods from a warehouse or depot to designated delivery points, as well as authorities responsible for public or private transportation systems, have become increasingly aware of the need to maximize operational efficiency and minimize delivery costs such as fuel, replacement of vehicles, and labor. To illustrate the economical significance of the Vehicle Routing Problem (VRP), Bodin [7] used a survey by Kearney (1980) to show that about 16 percent of the sales value of an item is based on the physical distribution costs of that item, and of this, about one-fourth is due to downstream distribution of the final product from distribution centers to customers. Turner and Vu further reported that in 1974 about 10 percent of the average community's budget was spent on refuse collection and disposal, a total of 7.8 billion dollars [60]. Factors such as these have attracted a great deal of attention to the VRP, and very recently, to the Stochastic Vehicle Routing Problem (SVRP).

The result of this unprecedented interest has been the development and utilization of computerized procedures to solve certain types of vehicle routing problems which reduce associated distribution costs and delivery time and increase customer satisfaction.
Briefly then, an essential element in any logistics routing system is the allocation and routing of vehicles for the purpose of collecting and delivering goods and services on a regular basis. However, routing decisions are complicated by the need of managers to reduce associated costs, and at the same time, satisfy customer demands by making certain that goods are delivered safe, at the right time, and in the right quantity.

The Vehicle Routing Problem (VRP) is a generic name given to a whole class of problems involving the visiting of customers by vehicles [6, 13]. The VRP is also known in the literature as "vehicle scheduling" [8, 11, 17, 24, 35, 41, 49, 64, 65, 67], "vehicle dispatching" [19, 26, 27, 50], or "delivery problem" [57, 58, 59]. Dantzig and Ramser are generally credited for the first formulation of the VRP as presented in their 1959 paper "The Truck Dispatching Problem" [19]. The VRP can be stated as follows: given a set of nodes and arcs to be visited by a fleet of vehicles, construct a low-cost, feasible set of routes for each vehicle [7].

The SVRP is to design a set of feasible routes starting from and eventually returning to a central depot, in order to deliver commodities to a finite number of demand points with randomly distributed customer demands, randomly distributed travel and unload times having known distribution functions, such that the capacity constraint and time constraints of the problem are satisfied. If, however, the amounts of demand at each location, travel time between any two stations, and unload time at each location are known with certainty, and providing that a vehicle capacity restriction exists, this problem is a deterministic VRP.
In the multiple objective SVRP, more than one criteria are considered in the same problem which, depending on the nature of the criteria, are either maximized or minimized. For example, if the safety of the products on the vehicle route is considered to be one of the criteria, it is to be maximized. If, on the other hand, the total cost or total elapsed time is considered to be one of the criteria, it is to be minimized.

Relevant objective functions for the SVRP may contain the following:

1. Minimize the total cost
2. Minimize the total elapsed time on the route (travel time and unload time)
3. Maximize the safety of products on the route
4. Maximize the fulfillment of emergency services [49]
5. Maximize the fulfillment of conditional dependencies of stations such as deadlines and earliest delivery times [49]
6. Minimize the total deterioration of goods on the route
7. Minimize the safety stock for each vehicle route (this is due to the nature of the probabilistic demands)

Frequently, managers are interested in achieving two or more of the above objectives up to satisfactory levels instead of optimizing a single criteria. Goal Programming (GP), which is one of the techniques for multiple objective decision analysis, can be employed to provide a simultaneous solution to this system of competing objectives. Hence, it is desirable to formulate a GP model of the problem within the framework of the SVRP, such that capacity constraints, time constraints, customer
demand, and decision making requirements are satisfied. Due to the complexity of the SVRP, a set of stations to be visited by a fleet of vehicles needs to be partitioned into feasible sets of routes, one for each vehicle enabling the application of the multiple objective GP technique to each of the vehicle routes. The multiple objective SVRP, then, consists of the following two major stages:

Stage I: Route Construction Stage (RCS)
Stage II: Route Improvement Stage (RIS)

The primary task of the RCS is partitioning a set of stations which are scattered around the central depot, into feasible subsets by applying a VRP heuristic approach. Using concepts of the GP technique, the RIS is used to sequence the stations on each vehicle route to meet the customers' and decision makers' requirements.

1.2 Research Objectives

The primary and secondary objectives of this research are described more specifically in the following sections.

1.2.1 Primary Objectives

The primary objectives of the proposed research are as follows:

1. Within the framework of the SVRP, develop a mathematical formulation for a multiple objective GP model. To accomplish this objective, the following subobjectives must be met:

a. Develop a formulation of the SVRP in which travel time, unload time, and customer demands may be represented as random variables having known distribution functions.
b. Transform the general SVRP into an equivalent deterministic VRP for each stage of the problem.

c. Mathematically prove the existence of a set of deterministic linear time constraints which are equivalent to the nonlinear set of time constraints of the problem for distributions such as the Poisson and chi-square.

d. Develop the Linear Goal Programming (LGP) mathematical formulation of the RIS of the problem where the conflicting multiple objectives are treated explicitly.

e. Mathematically prove the effects of the route failing probabilities of $\alpha_k$ and $\beta_k$ of the total elapsed time of the system where $0 \leq \alpha_k \leq 1$ and $0 \leq \beta_k \leq 1$ for all $k$.

2. Determine an appropriate solution technique for the RCS of the problem. In order to accomplish this objective, the following sub-objectives must be met:

   a. Mathematically prove the existence of the optimum solution for the RCS of the problem.

   b. Develop an algorithm that gives the most satisfactory vehicle routes for the RCS of the problem.

1.2.2 Secondary Objectives

The secondary objectives to be achieved are as follows:

1. Develop a computer program of the algorithm for the heuristic approach which is designed to construct feasible vehicle routes in the RCS of the problem.
2. Develop a computer program of the interactive LGP procedure that will allow the decision maker's involvement in the solution process of the RIS of the problem.

The scope of the proposed research is limited to the single depot, multiple vehicle, node routing problem with stochastic demand and travel and unload times and the development of the multiple objective goal programming formulation of the SVRP.

1.3 Outline of Succeeding Chapters

Chapter I defines the problem and states the objectives and subobjectives of this research. Chapter II reviews the existing literature and the solution techniques of the VRP and SVRP. Chapter III discusses Chance-Constrained Programming (CCP) used with random variables in programming models. Chapter IV reviews the literature on linear goal programming techniques. In Chapter V, the SVRP and its equivalent deterministic forms are developed and some necessary theorems are proven. Chapter VI is devoted to the development of linear integer goal programming (LIGP) techniques. Chapter VII demonstrates the development of an appropriate heuristic approach for solving the SVRP. The heuristic approach developed in this study is a modification of the Clark and Wright algorithm. Chapter IX discusses the details of the interactive computer programs for the SVRP and LIGP techniques. Chapter X gives a conclusion and recommendations for future research in the field of SVRP.
CHAPTER II

LITERATURE REVIEW

2.1 Vehicle Routing Problem

The VRP is a challenging logistics management problem with variations that range from school bus routing to the dispatching of delivery trucks for consumer goods. Regardless of the variations, the basic components of the problem are a fleet of vehicles with fixed capacities and a set of demands for transporting passengers or certain objects (consumer goods, etc.) between specified depots and delivery points. The problem is complicated because managers must also take into consideration a variety of constraints such as fixed vehicle capacity and the duration of a route.

Some of the problems classified under the generic name are the Travelling Salesman Problem (TSP) and its variants; Multiple TSP and Time Constrained TSP; Single Depot, Multiple Vehicle Node Routing (SMVR); Multiple Depot, Multiple Vehicle, Node Routing (MMVR); and Single Depot, Multiple Vehicle, Node Routing Problem (SMVR) with stochastic demands. These problems have a pronounced discrete and combinational structure and are problems in the mathematical programming area known as "combinatorial optimization."

The TSP, a combinatorial optimization problem with some real life applications, is the substructure of all VRP's [14] and has been studied extensively in the literature. Dantzig and Ramser [19] describe the TSP...
as follows: "Find the shortest route (tour) for a salesman, starting from a given city, visiting each of a specified group of cities, and returning to the original point of departure" [19, p. 80]. Mathematically, this problem can be formulated as:

\[
\text{Minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} X_{ij} \quad (2.1)
\]

Subject to:

\[
\sum_{i=1}^{N} X_{ij} = 1, \text{ for all } j \in S = \{1,2,\ldots,N\} \quad (2.2)
\]

\[
\sum_{j=1}^{N} X_{ij} = 1, \text{ for all } i \in S \quad (2.3)
\]

\[
X_{ij} = \begin{cases} 0 \\ 1 \end{cases}, \text{ for all } i,j \in S \quad (2.4)
\]

\[
X_{ij} \text{ (form a tour)} \quad (2.5)
\]

where \( C_{ij} \) is the cost of travelling from node \( i \) to node \( j \), \( C_{ii} = \infty \), where \( i = 1,2,\ldots,N \). Constraint (2.5) can thus be written in the form of

\[
Z_i - Z_j + NX_{ij} \leq N - 1, \text{ for } 2 \leq i \neq j \leq N \quad (2.6)
\]

and for some nonnegative real numbers \( Z_i \).

Since 1959, when Dantzig and Ramser [19] first introduced the VRP and proposed a linear programming based heuristic for its solution, the heuristic method has been widely researched [15, 27]. Christofides and Eilon [13] indicated the largest VRP of any complexity solved to date by exact methods and reported in the open literature contains only 31
demand points. Before considering different approaches for solving the VRP, a formulation of the problem as a 0-1 integer program is given. This problem, known as the "pure delivery" problem, can be formulated as follows [31]:

\[
\text{Minimize: } \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{NV} d_{ij} X_{ijk} \tag{2.7}
\]

Subject to:

\[
\sum_{i=1}^{N} \sum_{k=1}^{NV} X_{ijk} = 1 \quad j = 2, 3, \ldots, N \tag{2.8}
\]

\[
\sum_{i=1}^{N} (\sum_{j=1}^{N} X_{ijk} - \sum_{p=1}^{N} X_{pjk}) = 0 \quad k = 1, 2, \ldots, NV \tag{2.9}
\]

\[
\sum_{i=1}^{N} d_{i} (\sum_{j=1}^{N} X_{ijk}) \leq Q_{k} \quad k = 1, 2, \ldots, NV \tag{2.10}
\]

\[
\sum_{j=2}^{N} X_{ijk} \leq 1 \quad k = 1, \ldots, NV \tag{2.11}
\]

\[
Z_{i} - Z_{j} + N \sum_{k=1}^{NV} X_{ijk} \leq N - 1 \quad i \neq j = 1, 2, \ldots, N \tag{2.12}
\]

\[
\sum_{i=1}^{N} t_{ik} \sum_{j=1}^{N} X_{ijk} + \sum_{i=1}^{N} \sum_{j=1}^{NV} t_{ijk} X_{ijk} \leq T_{k} \quad k = 1, 2, \ldots, NV \tag{2.13}
\]

\[
X_{ijk} = \begin{cases} 
0 & \text{for all } i,j,k, \text{ and } i \neq j \tag{2.14}
\end{cases}
\]
where

\[ N = \text{number of nodes} \]
\[ \text{NV} = \text{number of vehicles} \]
\[ Q_k = \text{capacity of truck } k \]
\[ T_k = \text{maximum time allowed for vehicle } k \text{ on a route} \]
\[ d_i = \text{demand at node } i \ (d_1 = 0) \]
\[ t_{ik} = \text{time required for vehicle } k \text{ to deliver or collect at node } i \ (t_{lk} = 0) \]
\[ t_{ijk} = \text{travel time for vehicle } k \text{ from node } i \text{ to node } j \ (t_{lik} = \infty) \]
\[ d_{ij} = \text{distance from node } i \text{ to node } j \]
\[ X_{ijk} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is traversed by vehicle } k \\ 0 & \text{otherwise} \end{cases} \]
\[ Z_i = \text{arbitrary real numbers, } i = 1,2,\ldots,N \]

The objective function (2.7) represents minimization of total distance travelled by \( \text{NV} \) vehicles. Alternatively, costs could be minimized by replacing \( d_{ij} \) with \( C_{ij} \), depending on the vehicle type. Equation (2.8) ensures that each demand node is served by exactly one vehicle; equation (2.9) ensures that if a vehicle enters a demand node it must exit from that node; equation (2.10) is the vehicle capacity constraint and (2.11) guarantees that vehicle availability is not exceeded; equation (2.12) prohibits subtours; and finally, equation (2.13) is the total elapsed route time constraint.
2.2 Solution Techniques for the VRP

2.2.1 Background

Solution techniques for the VRP fall into two categories: those which solve the problem heuristically and those which solve the problem optimally. Basically, heuristic techniques have proved to be an attractive alternative to exact methods because they are easy to understand, readily accepted by managers, easy to program and maintain for computerized planning, and effective in solving a wide range of practical problems which provide solutions that are usually accepted as "reasonable" [35]. The literature review concentrates on single-depot, multiple-vehicle and multiple-depot, and multiple-vehicle situations.

2.2.2 Heuristic Algorithms

The majority of the previous efforts on the VRP have involved heuristic algorithms. Also, the heuristic methods which have been developed for the VRP are largely modifications of TSP heuristics. These algorithms can be categorized into the following four groups:

1. Tour building heuristics,
2. Tour improvement heuristics,
3. Two-phase methods, and
4. Lagrangian relaxation heuristics.

2.2.2.1 Tour Building Heuristics. The Clarke and Wright "savings" approach is the one used most often in tour building heuristics [17, 30, 31]. This approach calculates the saving between nodes i and j, $S_{ij}$, as shown below:
where \( C_{ij} \) is the delivery cost for moving goods from node \( i \) to node \( j \).

More detail of this approach is given in Section 7.2. Gaskell [25] introduced the following alternatives that give results which are at least as good as the one found by Clarke and Wright's procedure. The savings are calculated as shown below:

\[
\lambda_{ij} = S_{ij} [\bar{d} + |d_{0i} - d_{ij}| - d_{ij}] \text{ and }
\]

\[
\pi_{ij} = S_{ij} - d_{ij}
\]

where \( \bar{d} \) is the average of all \( d_{0i} \). The rest of the procedure is the same as Clark and Wright's; however, the concept of modified savings can be given by \( \pi_{ij} = S_{ij} - \theta d_{ij} \) where \( \theta \) is a shape parameter. By varying \( \theta \), the analyst can place greater or less emphasis on the cost of travel between two nodes, depending on their position relative to the depot.

Yellow [67] suggested using a simple geometrical search technique on an ordered list of the polar coordinates of the delivery points. The saving was defined as

\[
S = d_{0i} + d_{0j} - \gamma d_{ij}
\]

where \( \gamma \) is the shape parameter. Special cases are \( \gamma = 1 \) for the Clarke and Wright procedure and \( \gamma = 2 \) for Gaskell's \( \pi \) method. Equation (2.18) may be expressed by polar coordinates relative to the delivery depot

\[
S = r_i + r_j - \gamma (r_i^2 + r_j^2 - 2r_i r_j \cos (\theta_i - \theta_j))^{\frac{1}{2}}
\]

where \( r_i \) and \( \theta_i \) are the polar coordinates of point \( i \). The rest of the procedure is the same as Clarke and Wright's.
Tillman and Cochran [59] modified the Clarke and Wright algorithm. The essential difference of the two methods is that Tillman and Cochran's method allows for the inclusion of restrictions on the system, and in some cases, will yield a better answer.

Holmes and Parker [35] constructed an extension of Clarke and Wright's approach. This new approach is concerned with the classical VRP where a set of vehicles with known capacities service a known set of points with deterministic demands at the lowest possible cost. The mechanics of the so-called "saving" approach are utilized as the foundation of the algorithm. This procedure is capable of handling the symmetric and nonsymmetric interpoint distances (costs) matrix.

Mole and Jameson [48] proposed a technique which is largely dependent on the Clarke and Wright savings criterion and the r-opt method introduced by Lin and Kernighan [42]. In this technique, a general parametric criterion of the following form was developed for including a node C between nodes A and B in the tour:

$$MSAV_c (A, B) = \lambda d_{oc} + \mu d_{AB} - d_{AC} - d_{BC}$$

where $\lambda$ and $\mu$ are the route shape parameters. In the case where node C is introduced between depot o and node K, the above equation can modify as:

$$MSAV_c (K, o) = (\lambda - 1) d_{oc} + \mu d_{ok} - d_{kc}$$

(2.20)

For $\lambda$ in the range of $1 \leq \lambda \leq 2$ and $\mu = \lambda - 1$, a ranking identical to Gaskell's $\pi$ criterion would be generated from the latter equation where $\pi = (\lambda - 1)^{-1} - 1$. This sequential route building algorithm may be thought of in terms of a repeating sequence of the following steps:
1. Determine the most advantageous position to introduce customer C.

2. Identify the next customer to be placed on the emerging route.

3. Possible resequencing of customers on the emerging route is explored using the r-optimal technique.

Buxery [8] proposed a new model for planning the VRP using the "savings" heuristic rule along with the Monte Carlo simulation, subject to a maximum load restriction. The heart of this technique is similar to the one developed by Clarke and Wright; i.e., "Its function is to monitor the feasibility of the chosen new journey, at any particular juncture, for incorporation into the existing route pattern" [8, p. 566]. The main idea for utilizing the Monte Carlo simulation is based on (1) all methods rely a great deal on time consuming "trial and error" evaluation procedures, and (2) good solutions cannot be obtained without explicitly constructing some alternatives. The procedure requires various parameters such as location of depot, location of demand points, demands, the number of point-pairs contained in the selection list, weighting factor M to control the relative probability of generating each point-pair from the selection list, and finally, the run length if it is desired.

Williams [64] proposed a heuristic technique that could be used in attaining a visual solution. This method is based on joining customers farthest from the depot to the closest feasible customers within the immediate proximity. The route construction starts with nodes at extreme points in the area in order to avoid single long journeys and to minimize the total distance as nodes are added to the solution. Linking
together the closest nodes to the peripheral starting point will generally minimize the distance travelled to service those nodes; thus, sorting of the distance matrix is highly reduced because initially only the closest node is required. After the initial link of a route has been found, then, from the distance matrix, the closest two feasible nodes to the farthest node is a link which has two nodes to which nodes can be assigned. A feasible node is a node that, if added to a link, will not cause the link to violate any restrictions.

As previously mentioned, the VRP has been studied widely, but the multidepot VRP has attracted less attention and only a few articles are presented in the literature. Tillman [57], however, is credited for introducing the multiple terminal delivery problem. Specifically, the procedure begins with an initial feasible solution by assigning each vehicle to its closest depot. The algorithm is based on the "saving" criterion that was developed by Clarke and Wright [17]. Generally, this method involves determining savings from joining points on routes and making possible assignments as a function of the maximum savings for joining demand points on routes. The algorithm permits restrictions to be imposed on the system. One such procedure, however, is Tillman and Cain's [58] modification of the Clarke and Wright procedure which determines the initial solution by passing exactly one route from each demand point to the closest depot. When the distance between demand points i and j \( (d_{ij}) \) and the farther distance between demand point i and depot k \( (U_{ik}) \) is known, then the total distance of all routes is defined as

\[
D = \sum_{i=1}^{N} \sum_{k} \min \{U_{ik}\}
\]
where \( N \) is the number of demand points. This method successfully links pairs of nodes in order to decrease the total distance travelled. However, it should be noted that the computation of savings is not as straightforward as in the case of a single depot problem. Hence, the savings \( S_{ijk} \) must be evaluated by

\[
S_{ijk} = \tilde{U}_{ik} + \tilde{U}_{jk} - d_{ij}
\]

(2.21)

where

\[
U_{ik} = \begin{cases} 
2 \min \{U_{it}\} - U_{ik} & \text{if } i \text{ has not yet been given a permanent assignment} \\
U_{ik} & \text{otherwise.} 
\end{cases}
\]

(2.22)

Savings \( S_{ijk} \) are computed for \( i,j = 1,2,\ldots,N \) \( (i \neq j) \) and \( k = 1,2,\ldots,M \) at each step and can be stored in \( M \) matrices, each \( N \) by \( N \).

Golden, Magnanti, and Nguyen [31] have proposed two algorithms for the multiterminal VRP. The first is based on the saving criterion method and the other is based on the Gillett and Johnson's philosophy [27]. The "saving" based algorithm uses Tillman and Cain's approach for computing savings but excludes the idea of a penalty function. The second algorithm is precisely developed for large problems where the multi-depot VRP is viewed as a two-step process: first, nodes have to be allocated to depots and then routes are built which link nodes assigned to the same depot. A large problem is introduced by dividing it into as many subproblems as there are depots and then solving each problem separately [27].

2.2.2.2 Tour Improvement Heuristics Approach. The best known heuristic approach for the TSP is the branch exchange approach introduced by Lin (1965) and later modified by Lin and Kernighan [45]. Lin
and Kernighan define a tour to be r-optimal if no improvement can be made by replacing any r of its links with any other set of r links. An r-optimal tour has a certain probability of being optimal, and Lin suggests that three-optimal tours should normally be used since these give the best trade-off between computing time and probability of the tour is optimal.

Christofides and Eilon [13], who have modified the Lin "r-opt" procedure, developed a new approach that starts with a feasible solution and tests perturbations to obtain r-optimality. This approach for r = 2 examines each pair of arcs to build a new feasible and economical route which is replaced by any two old arcs from the route. The chief advantage, however, is that it is able to handle restrictions such as

1. Customer wants delivery at a certain time,
2. Capacity may vary between vehicles, and
3. Customer wants delivery by a certain vehicle.

Christofides and Eilon [15] and Lin and Kernighan [45] have shown that the number of operations needed for an r-optimal tour is polynomial in n (number of customers on a tour), exponential in r, and bounded below by n^r; thus, only "small" values of r can be used. Additionally, Christofides and Eilon discovered that when all possible links are considered in joining r changes into a tour, approximately \( \binom{n}{r} (r-1)! 2^{r-1} \) combinations need to be checked in order to ensure r-optimality.

Wren and Holliday [65] generated a customer list in order of the angular coordinate along the most sparse direction. In contrast to the Clarke and Wright method, the number of vehicles available at the depot must first be specified, which allows routes to be built up regarding the number of vehicles available. Customers are then introduced into
the algorithm and each customer is assigned to a vehicle. Next, a
"refine" procedure is activated to determine whether improvements can be
obtained by simple categories, resequencing within routes, or realloca-
tion between routes. The coordinate axis is rotated in equal increments
of 90°, the algorithm is repeated each time, and the best of the four
resulting route structures is chosen. This heuristic approach is capa-
ble of handling both single and multiple depot VRPs.

Russell [50] extended the Lin and Kernighan heuristic procedure to
an approach called "MTOUR." It is directly analogous to the
Christofides and Eilon [15] method in which they extend Lin's 3-opt TSP
heuristic procedure to solve the vehicle dispatching problem. MTOUR is
able to handle side conditions such as due date or interval constraints
requiring that a visit be made only during certain time intervals. The
MTOUR algorithm requires a feasible solution of the VRP with M vehicles
as input. This MTOUR solution is expressed as a travelling salesman
tour on an expanded network, then a modified 3-opt procedure or any
other improvement scheme is used to reduce total cost [7]. At each step
of the modified 3-opt procedure, a check for feasibility must be carried
out to have an improved total cost and feasible solution. Run times,
however, grow approximately as $N^{2.3}$, where $N$ is the number of demand
points.

2.2.2.3 Two-Phase Methods. In the two-phase method, customers
are first assigned to vehicles without specifying the sequence in which
customers are visited. In the second phase, routes are obtained for
each vehicle using a TSP heuristic. The procedures introduced by
Gillett and Miller [26] and Christofides and Eilon [15] are two-phase
methods that use a modified Lin-Kernighan heuristic in phase two.
Gillett and Miller [26] also introduced an algorithm called the "sweep" algorithm. This algorithm consists of two parts, a forward sweep and a backward sweep. In this procedure, the problem is broken down into smaller subproblems which can be solved more easily. The locations are ordered according to their polar coordinate angles from a central depot and assigned to a single route as they are swept by, going through an increasing list of the angles until the vehicle capacity or distance constraints are exceeded. Rectangular coordinates for each demand point are required in order to evaluate the polar coordinates. A customer is chosen at random and the ray from the origin through the customer is "swept" either clockwise or counter-clockwise. Customers are assigned to a given vehicle as they are "swept" until the capacity constraint for that vehicle is reached. A new vehicle is then selected and the sweep continues with assignments now being made to the new vehicle. The "refine" phase checks for improvement which could result from resequencing of customers within a route and reassignment of customers between routes. The procedure is repeated twice, once in the direction of increasing angular coordinates and once in the direction of decreasing angular coordinates. In most cases, the two procedures produce different routes and consequently different minimum total distances. The best approximate solution is the one that has the smallest value. According to Turner and Vu [61], the main disadvantages of this procedure are as follows:

1. It applies only to single-depot problems
2. The computer time increases quadratically with the average number of sites per route if the total number of sites remain relatively constant
3. The second phase requires a TSP procedure to solve each route individually.

The main advantages of this procedure are (1) little computer time is required to solve large problems with small numbers of sites per route, and (2) it is quite simple to program.

The Christofides and Eilon ([13] pp. 332-333) two-phase method begins with a minimal insertion cost heuristic for inserting customers into emerging routes. The following scores are calculated for all the unrouted customers:

$$\delta_r = c_{0r} + \lambda X_{r_ih} \quad (\lambda \geq 1) \tag{2.23}$$

where $X_{r_ih}$ indicates an unrouted customer on the route $R_h$. Then a feasible customer $x_{r*}$ is inserted into route $R_h$ where

$$\delta_{r*} = \min (\delta_r), \quad x_r \text{ unrouted and feasible} \tag{2.24}$$

At each step, route $R_h$ is optimized by using the Lin-Kernighan $r$-optimal method. In the second phase, a customer is designated in each of the routes formed in Phase 1. Beginning with the $K$ routes that join the depot to $i_k$, $k = 1,2,\ldots,K$, the remaining customers are inserted using a rule based on the cost of inserting a customer into alternative routes.

Cheshire, Malleson, and Noccache [11] presented a technique which is a dual heuristic because it retains local optimality at each step while gradually approaching feasibility. This procedure, where solutions are built up by retaining feasibility while gradually approaching optimality, is in contrast with other VRP approaches which are primal heuristic. However, the main features of the proposed algorithm are initial schedule building, the construction of a complete but infeasible
schedule, and feasibility enforcement. An initial schedule is generated by one delivery per vehicle route. At each step of construction of the initial schedule, the next delivery which is farthest from the depot and from those deliveries are already included in the initial schedule is chosen. The total number of vehicle routes must be estimated either by the schedule or by the algorithm. The complete schedule is built up by including deliveries one at a time, but before a new delivery is included in the partial schedule, the existing partial schedule is locally optimized until it is impossible to gain an improvement by repositioning any delivery already included.

The cost function for a delivery on a route is made up of a time and a penalty function. The penalty function is a sum of terms, each proportional to the additional degree of violation of any constraint caused by the inclusion of the delivery into the existing partial schedule. The Lagrangian multipliers are initially set to some low values, then, when all deliveries have been included in the schedule, the Lagrangian multipliers associated with each violated constraint are increased in value and the total schedule is adjusted by single delivery repositioning until 1-opt is again achieved. Next, the cost reduction in each vehicle route is checked. This process is repeated until a feasible total schedule is achieved.

The Gillett and Johnson algorithm [27], an extension of the Gillett and Miller [26] "sweep" algorithm, is a two-stage procedure. During the first stage the assignment of locations to depots are determined and during the second stage, several single depot VRP's are solved. For any location i, t'(i) and t''(i) are considered to be the closest and the second closest depot to i. Then, based on the value of r(i), where
r(i) = \frac{d_{i,t'(i)}}{d_{i,t''(i)}}\text{ for all } i \quad (2.25)

locations are ranked in an increasing value of r(i). Based on this ranking, the nodes that are relatively close to a depot are considered first and assignment of nodes then starts from the list of r(i) until a cluster is constructed around every depot. Say that two nodes, j and k, are already assigned to a depot t and then a new node i is inserted between j and k on a route linked to i, an additional distance \( d_{ji} + d_{ij} - d_{jk} \), which represents a part of the total distance (or costs), will be created. The sweep algorithm, however, is utilized to construct and sequence a route in the cluster around the depot independently.

2.2.2.4 Lagrangian Relaxation Heuristic. According to Bodin [7], Stewart and Golden presented a heuristic algorithm that considers the customer demands explicitly. This procedure treats the capacity constraints by moving them into the objective function and then imposing a penalty when demand on a route exceeds capacity. The mathematical formulation of this VRP is

\[
\text{Minimize:} \quad \sum_{k=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} x_{ijk} \quad (2.26)
\]

Subject to:

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} d_{i} x_{ijk} \leq Q, \quad k = 1,2,\ldots,M \quad (2.27)
\]

\[
X = \{x_{ijk}\} \in S^*
\]

\[
x_{ijk} = \begin{cases} 
0 \\
1 
\end{cases} \quad (2.28)
\]
where

\[ C_{ij} \] = cost or distance of moving from \( i \) to \( j \),

\[ d_i \] = demand at point \( i \),

\[ Q \] = vehicle capacity,

\( S^* \) = the set of all M-TSP solutions.

Then the Lagrangian problem associated with this VRP is

\[
\text{Minimize: } \sum_{k=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} x_{ijk} + \sum_{k=1}^{M} \lambda_k \sum_{i=1}^{N} \sum_{j=1}^{N} d_i x_{ijk} \tag{2.30}
\]

Subject to:

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} d_i x_{ijk} \geq \sum_{i=1}^{N} \sum_{j=1}^{N} d_i x_{ijr} \tag{2.31}
\]

for all \( r = k + 1 \) and \( k = 1, 2, \ldots, M-1 \) and

\[ x = (x_{ijk}) \in S^* \tag{2.32} \]

\[ \lambda_k \geq 0 \] (is the penalty route failure).

Constraint (2.31) is redundant, however, the effect is to assign the largest demand route (number 1), the second largest (number 2), and so on. The Lagrangian problem is solved for \( x(\lambda) \) each time that \( \lambda \) is varied. Hence, the procedure is heuristic due to the fact that the VRP is solved approximately and not exactly. Also, the exact procedure might not give an optimal solution to the VRP since there may be a duality gap between the objective value for the "best" \( x(\lambda) \) and the optimal solution to the VRP.

Stewart and Golden [55] also proposed a newer heuristic algorithms for the VRP which makes use of the lagrangian relaxation to transform the VRP into a M-TSP. The new formulation suggested by the authors is
Minimize \[ \sum_{k=1}^{M} \sum_{i,j=1}^{N} c_{ij} x_{ijk} + \sum_{k=1}^{M} \lambda_k \left( \sum_{i,j=1}^{N} \mu_i x_{ijk} - Q \right) \] (2.33)

Subject to:

\[ \sum_{i,j=1}^{N} \mu_i x_{ijk} - \sum_{r=k+1}^{\infty} \sum_{i,j=1}^{N} \mu_i x_{ijr} \geq 0 \]
\[ k = 1, 2, \ldots, M-1 \]

\[ x = \{x_{ijk}\} \in \mathcal{S}^*, \quad \text{for all } i, j, k \] (2.34)

\[ \lambda_k \geq 0, \quad \text{for all } k \] (2.35)

where \( \lambda_k \) can be thought of as a penalty for each demand on route \( k \) in excess of vehicle capacity. The penalties for larger demand routes are considered to be higher than the small demand routes; however, the arc exchange procedure is used to solve this heuristic algorithm. The key to the algorithm is in the selection of values for the Lagrangian multipliers \( (\lambda_k, k = 1, 2, \ldots, M) \). Only \( \lambda_1 \) is set at a positive level (all other \( \lambda \) are zero), and the value of \( \lambda_1 \) is increased at each iteration until the 3-opt procedure produces a feasible solution to the original VRP. However, \( \lambda_1 \) is the multiplier associated with the first and largest demand route. Usually, a better solution is generated when \( \lambda_1 \) is applied to each route that is infeasible. Then the objective function becomes

Minimize: \[ \sum_{k=1}^{M} \sum_{i,j=1}^{N} c_{ij} x_{ijk} + \lambda_1 \sum_{k \in S} \left( \sum_{i,j} \mu_i x_{ijk} - Q \right) \] (2.36)

where

\[ S = \{k \mid \sum_{i,j} \mu_i x_{ijk} > Q\} \] (2.37)
2.3 Exact Solutions to the VRP

The VRP formulation as an integer program is actually the one presented previously (2.7) - (2.14), as originally formulated by Golden et al. [31].

Balinski and Quandt [5] formulated the VRP as an integer program where it is a representative of a cluster-first, route-second approach to the VRP in which demand points are first assigned to the vehicle clusters and then each vehicle is routed over the demand points assigned to it to determine a delivery sequence. The formulation is

Minimize:  \[ \sum_{j=1}^{M} c_j Z_j \]  \hspace{1cm} (2.38)

Subject to:  \[ \sum_{j=1}^{M} a_{ij} Z_j = 1 \quad i = 1,2,\ldots,N \]  \hspace{1cm} (2.39)

\[ \sum_{i=1}^{N} a_{ij} d_i \leq Q \quad j = 1,\ldots,M \]  \hspace{1cm} (2.40)

\[ Z_j = \begin{cases} 0 & j = 1,\ldots,M \\ 1 & \end{cases} \]  \hspace{1cm} (2.41)

where decision variables \( Z_j \) are binary and specify whether or not cluster \( j \) is used; \( a_{ij} = 1 \), if demand point \( i \) is assigned to clusters \( j \) and 0, otherwise, these coefficients are fixed and defined for each cluster \( j \); \( d_j \) is the demand of station \( j \) and \( Q \) is the capacity of any vehicles in the fleet which are assumed to be homogeneous; \( C_j \) is the minimum cost of any vehicle route passing the demand points \( i \) assigned to the \( j \)th cluster (i.e., the demand point \( i \) with \( a_{ij} = 1 \)).
Foster and Ryan [24] proposed an integer programming formulation of the VRP which is solved using the Revised Simplex Method. This method is strictly primal in that both feasibility and integrality are withheld at all stages. An integer programming formulation of the VRP with a planning horizon of more than one day is extended to incorporate the linear constraints. The suggested formulation is

\[
\text{Minimize: } \sum_{j \in J} (V + m_j) X_j \tag{2.42}
\]

\[
\text{Subject to: } \sum_{j \in J} a_{ij} X_j = 1 \quad i = 1, 2, \ldots, N \tag{2.43}
\]

where \(X_j\) is 0 or 1, represents the probability that route \(j\) is in the schedule; \(V\) is the mileage equivalent cost of each vehicle; \(m_j\) is the total mileage of route \(j\); \(a_{ij} = 1\) if delivery \(i\) is made on route \(j\); and \(N\) is the number of deliveries. \(J\) is the set of all feasible routes.

Fisher and Jaikumer [23] have formulated a heuristic approach which describes the VRP as consisting of two interrelated components: the TSP and the Generalized Assignment Problem. Finally, Christofides, Mingozzi, and Toth [12] have formulized the VRP as a dynamic program problem.

2.4 Stochastic Vehicle Routing Problem (SVRP)

2.4.1 An Overview

The SVRP has attracted less attention in the literature than the deterministic VRP. However, the SVRP is a problem of interest to operation researchers due to the wide applicability of such a model in real life situations. The SVRP is to design a set of routes starting from and eventually returning to a central depot and to deliver products to a
fixed number of demand points such that the capacity constraints, probabilistic customer demands, and the duration of the routes are satisfied.

Tillman [57] proposed a modification of the Clarke and Wright procedure for multidepot delivery and collection problems having probabilistic demands that are poisson distributed. The objective function of the delivery problem for a given number of stop points on a proposed route is

\[
\min \ E[\text{cost}] = \min_{R} \left( \int_{0}^{R} C_1(D)h(D)dD + \int_{R}^{\infty} C_2(D)h(D)dD \right) \quad (2.42)
\]

where the first expression from the right indicates the cost of not hauling enough commodity to satisfy all customer demands on a route and the second expression from the right represents the cost of hauling excess commodity on the route that is not needed.

The value of \( R \) determined for each route is the load assigned to the truck for that route. Notations are

\[
C_1(D) = \begin{cases} 
\text{cost of hauling excess commodity on the route that is not needed, or} \\
\text{cost of completing scheduled route and having unfilled capacity}
\end{cases}
\]

\[
C_2(D) = \begin{cases} 
\text{cost of not hauling enough commodity to satisfy all the demands on the route, or} \\
\text{cost of filling truck prior to completing the scheduled route}
\end{cases}
\]

\[
D = d_1 + d_2 + d_3, \ldots, + d_n \quad (2.43)
\]

\( d_i \) = the probabilistic demand for the \( i^{\text{th}} \) stop
\( f_i(d_i) \) = probability density function of the random variable \( d_i \)

\( h(D) \) = probability density function of \( D \)

Golden and Stewart [30] have extended Tillman's SVRP in a different way considering only a single depot problem. In this technique the locations on the route are \( n_1, n_2, n_3, \ldots, n_k \), and it is assumed that all vehicles have the same capacity \( Q \) and that the total demand for all locations is

\[
X = d_{n_1} + d_{n_2} + d_{n_3} + \ldots, + d_{n_k}
\]  

(2.44)

where \( d_{n_i} \) is the demand at location \( i \) which is described by the independent poisson distribution with mean and variance \( \lambda n_i \). Then

\[
E(X) = \text{Var}(X) = \lambda n_1 + \lambda n_2 + \ldots, + \lambda n_k
\]  

(2.45)

for that route. Using the central limit theorem and approximating with normal distributions, then \( \mu = \lambda n_1 + \ldots, + \lambda n_k \), and \( \sigma = \sqrt{\mu} \). However, by considering the definitions of primary and secondary errors from Chapter V, Section 5.5.5, one can write,

\[
P(X \geq Q) = P(\text{primary error}) = P(Z \geq \frac{0 - \mu}{\sqrt{\mu}}) \geq (1 - \alpha) \]  

(2.46)

and

\[
P(X \leq aQ) = P(\text{secondary error}) = P(Z \leq \frac{a0 - \mu}{\sqrt{\mu}}) \geq \alpha
\]  

(2.47)

where \( Q \) is the truck capacity and \( 0 < a, \alpha \leq 1 \).

Assuming that \( \mu \) is nearly the same for most of \( K \) routes, then an artificial capacity \( \overline{\mu} \), as the vehicle capacity, can be used along with
the $\lambda_n i$ as demand points and the "saving" approach of Clarke and Wright to obtain a fixed set of routes. Therefore, the following problem is the one that must be solved:

Minimize: \text{expected total cost} \quad (2.48)

Subject to:

1. a fixed set of routes,
2. satisfaction of customer demands,
3. $P(\text{primary error}) \geq (1 - \alpha)$, and
4. vehicle capacity is obeyed.

where $0 \leq \alpha \leq 1$.

Golden and Yee [29] extended this work to several other demand distributions and presented a more comprehensive view of vehicle routing.

In a later article, Yee and Golden [66] presented a dynamic programming approach to determine the driver operating strategies when demands on a route are probabilistic. Specifically, after delivery of goods to a demand point on a fixed route, which has already been determined by the Clarke and Wright procedure, the driver is faced with the decision of whether to return to the depot to replenish the supply. However, the optimal decision is based on whether the remaining supply of goods in the vehicle is greater or less than some critical value which must take into account the following criteria:

1. The probabilistic demands on the remaining portion of the route, and
2. The distances between the remaining customers.
Cook and Russell [18] have successfully treated a large routing problem with timing constraints and stochastic travel times and demands. The authors approach the problem by generating a deterministic solution using the MTOUR algorithm and then testing these routes via simulation to demonstrate that they are effective; however, the stochastic nature of the problem is not explicitly considered in the route generation stage. The basic procedure for the generation of travel times and pickup times is based on the development of the multiple regression equations for each random variable so that the point estimates can be calculated. The regression equation for the intra-city transit times is derived by employing the euclidean distance and average speed limit as the independent variables. The service time (pickup time) is considered to be a function of two independent variables: number of containers and the total capacity of the containers. Based on these assumptions, the second regression equation for pickup times is determined.

2.5 Interactive Heuristic Approach

Interactive vehicle routing is a general approach in which a high degree of human interaction is incorporated into the problem solving process [7]. It is a method of building routes which is under the control of the decision maker who uses an interactive computer program to indicate the results of decisions made in terms of cost, time, distance, or vehicle utilization.

Krolak et al. [41] proposed a man-machine approach which takes the following steps:

1. The decision maker defines the problem,
2. The computer organizes the data and then gives several alternative solutions using a sophisticated heuristic technique,
3. The decision maker creates another solution and the computer compares the solutions using a pictorial display, and
4. The decision maker attempts to modify the computer solution.

This process continues until the decision maker is satisfied with the solution.

Stacy [53] has developed an interactive vehicle routing algorithm which creates various logical stages in the trail of project design, data collection, validation, and staff training. Also, Waters [63] has developed an interactive vehicle routing algorithm which is able to introduce the concept to new or trainee schedulers. Some of the advantages and disadvantages of the interactive procedure as summarized by Turner and Vu [61] are given below:

Advantages of the Interactive Vehicle Routing problem:

1. Human interaction is allowed, yielding better solutions
2. The computer helps organize the data for the decision maker

Disadvantages of the Interactive Vehicle Routing problem:

1. It is time consuming for both the decision maker and computer
2. The solution is usually suboptimal
3. The concepts require trained or experienced personnel

Park [49] presented a heuristic algorithm to determine vehicle routes for the multiple-vehicle, single-depot case where conflicting multiple objective functions are treated explicitly. This heuristic approach is based on the ideas of Gillett and Miller [26], Clarke and
Wright [17], and Williams [64], which were discussed earlier. Park's heuristic approach implies different upper bounds for the constraints on vehicle travel distance and are based on the preemptive goal priority structure.

Allison [2] developed an interactive model to solve the Workload Balancing Vehicle Routing Problem (WBVRP) using multiple criteria analysis. This research is concerned with the VRP in order to minimize the total distance of the whole delivery system and the deviation in workload among the routes. The workload elements are defined to be (1) total distance on time spent driving, and (2) the total weight or amount of goods delivered. The WBVRP is a multiple criteria optimization problem and is concerned with the deterministic customer demand and travel time.

2.6 Summary

This chapter has presented a literature review of the VRP, multiple-depot VRP, and SVRP. As indicated, the single-depot, multiple-vehicle, node routing problem has attracted the attention of most researchers whereas little research has been conducted on the SVRP and multidepot, multiple-vehicle, node routing problem.

As previously discussed, solution techniques for the VRP are divided into two main categories: those which solve the problem optimally and those which solve the problem heuristically. Optimal seeking procedures are only practical for solving small-sized problems while heuristic techniques are the most promising tools for solving large-scale problems. For this reason, a great deal of attention has been given to the Clark and Wright [17] heuristic approach and its modifications and as well as to the Gillett and Miller [26] approach.
In this research, the heuristic methods were categorized into four groups: tour building heuristics, tour improvement methods, two-phase methods, and Lagrangian relaxation heuristic approaches. It should be noted that there are relatively few interactive approaches that solve the VRP and only two procedures that are capable of handling the VRP in a multiple objective environment [2, 49]. Moreover, each of the procedures described, with the exception of [2, 49], has a single objective cost, time, or distance minimization.
CHAPTER III

CHANCE-CONSTRAINED PROGRAMMING (CCP)

3.1 Introduction

When the parameters in a mathematical programming model are presumed to be random variables rather than constants, a stochastic programming problem must be solved. These problems involve risk if the probability distributions of the random variables are known, or involve uncertainty if the distribution of at least one random variable is unknown. The difficulties of dealing with risk and uncertainty in programming problems have been discussed in the literature since the 1950's.

Chance-Constrained Programming has been introduced into stochastic programming literature mainly through the exposition of Charnes and Cooper [10]. These authors suggest the E, V, and P models. In the E model the expected value of the objective function is to be maximized; in the V model the objective is to minimize a generalized mean square error; and in the P model the purpose is to maximize the probability that $C'X$ does not exceed a given constant $C_0'X_0$. In this technique, a decision vector $X$ has to be selected such that each constraint is satisfied at least $\alpha (0 \leq \alpha \leq 1)$ percent of the time. The topic of CCP is perhaps best introduced by first exhibiting an ordinary LP problem in its general form as:
Minimize $Z = C'X$

Subject to: $AX \leq b$, $X \geq 0$

where $A$ is an $m \times n$ matrix of constraints and $C'$ is a $1 \times n$ matrix while $b$ is an $m \times 1$ matrix. A chance-constrained formulation would replace the above problem with one of the following kind:

Minimize $Z = \sum_{j=1}^{n} C_jX_j$ \hfill (3.1)

Subject to:

$$P(\sum_{j=1}^{n} a_{ij}X_j \leq b_i) \geq \alpha_i \quad \text{for all } i = 1, \ldots, m \hfill (3.2)$$

$$X_j \geq 0 \quad \text{for all } j \hfill (3.3)$$

where "$P$" means probability and $0 \leq \alpha_i \leq 1$. The parameters of this problem are the objective function coefficients $C_j$, the coefficients $a_{ij}$, and the right hand side values $b_i$. Practically, $a_{ij}$, $C_j$, and $b_i$ are not necessarily constant, and in general, some or all of their elements are random variables. The vector $\alpha$ is a set of constants that are probability measures which determines the extent of the constraint violations.

The value of the objective function, $Z$, will depend upon the values of $C_j$, $b_i$, and $a_{ij}$ when they are random variables having known distribution functions. The "E Model" [10] that optimizes the expected value of the objective function may be used only when the $C_j$ are random variables. When one or more $a_{ij}$ and/or one or more $b_i$ are random variables, the "E" Model cannot be applied. In this case, the surrogate
models of stochastic programming such as CCP and stochastic programming with recourse may be applied [33].

Problem (3.1)-(3.3) seek a solution vector \( X \) that satisfies the CC (3.2) and minimizes the value of \( Z \). Many authors [20, 22, 28, 33, 62], as well as Charnes and Cooper [10], have offered methods to convert such stochastic models into their deterministic models (not necessarily linear) which can be solved by the existing mathematical programming techniques.

3.2 Development of Deterministic Equivalents

In this section, two cases where constraint requirements, \( b_i \), and input-output coefficients, \( a_{ij} \) are random variables having known distribution functions will be discussed. To develop the equivalent deterministic form of chance-constrained inequality (3.2), consider in more detail a constraint

\[
\sum_{j=1}^{n} a_{ij} X_j \leq b_i \quad \text{for all } i = 1, \ldots, m \tag{3.4}
\]

in the following two situations:

1. \( b_i \) are independently distributed random variables with mean \( \mu_{b_i} \) and variance \( \sigma_{b_i}^2 \), and
2. \( a_{ij} \) are random variables with mean \( \mu_{a_{ij}} \) and variance \( \sigma_{a_{ij}}^2 \), \( a_{ij} \) are distributed independently of \( a_{ik} \) (\( j \neq k \)).

The random variables \( b_i \) and \( a_{ij} \) are assumed to be normal random variables. However, Sections 3.2.1 and 3.2.2 deal with the development of the equivalent deterministic form of the probabilistic constraints considering the above situations, respectively. A comprehensive survey
for the development of the equivalent deterministic forms for other situations is discussed in [68].

3.2.1 Constraint Requirements Random Variable, $b_i$

In particular, constraint (3.4) is

$$P\left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i\right) \geq \alpha_i$$

for all $i = 1, 2, \ldots, m$ \hspace{1cm} (3.5)

where $(1 - \alpha_i)$ denotes the allowable "risk" that a random variable will be chosen such that

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i.$$

The equivalent deterministic form of the constraint (3.5) for normally distributed random variables of $b_i$ is

$$\sum_{j=1}^{n} a_{ij} x_j \geq \mu_{b_i} + \sigma_{b_i} K_{\alpha_i}.$$

Where $K_{\alpha_i}$ is a standard normal value such that $\Phi(K_{\alpha}) = \alpha$ and $\Phi$ represents the cumulative distribution function for the standard normal [28, pp. 275]. Hence, by solving the problem

Minimize: $\sum_{j=1}^{n} C_j x_j$ \hspace{1cm} (3.6)

Subject to: $\sum_{j=1}^{n} a_{ij} x_j \geq \mu_{b_i} + \sigma_{b_i} K_{\alpha_i}, x_j \geq 0, i = 1, \ldots, m$ \hspace{1cm} (3.7)

one can obtain the optimal solution to problem (3.1)-(3.3).
3.2.2 Input-Output Coefficients

Random Variable, $a_{ij}$

To deal with one constraint at a time, drop the subscript $i$ and let $\mu_j$ and $\sigma_j^2$ be the mean and variance of $a_j$. Then given $X_j$, the mean value of $\sum a_{ij} X_j$ is $M = \sum_j \mu_j X_j$ and its standard deviation is $S = (\sum_j 2^{\frac{1}{2}} \sigma_j^2 X_j^2)^\frac{1}{2}$ (assuming $M$ and $S$ exist). Now if there exists a constant $\tau$ such that

$$P\left(\frac{\sum a_{ij} X_j - M}{S} \leq \tau\right) = \alpha \quad (3.8)$$

then constraint $P(\sum a_{ij} X_j \leq b) \geq \alpha$ is equivalent to the nonstochastic constraint

$$M + \tau S \leq b. \quad (3.9)$$

Of course, $M$ and $S$ contain the unknown $X_j$. Constraint (3.9) is generally nonlinear in nature, as can be seen when it is written in the following form:

$$\sum_{j=1}^{n} \mu_{a_{ij}} X_j + \tau \left(\sum_{j=1}^{n} \sigma_{a_{ij}}^2 \right)^\frac{1}{2} \leq b_i \text{ for all } i, \quad (3.10)$$

where $\tau = -K_{\alpha}$ and $K_{\alpha}$ is as previously defined [28]. The constraint (3.9) can be substituted for (3.2) if $M$ and $S$ exist and $\tau$ is independent of $X_j$. This will be the case if the distribution of $(\sum a_{ij} X_j - M)/S_j$ is the same as $(a_{ij} - \mu_{a_{ij}})/\sigma_{a_{ij}}$. This case is true when $a_{ij}$ are normally distributed or random variables with the same stable distribution with parameters $U_{ij}$ and $V_{ij}$, respectively. Vajda [62, p. 84]
claims that stable distributions have the common property of being completely determined by the specifications of two parameters \(U\) and \(V\) (not necessarily the mean and standard deviation) where \(U\) is real and \(V > 0\). The convolution of any \(n \times m\) distributions \(F((a_{ij} - U_{ij})/V_{ij})\) for \(i = 1,2,\ldots, m\) and \(j = 1,2,\ldots, n\) is of the form \(F((a - U)/V)\).

Poisson, binomial, chi-square, and normal distributions belong to this family.

The deterministic constraint (3.10) can be written as:

\[
\sum_{j=1}^{n} \mu_{aij} x_j - b_i \leq -\tau \left( \sum_{j=1}^{n} \sigma_{ij} x_j^2 \right)^{\frac{1}{2}}
\]

(3.11)

When \(\alpha > 0.5\) then \(\tau\) is negative, which requires that one square both sides of the inequality to obtain

\[
b_i^2 + \left( \sum_{j=1}^{n} \mu_{aij} x_j \right)^2 - 2b_i \sum_{j=1}^{n} \mu_{aij} x_j - \tau \sum_{j=1}^{n} \sigma_{ij} x_j^2 \leq 0
\]

(3.12)

which is a quadratic constraint.

When the random variables \(a_{ij}\) or \(b_i\) are not normally distributed, the development outlined above does not apply. Goicoechea, Hansen, and Duckster [28, pp. 276-281] developed the equivalent deterministic forms of the probabilistic constraints which consist of random variables other than normal, such as exponential, uniform, and beta random variables.

3.3 Summary

The most common method for dealing with random variables in programming models is through certainty equivalents which can be achieved by transforming the CCP problems into nonstochastic problems. The
equivalent deterministic forms of the probabilistic constraints are either linear, as shown in (3.7), or nonlinear, as shown in (3.10) or (3.12).
CHAPTER IV

MULTIPLE OBJECTIVE GOAL PROGRAMMING MODELS

4.1 Introduction

Goal Programming draws upon the highly developed and tested techniques of linear programming, yet provides a solution to a complex system of competing objectives. This technique can handle problems having a single goal with multiple subgoals as well as problems having multiple goals and subgoals [69]. The basic concept of GP involves incorporating some managerial goals into the constraints of the model.

Goal Programming technique was originally introduced by Charnes and Cooper [10] in early 1961 for a linear model. The GP has been extended into many areas, including the capital budgeting problem [38] and aggregate production and manpower planning [1]. Lee [42] applied goal programming to problems in production planning, financial decisions, academic planning, and medical care, to mention a few. A GP model is useful for the following three types of analysis [38]:

1. To determine the input (resource) requirements to achieve a set of goals
2. To determine the degree of attainment of defined goals with given resources and
3. To provide the optimum solution under varying inputs and priority structures.
In general, a goal programming problem can be categorized as:

1. Linear goal programming (LGP) problem,
2. Linear integer goal programming (LIGP) problem, and
3. Nonlinear goal programming (NGP) problem.

The LGP problem is discussed in section 4.2 and LIGP has been delayed until Chapter VI. The importance of NGP has been recognized by many authors including Griffith [32], Ignizio [38], and Lee and Wynne [43].

4.2 General Model

The general model of LGP can be stated as follows [38]:

Minimize \[ Z = \sum_{j=1}^{k} \sum_{i=1}^{m} p_j (\tilde{w}_{ij} n_i + \tilde{w}_{ij} p_i) \] (4.1)

Subject to:

\[ \sum_{r=1}^{n} a_{ir} x_r + n_i - p_i = b_i \quad i = 1, 2, \ldots, m \] (4.2)

\[ f_i(x) \begin{cases} \leq b_i & i = m + 1, \ldots, s \end{cases} \] (4.3)

\[ n_i p_i = 0 \quad i = 1, 2, \ldots, m \] (4.4)

\[ x_r \geq 0, n_i, p_i \geq 0, \quad i = 1, 2, \ldots, m \] (4.5)

\[ r = 1, 2, \ldots, n \]

where

- \( p_j \) is the preemptive priority weight assigned to goal \( j \)
- \( \tilde{w}_{ij}, \tilde{w}_{ij} \) are numerical (differential) weights assigned to the deviational variables of goal \( i \) at a given priority level \( j \)
Pi = represents the positive deviations or surplus variables from goal j (overachievement)

ni = represents the negative deviations or slack variables from goal j (underachievement)

bi = is the ith target level where i = 1, 2, ..., m

ai_r = is the technological coefficient of X_r in goal i.

The sets of goal constraints are those with i = 1, 2, ..., m and the sets of rigid constraints are those with i = m + 1, ..., s.

There are three basic approaches to problems characterized by an a priori set of goals: Preemptive Goal Programming, Archimedean (or Non-preemptive) Goal Programming, and Multigoal Programming. These three approaches are discussed in more detail below.

4.2.1 Preemptive Goal Programming

The objective function for preemptive goal programming is often written as [69]:

\[
\sum_{i=1}^{k} p_i f_i (n_i, p_i) = p_1 f_1 (n_1, p_1) + \ldots + p_k f_k (n_k, p_k). \tag{4.6}
\]

The purpose of preemptive goal programming is the minimization of \( f_i (n_i, p_i) \), one by one, in the order of their (preemptive) priorities. Functions \( f_i \) are typically linear functions of deviational variables; i.e., \( f_i = n_i, f_i = p_i, f_i = n_i + p_i, \) or \( f_i = [W_i n_i + (1 - W_i) p_i] \), and so on. The summation above is redundant and meaningless; however, it is prevalent in the literature and thus cannot be ignored.
4.2.2 Archimedean Goal Programming

The objective function is to minimize [69]:

\[ \sum_{i=1}^{k} W_i \left[ f_i \left( n_i, p_i \right) \right]^r = W_1 \left[ f_1 \left( n_1, p_1 \right) \right]^r + \ldots + W_k \left[ f_k \left( n_k, p_k \right) \right]^r. \]  (4.7)

All the objective functions are considered simultaneously and their weights \( W_i \) are not preemptive. Powers \( r \) can take any value, but usually \( r = 1, 2, \) or \( \infty \).

4.2.3 Multigoal Programming

The purpose of Multigoal Programming is to minimize

\[ [f_1 \left( n_1, p_1 \right), \ldots, f_k \left( n_k, p_k \right)], \] as in Multiobjective Linear Programming [3, 36, 37, 69]. However, it is not necessary to write the objective function in terms of an aggregate preference function. Other variants of GP and multiobjective linear programming can be found in [28, 37, 43].

4.3 Solution Methods for Linear Goal Programming Problems

The most commonly used solution techniques for solving LGP problems are, partitioning goal programming [4], multiphase linear goal programming [28, 31], and interactive sequential goal programming [43].

Arthur and Ravindran [4] have modified the method of solution of GP problems with preemptive weights into a procedure called the Partitioning GP algorithm. In the partitioning procedure constraints should be categorized such that a nested series of GP problems can be formed:

\[ SP_1 \subseteq SP_2 \subseteq \ldots \subseteq SP_k \subseteq \ldots \]
In general, \( SP_k \) stands for the \( k^{th} \) subproblem which consists of those goal constraints assigned to the first \( k \) priority levels and the corresponding terms in the objective function of the \( k^{th} \) subproblem. The solution procedure starts with the smallest subproblem, \( SP_1 \), which consists of all goal constraints assigned to this priority, the system constraint, and the corresponding terms in the objective function. The main idea after obtaining the optimal solution of each subproblem is to examine the optimal tableau for alternate optimal solutions. If no alternative solutions exists, then the solution is optimal for the GP problem. In this case, the value of decision variables of the optimal tableau is substituted into the goal constraints of the lower priority levels (if any exist) to calculate their attainment levels. If alternate optimal solutions exist, the optimization process is continued after augmenting the next set of goal constraints and their objective function terms into the optimal tableau. However, the process of addition of goal constraints and objective function terms continues until no alternate optimum solution exists for one of the subproblems, or until all priority levels have been considered in the optimization process. The linear independency between each pair of individual variables guarantees no need for the dual simplex operation on the updated tableau at the beginning of each optimization process. Most important, when the optimal solution to the \( SP_{k-1} \) is obtained, before the addition of new goal constraints (for \( SP_k \)) into the optimal tableau, one should delete all nonbasic columns which have a negative value for \( (Z_j - C_j) \) from the optimal tableau of \( SP_{k-1} \) for further consideration.
The multiphase (or modified simplex) algorithm is simply a refinement of the well-known two phase method. In this method, the basic simplex method of linear programming is utilized to minimize the deviational variables. The deviational variables are ranked according to preemptive priority factors so that during the solution process the goals are considered in order of their priorities. The weighting method is allowed to incorporate the cardinal values to goals at given priority levels [27, 44].

In most cases, the multiple objective problems cannot be optimized simultaneously because such problems involve making trade-off decisions to get the "best compromise" solution. However, Interactive Sequential GP [46] (ISGP) is a link between GP and interactive approaches which is based on the implicit assumption that the decision maker can adjust the desired goals through an interactive learning process based on the information in a set of solutions. Any iteration, say r, consists of two phases: calculation and evaluation. The Principal Solution and a set of Alternate Solutions are obtained in the calculation phase, and the evaluation phase consists of the decision maker's indication of his preference judgment about these solutions in the form of new desirable goal levels. With this new information, the process goes back to the calculation phase of the \((r+1)\)th iteration. Both linear and nonlinear problems can be solved by ISGP. Additional details concerning this procedure are given by Masud and Hwang [46].

4.4 Summary

The GP approach appears to be an appropriate solution technique in developing a model to attain multiple, competitive, and often conflicting objectives with varying priorities. However, GP is not the answer
to all decision problems. In fact, there are a large number of problems that cannot be solved by this method, nor can this technique replace the subjective aspects of decision making. The application of GP for decision analysis does force the decision maker to think of goals and constraints in terms of their importance to the organization, and thus are an invaluable aid to the decision-making process.
CHAPTER V

DEVELOPMENT OF THE STOCHASTIC
VEHICLE ROUTING PROBLEM

5.1 Development

This chapter is concerned with the development of the VRP within the framework of stochastic programming and addresses the goal programming formulation of the problem in which priorities of various goals are identified. The sensitivity of time and truck capacity upon the probability of route failures is analyzed and some necessary theorems are proven.

The SVRP examined in this research is concerned with the multiple-vehicle, single-depot node routing problem in which restrictions are placed on the total travel and unload times of each vehicle route. Alternatively, restriction can be imposed by the Decision Maker (DM) regarding total elapsed time of each vehicle route instead of specifying each type of time constraint individually. For example, the DM may specify that routes must require less than 10 hours time for both travel and unload times.

The time constraints arise in many real life problems such as industrial refuse collection and scheduled mail pick-up and delivery problem [18]. The importance of time constraints has been recognized by many authors including Cheshire et al. [11], Fisher and Jaikumer [23], Evans et al. [21], and Williams [64].
Some of the work that has been published in the literature deals with stochastic elements within the framework of linear, single objective, and heuristic approaches [18, 29, 30, 57]. As mentioned in Chapter I, this author considers two major stages for the stochastic VRP, the Route Construction Stage (RCS) and Route Improvement Stage (RIS). The following is a brief description of these two stages.

5.1.1 The Route Construction Stage

The RCS of the SVRP consists of problem formulation and partitioning a set of stations into feasible sets of vehicle routes. The presence of nonlinearities in the equivalent deterministic form of the SVRP generally make the problem more complex than similarly-sized VRPs. For this reason, only heuristic methods for solving SVRP are considered in this study.

The RCS of the SVRP consists of the following steps:

1. Problem formulation in which objective functions and probabilistic constraints are identified,
2. Transformation of the above stochastic problem into an equivalent deterministic form, and
3. Partitioning of a set of stations into feasible subsets using an appropriate heuristic approach.

The RIS of the problem consists of problem formulation and sequencing of stations on each vehicle route to meet the customer's and decision maker's requirements.
5.1.2 The Route Improvement Stage

This stage of the problem is important because the final results depend on the decision maker's policy and the way in which goals and their relevant priorities are listed. Obviously, the goal priority structure of all objectives must be carefully stated because the achievement of one goal may result in a very poor achievement of the remaining goals.

The RIS of the SVRP consists of the following steps:

1. Problem formulation in which goals and probabilistic constraints are identified,

2. Transformation of the above stochastic problem into an equivalent deterministic form, and

3. GP formulation of RIS in which priorities of various goals are identified.

Here, it should be noted that the mathematical formulation of the RIS, and in turn the derivation of its equivalent deterministic form, is easier to formulate than the RCS of the problem. Also, it is necessary to develop mathematically the objective functions of the RIS of the problem in terms of the decision variables. For these reasons, the problem formulation of the RIS of the problem is given in Sections 5.4 and 5.5 and the problem formulation of the RCS of the problem is delayed until Section 5.6.
5.2 Notations

The following notations are utilized in this research:

- **NS** = number of stations on a vehicle route, excluding the central depot
- **TNS** = the total number of stations to be served, excluding the central depot
- **C** = the total cost of each vehicle route
- **C_{ij}** = the travel cost of moving from station i to j, \( C_{ii} = \infty \)
- **d_i** = the demand at station i (i = 1, 2, ..., TNS), is a random variable having a known distribution function
- **d_{ij}** = the distance between station i and station j, and \( d_{ii} = \infty \)
- **D** = a 1 x NS vector with components of \( d_i \)
- **I** = set of stations on a vehicle route, including the central depot, \( 0 \) stands for central depot
- **J** = I - \{0\}
- **M** = the mean of travel time on a vehicle route
- **NV** = number of vehicles
- **n(i)** = a set of negative deviations for constraint (i)
- **p(i)** = a set of positive deviations for constraint (i)
- **P(.)** = stands for probability of (.)
- **Q** = a vehicle capacity
- **\( \overline{Q} \)** = the artificial capacity of a vehicle
- **R** = the variance-covariance (dispersion) matrix for customer demand
- **S** = a set of feasible solutions for each vehicle route
- **S_{NV}** = a set of feasible solutions for NV trucks
SS = the safety stock

t_i = the unload time at station i, \(i = 1,2,\ldots,TNS\), is a random variable having a known distribution function

t_{ij} = the travel time from station i to station j, is a random variable having a known distribution function

\((i = 0,1,2,\ldots,TNS)\) and \((j = 0,1,\ldots,TNS)\) and \(t_{ii} = \infty\)

E = a 1 x NS vector with components of \(t_i\)

T1 = the maximum total travel time allowed on each vehicle route

T2 = the maximum total unload time allowed on each vehicle route

TR_k = a predetermined maximum total travel time allowed for the \(k^{th}\) vehicle route

TT = the total time required to complete a vehicle route

UT_k = a predetermined maximum total unload time allowed for the \(k^{th}\) vehicle route

V = the variance-covariance (dispersion) matrix for travel time

W = the variance-covariance (dispersion) matrix for unload time

X_{ij} = decision variables, 1 if a truck goes from station i to station j, 0 otherwise

G = a 1 x NS vector with components \(X_{ij}\)

X_{ijk} = decision variables, 1 if the \(k^{th}\) truck goes from station i to station j, 0 otherwise

\(\alpha,\beta,\gamma,\eta\) = the predetermined level of constraint violations, where

\(0 < \alpha \leq 1,\ 0 < \beta \leq 1,\ 0 < \gamma \leq 1,\) and \(0 < \eta \leq 1\)

\(\alpha_k,\beta_k,\eta_k\) = the predetermined level of constraint violation of the \(k^{th}\) truck, where \(0 < \alpha_k \leq 1,\ 0 < \beta_k \leq 1,\) and \(0 < \eta_k \leq 1\)

\(\mu_{d_i}\) = the mean of the demand at station \(i\)

\(\sigma_{d_i}^2\) = the variance of the demand at station \(i\)
\( \mu_{t_i} \) = the mean of the unload time at station \( i \)
\( \sigma^2_{t_i} \) = the variance of the unload time at station \( i \)
\( \mu_{t_{ij}} \) = the mean of the travel time between station \( i \) and \( j \)
\( \sigma^2_{t_{ij}} \) = the variance of the travel time between station \( i \) and \( j \)

\( \Psi \) = a constant value

\( U \) = a 1 x NS vector with components \( \mu_{t_i} \)
\( H \) = a 1 x NS vector with components \( \mu_{d_i} \)
\( \bar{T}_1 \) = a target level for travel time for each vehicle route
\( \bar{T}_2 \) = a target level for unload time for each vehicle route
\( T \) = a TNS \( \times \) TNS matrix with components of \( t_{ij} \)

5.3 Assumptions

The following assumptions were considered in this model building:

1. The demand at each destination is a random variable having a known distribution function

2. The unload time at each destination is a random variable having a known distribution function

3. The travel time from one station to another is a random variable having a known distribution function

4. The commodity to be transported is homogeneous

5. All vehicles have the same capacity

6. The shortest distance between two stations is considered to be euclidian

7. The maximum allowable total travel time of each vehicle route is \( T_1 \)
8. The maximum allowable total unload time of each vehicle route is $T_2$

9. $\alpha, \beta, \gamma, \eta, \alpha_k, \beta_k$, and $\eta_k$ are predetermined.

5.4 Route Improvement Stage: Problem Formulation

5.4.1 Time Constraints

The formulation to be presented is based on the previous assumptions and notations. The basic form of the problem is typified by the situation in which deliveries are made from a central depot to the destinations by $N_V$ vehicles. All goods as well as the $N_V$ trucks are assumed to be available for delivery at an arbitrary time zero. This formulation allows different predetermined conditions on each vehicle route. $T_1$ is considered to be the maximum value of the total travel time on each vehicle route for $100(1 - \alpha)$% of the time. On the other hand, the maximum value of the total unload time on each vehicle route is $T_2$ for $100(1 - \beta)$% of the time. In general, when the travel time between any link and unload time at each station are deterministic, then the total elapsed time on the vehicle route is

$$\text{Total elapsed time} = \sum_{i=0}^{NS} (t_{i,i+1} + t_i). \quad (5.1)$$

where 0 stands for depot, $t_0 = 0$, and $NS + 1$ is defined to be 0.

The DM is generally interested in minimizing the total travel cost and total travel and unload times of each vehicle route to target levels.
C, $\bar{T}1$, and $\bar{T}2$, respectively. Additionally, other criteria such as customer satisfaction may attract the attention of the DM in order to satisfy the customer's requirements. Hence, the multiple objective SVRP can be formulated having the following goals and constraints. However, the method of calculation of $\bar{T}1$ and $\bar{T}2$ are delayed until Section 5.7.

**Problem A**

**Goals:**

1. Minimize total travel cost or distance of each vehicle route

$$C = \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij}$$  \hspace{1cm} (5.2)

2. Minimize total travel and total unload times of each vehicle route to the target levels which are set to be $\bar{T}1$ and $\bar{T}2$, respectively.

3. Maximize the dependency conditions such that station $r$ follows station $s$.

**Constraints:**

$$P(\sum_{i \in I} \sum_{j \in I} t_{ij} x_{ij} \leq \bar{T}2) \geq (1 - \beta)$$  \hspace{1cm} (5.3)

$$P(\sum_{i \in I} \sum_{j \in I} t_{ij} x_{ij} \leq \bar{T}1) \geq (1 - \alpha)$$  \hspace{1cm} (5.4)

$$G = [x_{ij}] \in S$$  \hspace{1cm} (5.5)
where \( t_{ij} \) and \( t_i \) are independent random variables and assumed to be normally distributed with means \( \mu_{t_{ij}} \) and \( \mu_{t_i} \), and variances \( \sigma_{t_{ij}}^2 \) and \( \sigma_{t_i}^2 \), respectively. The \( T_1 \) and \( T_2 \) introduced above indicate suitable upper limits on travel and unload times for each vehicle route.

### 5.4.2 Demand Constraint

In the field of VRP, one of the difficulties which occurs in the application of mathematical programming is that the demands at stations are not constants but are either fluctuating or of uncertain values. However, used on the previous assumptions and notations, a probabilistic demand can be handled by using the concept of probabilistic constraints. For example, suppose that \( \eta \) is the maximum allowable probability that a vehicle route will fail due to the total probabilistic demand exceeding the truck capacity, then:

\[
P\left( \sum_{i \in I} \sum_{j \in I, i \neq j} x_{ij} \leq Q \right) \geq (1 - \eta) \tag{5.6}
\]

where \( d_i \)'s are independent random variables representing demand at location \( i \) and assumed to be normally distributed with mean \( \mu_{d_i} \) and variance \( \sigma_{d_i}^2 \), and \( Q \) is the truck capacity. The existence of the probabilistic customer demand forces the decision maker to minimize the safety stock (unused capacity) which is \( Q - \bar{Q} \). However, the method of calculation of \( \bar{Q} \) is delayed until Section 5.7. By incorporating this idea, Problem A can be modified to a more general form of a multicriteria SVRP, as presented in Problem B:
Problem B

Goal:

1. Minimize total travel cost or distance of each vehicle route

\[ C = \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} \]  \hspace{1cm} (5.7)

2. Minimize total travel and unload times of each vehicle route to the target levels which are set to be \( T_1 \) and \( T_2 \), respectively

3. Minimize the route safety stock

\[ SS = Q - \bar{Q} \]  \hspace{1cm} (5.8)

4. Maximize the dependency conditions such that station \( r \) follows station \( s \)

Constraints:

\[ P(\sum_{i \in I} \sum_{j \in I} t_{ij} x_{ij} \leq T_2) \geq (1 - \beta) \]  \hspace{1cm} (5.9)

\[ P(\sum_{i \in I} \sum_{j \in I} t_{ij} x_{ij} \leq T_1) \geq (1 - \alpha) \]  \hspace{1cm} (5.10)

\[ P(\sum_{i \in I} \sum_{j \in I} d_{ij} x_{ij} \leq Q) \geq (1 - \eta) \]  \hspace{1cm} (5.11)

\[ G = [X_{ij}] \in S. \]  \hspace{1cm} (5.12)
It is worth noting that when travel and unload times are probabilistic, then Problem A should be chosen for the purpose of the GP formulation of the RIS of the problem. On the other hand, when customer demand and travel and unload times are probabilistic, then Problem B should be used to construct the GP formulation of the RIS of the problem.

Problems A and B are used in Section 5.8 for the purpose of the GP formulation of the SVRP.

5.5 Development of the Deterministic Forms for the Set of Constraints of Problem B

It has been shown [39] that it is possible to deal effectively with random variables in the constraint set of a stochastic programming problem. When random variables appear in the constraint set, deterministic equivalents must be derived to replace the original chance-constrained inequalities. Therefore, this section is devoted to the development of the equivalent deterministic form of the constraints of Problem B. The following subsections consider each constraint separately:

1. Deterministic form for unload time constraints
2. Deterministic form for travel time constraints
3. Deterministic form for demand constraints

5.5.1 Deterministic Form for Unload Time Constraint

The first constraint of Problem B is called the unload time constraint which is written as
\[ P(\sum_{i \neq j} \sum_{i \in I} \sum_{j \in I} t_{ij} X_{ij} \leq T_2) = (1 - \beta). \]  
(5.13)

Now one can consider an NS component vector \( E = (t_1, t_2, \ldots, t_{NS}) \) as a multivariate normal with mean vector \( U = (\mu_{t_1}, \mu_{t_2}, \ldots, \mu_{t_{NS}}) \) and variance-covariance matrix

\[
W = \begin{pmatrix}
\sigma_{t_1}^2 & \cdots & 0 \\
\cdots & \ddots & \cdots \\
0 & \cdots & \sigma_{t_{NS}}^2
\end{pmatrix}
\]  
(5.14)

where the \( t_i \)'s are independent random variables corresponding to the unload time at each of the \( i \) stations. Let \( Z = (Z_1, Z_2, \ldots, Z_{NS}) \), be a vector of NS elements where each component of \( Z \) is

\[ Z_i = \sum_{j \in I} X_{ij} \quad \forall i \in I, i \neq j \]  
(5.15)

According to multivariate statistical analysis [9, 34, 39], the linear combination \( Z'E \) is univariate normal with mean \( Z'U \) and variance \( Z'WZ \).

Therefore

\[ P(Z'E \leq T_2) = P((Z'E - Z'U)/(Z'WZ)^{1/2} \leq (T2 - Z'U)/(Z'WZ)^{1/2}) = (1 - \beta). \]  
(5.16)

The above inequality exists if and only if

\[ N((T2 - Z'U)/(Z'WZ)^{1/2}) = (1 - \beta) \]

or

\[ (T2 - Z'U)/(Z'WZ)^{1/2} = N^{-1}(1 - \beta). \]
Finally, the deterministic form of (5.13) is

\[ Z'U + N^{-1}(1 - \beta)(Z'WZ)^{1/2} = T2 \]  

(5.17)

where \( Z'U \) and \((Z'WZ)^{1/2}\) are the mean and standard deviation of the unload time on the vehicle route and \( N^{-1}(1 - \beta) \) is the normalized deviate corresponding to the required probability.

\[ \beta = \frac{1}{(2\pi)^{1/2}} \int_{-\infty}^{N^{-1}(1 - \beta)} \exp(-x^2/2) \, dx. \]  

(5.18)

If the required probability \((1 - \beta) = 0.95\), then \( N^{-1}(1 - \beta) = 1.645 \) from the normal table [34, pp. 592-593]. The evaluated deterministic form given above is generally nonlinear in nature, which can be seen when it, (5.17), is written in the following form:

\[ \sum_{i \neq j} \sum_{i \in I} \mu_{t_{ij}} x_{ij} + N^{-1}(1 - \beta)(\sum_{i \in I} \sum_{j \in I} \sigma_{t_{ij}}^2 x_{ij}^2)^{1/2} = T2. \]  

(5.19)

5.5.2 Deterministic Form for Travel Time Constraint

The second constraint of problem B (inequality (5.10)), called the travel time constraint, is written as

\[ P(\sum_{i \in I} \sum_{j \in I} t_{ij} x_{ij} \leq T1) = (1 - \alpha). \]  

(5.20)

Suppose that \( \mu_{t_{ij}} \) and \( \sigma_{t_{ij}}^2 \) are the mean and variance of \( t_{ij} \) where \( t_{ij}'s \) are independent random variables corresponding to the travel time from
station i to station j and are assumed to be normally distributed. Then, one can consider a travel time vector \( T = (t_{ij}) \) \( \forall i,j, i \neq j \), as a multivariate normal with mean \( M = (\mu_{t_{ij}}) \) \( \forall i,j, i \neq j \), and variance-covariance matrix \( V \). Let \( G = [X_{ij}], \forall i,j, i \neq j \), be a vector of all elements \( X_{ij} \) which are arranged in the same order as the elements of \( T = (t_{ij}), \forall i,j, i \neq j \). According to multivariate statistical analysis, the linear combination \( G'T \) is univariate normal with mean \( G'M \) and variance \( G'VG \). Hence:

\[
P(G'T \leq T_l) = P((G'T - G'M)/(G'VG) \leq (T_l - G'M)/(G'VG)). \tag{5.21}
\]

Thus, by the definition of a cumulative distribution function, i.e., \( N_X(x) = P(X \leq x) \) where \( X \) is a random variable, one can write

\[
N((T_l - G'M)/(G'VG)) = (1 - \alpha).
\]

Then, after similar calculations as previously considered

\[
G'M + N^{-1}(1 - \alpha) (G'VG) = T_l. \tag{5.22}
\]

where \( G'M \) and \( (G'VG) \) are the mean and standard deviation of the travel time on a vehicle route and \( N^{-1}(1 - \alpha) \) is the normal deviate as described in Section 5.5.1. Constraint (5.22) is generally nonlinear in terms of \( X_{ij} \) as shown below:

\[
\sum_{i\in I} \sum_{j\in I} \mu_{t_{ij}} X_{ij} + N^{-1}(1 - \alpha) (\sum_{i\in I} \sum_{j\in I} \sigma_{t_{ij}}^2 X_{ij}^2) = T_l. \tag{5.23}
\]
5.5.3 Evaluation of the Total Time TT

This section determines the nature of the total elapsed time of each vehicle route in terms of the decision variables, $X_{ij}$, when decision maker needs to use $TT = T_1 + T_2$ as a criterion. Using equations (5.17) and (5.22), one can write:

$$TT = T_1 + T_2 = (X'M + Z'U) + [N^{-1}(1 - \alpha)(X'VX)^{\frac{1}{2}} + N^{-1}(1 - \beta)(Z'WZ)^{\frac{1}{2}}].$$

(5.24)

If $\alpha < 0.5$ and $\beta < 0.5$, then $N^{-1}(1 - \alpha)$ and $N^{-1}(1 - \beta) > 0$. Therefore, the above constraint can be written as:

$$(TT - X'M - Z'U)^2 = (N^{-1}(1 - \alpha)(X'VX)^{\frac{1}{2}} + N^{-1}(1 - \beta)(Z'WZ)^{\frac{1}{2}})^2$$

(5.25)

which is a quadratic function in terms of $X_{ij}$.

5.5.4 Deterministic Form for the Demand Constraint

This section is devoted to derivation of the deterministic form of the demand constraint where $d_i$'s are independent random variables corresponding to the demand at station $i$ and are assumed to be normally distributed. The demand constraint can be rewritten in the form

$$P(\sum_{i \in I} \sum_{j \in I, i \neq j} d_i X_{ij} \leq Q) = (1 - \eta).$$

(5.26)

If $\mu_{d_i}$ and $\sigma_{d_i}^2$ are the mean and variance of $d_i$, then one can consider an NS component vector $D = (d_1, d_2, \ldots, d_{NS})$ as a multivariate normal with mean vector $H = (\mu_{d_1}, \mu_{d_2}, \ldots, \mu_{d_{NS}})$ and variance-covariance matrix $R$. Let $Y = (Y_1, Y_2, \ldots, Y_{NS})$ be a vector of NS elements where each component of $Y$ is
\[ Y_i = \sum_{j \in I} X_{ij} \quad \forall i \in I, \ i \neq j. \] (5.27)

Again, according to multivariate statistical analysis, the linear combination \( Y'D \) is univariate normal with mean \( Y'H \) and variance \( Y'RY \). Hence,

\[ P(Y'D \leq Q) = P((Y'D - Y'H)/(Y'RY)^{1/2} \leq (Q - Y'H)/(Y'RY)^{1/2}) \]

or

\[ ((Q - Y'H)/(Y'RY)^{1/2}) = N^{-1}(1 - \eta). \] (5.28)

Then

\[ Q = Y'H + N^{-1}(1 - \eta)(Y'RY)^{1/2} \] (5.29)

where \( Y'H \) and \( (Y'RY)^{1/2} \) are the mean and standard deviation of the demand on the vehicle route and \( N^{-1}(1 - \eta) \) is the normal deviate as previously described. This deterministic form is also nonlinear in nature, as can be seen when it is written in expanded form as

\[ \sum_{i \in I} \sum_{j \in I} \mu_{d_i} X_{ij} + N^{-1}(1 - \eta) \sum_{i \in I} \sum_{j \in I} \sigma_{d_i}^2 X_{ij}^{2/2} = Q. \] (5.30)

### 5.5.5 Safety Stock and Surplus in Term of the Decision Variables \( X_{ij} \)

Before the development of the safety stock and surplus in term of the decision variables begins, it is necessary to define the following terms. Golden and Stewart [30, pp. 253-254] have defined the primary and secondary error as follows:
Primary Error

A primary error occurs when a vehicle cannot satisfy the demands of the customers on the route to which it has been assigned.

Secondary Error

A secondary error occurs when a vehicle returns to the central depot after satisfying the demands on its route with more than 100(1 - a) percent of its original load, where 0 ≤ a ≤ 1.

The primary error requires an additional trip to the central depot which causes additional cost and service delay. On the other hand, the existence of the secondary error is a waste of load and unload times and in some cases it is a waste of products (i.e., perishable goods).

By considering a delivery problem with one central depot, NS demand points, and a vehicle capacity Q with probabilistic demand $d_i$ for the $i^{th}$ customer, then by appealing to the Central Limit Theorem, one can argue that the total route demand, $TD$, is approximately normally distributed where

$$TD = d_1 + d_2 + \ldots + d_{NS}. \quad (5.31)$$

If $d_i$ are poisson distributed with mean and variance $\mu_{d_i}$, then the mean and variance of the total demand on the route are, respectively:

$$E(TD) = \mu_{d_1} + \mu_{d_2} + \ldots + \mu_{d_{NS}} \quad (5.32)$$

and

$$\sigma^2(TD) = E(TD). \quad (5.33)$$
The primary and secondary errors of each vehicle route are, respectively

\[ P(TD \geq Q) = P(z \geq (TD - E(TD))/\sigma(TD)) \] (5.34)

and

\[ P(TD \leq aQ) = P(z \leq (aQ - E(TD))/\sigma(TD)) \text{ where } 0 < a \leq 1 \] (5.35)

and \( z \) is a unit normal deviate.

One may treat the primary error as \( P(TD \geq Q) \leq \eta \) where \( 0 < \eta < 0.5 \) by incorporating the concept of an artificial capacity of a truck, \( \overline{Q} \), where \( \overline{Q} < Q \),

\[ P(z \geq (Q - \overline{Q})/\sigma(TD)) = \eta \] (5.36)

which has an equivalent deterministic form of

\[ Q = \overline{Q} + N^{-1}(1 - \eta) \sigma(TD). \] (5.37)

Notice that (5.37) is similar to (5.29). The quantity of \( Q - \overline{Q} \) is called the route safety stock (SS) which is protection against the primary error. Since \( (1 - \eta) > 0.5 \), then \( N^{-1}(1 - \eta) \) is positive. Using the SS as a criteria, one will have the following expression in terms of the decision variables \( X_{ij} \):

\[ SS = N^{-1}(1 - \eta)(\sum_{i \in I} \sum_{j \in I} \sum_{i \neq j} 2 \sigma_{d_{ij}} X_{ij}^{2})^{1/2}. \] (5.38)

The secondary error as defined in (5.35)

\[ P(TD \leq aQ) = P(z \leq (aQ - E(TD))/\sigma(TD)) = \gamma \]
with the following equivalent deterministic form:

\[ aQ = E(TD) + N^{-1}(\gamma)\sigma(TD). \] (5.39)

In this case, the quantity \( Q - aQ \) is the extra number of units of products carried on the vehicle routes if trucks are loaded up to their \( Q \) capacity. To minimize the carrying of these units on the vehicle route one may use the following nonlinear objective function:

\[
Q - \sum_{i \in I} \sum_{j \neq i} \mu_{d_{ij}} X_{ij} - N^{-1} (\gamma) \left( \sum_{i \in I} \sum_{j \neq i} \sigma_{d_{ij}}^2 X_{ij}^2 \right)^{1/2}.
\] (5.40)

The GP formulation of the RIS of the problem is delayed until Section 5.8.

5.6 Route Construction Stage:

Problem Formulation

The problem formulation of this stage is divided into two sections according to the type of criteria that is to be minimized. The criteria to be considered are:

1. Total cost (or distance) as presented in problem C, and
2. Total time as presented in problem D.

When cost (or distance) is considered as a criteria, the objective function is linear in terms of the decision variables \( X_{ijk} \). On the other hand, when total time is considered to be minimized, the objective function becomes nonlinear in terms of the decision variables \( X_{ijk} \).
5.6.1 Using Cost as a Criterion

Problem C

Minimize: \[ C = \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} c_{ij} x_{ijk} \] \hspace{1cm} (5.41)

Subject to:

\[ P(\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} t_{ij} x_{ijk} \leq TR_k) \geq (1 - \alpha_k), \ k = 1, \ldots, NV \] \hspace{1cm} (5.42)

\[ P(\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} t_{ij} x_{ijk} \leq UT_K) \geq (1 - \beta_k), \ k = 1, \ldots, NV \] \hspace{1cm} (5.43)

\[ P(\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} d_{ij} x_{ijk} \leq Q) \geq (1 - \eta_k), \ k = 1, \ldots, NV \] \hspace{1cm} (5.44)

\[ X = [x_{ijk}] \in S_{NV}. \] \hspace{1cm} (5.45)

5.6.2 Using Total Time as a Criterion

Problem D

Minimize: \[ \sum_{k=1}^{NV} [TR_k + UT_k] \] \hspace{1cm} (5.46)

Subject to:

\[ P(\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} t_{ij} x_{ijk} \leq TR_k) \geq (1 - \alpha_k), \ k = 1, \ldots, NV \] \hspace{1cm} (5.47)

\[ P(\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} t_{ij} x_{ijk} \leq UT_K) \geq (1 - \beta_k), \ k = 1, \ldots, NV \] \hspace{1cm} (5.48)
\[ P \left( \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} d_{ij} X_{ijk} \leq Q \right) \geq (1 - \eta_k), \quad k = 1, \ldots, NV \]  

(5.49)

\[ X = [X_{ijk}] \in S_{NV} \]  

(5.50)

where \( S_{NV} \) is the set of all feasible solutions to the NV travelling salesman problem, and \( \alpha_k, \beta_k, \) and \( \eta_k \) \((k = 1, \ldots, NV)\) are the probability of constraint infeasibility on the \( k^{th} \) route by violating the predetermined levels \( TR_k, UT_k, \) and \( Q, \) respectively. The process of transformation of the above probabilistic constraints to their equivalent deterministic forms are similar to those shown previously. Without loss of generality and for the sake of space, the deterministic forms of Problems C and D are shown in following section.

However, the equivalent deterministic form of each situation is completely different and, hence, each requires a different solution technique.

5.6.3 Equivalent Deterministic Forms of Problems C and D of the RCS of the Problem

The equivalent deterministic form of Problem C is

**Problem E**

Minimize:  
\[ C = \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} C_{ij} X_{ijk} \]  

(5.51)
Subject to:

\begin{equation}
\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \mu_{t_{ij}} x_{ijk} + N^{-1}(1 - \alpha_k) \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sigma_{t_{ij}}^2 x_{ijk}^2 \right)^{1/2} \leq TR_k \quad (5.52)
\end{equation}

\begin{equation}
\sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{d_i} x_{ijk} + N^{-1}(1 - \beta_k) \left( \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \sigma_{d_i}^2 x_{ijk}^2 \right)^{1/2} \leq UT_k \quad (5.53)
\end{equation}

\begin{equation}
\sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{d_i} x_{ijk} + N^{-1}(1 - \eta_k) \left( \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \sigma_{d_i}^2 x_{ijk}^2 \right)^{1/2} \leq Q \quad (5.54)
\end{equation}

\begin{equation}
X = [x_{ijk}] \in S_{NV} \quad \text{and} \quad k = 1, 2, \ldots, NV. \quad (5.55)
\end{equation}

Similarly, the equivalent deterministic form of Problem D is

**Problem F**

Minimize:

\begin{equation}
\sum_{k=1}^{NV} \left( \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \mu_{t_{ij}} x_{ijk} \right) + \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{t_{ij}} x_{ijk} \right) + [N^{-1}(1 - \alpha_k) \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sigma_{t_{ij}}^2 x_{ijk}^2 \right)^{1/2}]
\end{equation}

\begin{equation}
+ \left( \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{d_i} x_{ijk} \right) + \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{d_i} x_{ijk} \right) \left( [N^{-1}(1 - \beta_k) \left( \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \sigma_{d_i}^2 x_{ijk}^2 \right)^{1/2}]ight)
\end{equation}

\begin{equation}
\quad \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{d_i} x_{ijk} + N^{-1}(1 - \eta_k) \left( \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \sigma_{d_i}^2 x_{ijk}^2 \right)^{1/2} \leq Q \quad (5.56)
\end{equation}

Subject to:

\begin{equation}
\sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \mu_{d_i} x_{ijk} + N^{-1}(1 - \eta_k) \left( \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \sigma_{d_i}^2 x_{ijk}^2 \right)^{1/2} \leq Q \quad (5.57)
\end{equation}

\begin{equation}
X = [x_{ijk}] \in S_{NV}. \quad \text{and} \quad k = 1, 2, \ldots, NV \quad (5.58)
\end{equation}
The deterministic forms of problems C and D will be called "E" and "F" type problems.

The most important characteristic of the "E" type problem is that while the decision maker desires to minimize total expected cost or distance of the whole delivery system, he can also restrict the travel and unload times of each single vehicle route.

On the other hand, the "F" type problem deals with the minimization of total elapsed time of the whole delivery system with no restriction on the travel and unload times of each vehicle route.

It is the decision maker's responsibility to determine which of these two problems are suitable for the delivery system. These two problems offer two benefits. One benefit is that they support the decision maker, quantitatively, in attempting to make a good decision. The decision maker may be willing to change some or all upper bounds, \( T_R^k \), \( U_T^k \), of each vehicle route when a previous solution was not favorable. The second benefit is that the "F" type problem allows the decision maker to minimize total elapsed time of the whole delivery system without time restriction put on each single vehicle route.

Since normal distributions are symmetrical about their mean, the objective function (5.59) is identical with the expected value reformulation of (5.56) when \( \alpha_k = 0.5 \) and \( \beta_k = 0.5 \) for all \( k \in \{1, 2, \ldots, NV\} \), which yields:

\[
\text{Minimize: } \quad E(\Sigma_{i=0}^{TNS} \Sigma_{j=0}^{TNS} [t_{ij} X_{ijk} + \Sigma_{i=1}^{TNS} \Sigma_{j=0}^{TNS} t_i X_{ijk}])
\]

(5.59)
from which one obtains the expression

\[
\begin{align*}
\text{Minimize:} & \quad \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} x_{ijk} + \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} x_{ijk} \\
& \quad \text{subject to:} \quad \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{d_{ij}} x_{ijk} \leq Q \quad k = 1, \ldots, NV
\end{align*}
\]

when \( \alpha_k = \beta_k = \eta_k = 0.5 \) for all \( k \in (1, 2, \ldots, NV) \) then problem (5.56) – (5.58) is converted to the following problem.

\[
\begin{align*}
\text{Minimize:} & \quad \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} x_{ijk} + \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} x_{ijk} \\
& \quad \text{subject to:} \quad \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{d_{ij}} x_{ijk} \leq Q \quad k = 1, \ldots, NV
\end{align*}
\]

When \( 0.5 < (1 - \alpha_k), (1 - \beta_k), (1 - \eta_k) < 1 \) for all \( k \in (1, 2, \ldots, NV) \), which is reasonable to assume, \( N^{-1}(1 - \alpha_k), N^{-1}(1 - \beta_k), \) and \( N^{-1}(1 - \eta_k) \) are all greater than zero, then (5.56) and (5.58) are convex, since \( (X'VX)^{\frac{1}{2}} \) is a convex function. Therefore, problem (5.56) – (5.58) is a convex programming problem.

The corresponding deterministic form of the previous problem, shown in (5.56) – (5.58), is shown below with \( q_k = N^{-1}(1 - \alpha_k) \) and \( f_k = N^{-1}(1 - \beta_k) \) and \( e_k = N^{-1}(1 - \eta_k) \) for \( k = (1, 2, \ldots, NV) \):

\[
\begin{align*}
\text{Maximize:} & \quad - \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} x_{ijk} + \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} x_{ijk}
\end{align*}
\]
where constraint set $AX \leq b$ is equivalent to constraints (2.8), (2.9), (2.11), and (2.12) where $N = TNS$ and depot is node 0.

Next, consider the following quadratic programming problem which is (5.62) with $R_{1k}$ and $R_{2k}$ inserted as shown:

Maximize:

\[
- \sum_{k=1}^{NV} \left[ \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} X_{ijk} + \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} X_{ijk} \right]
\]

Subject to:

\[
- \sum_{k=1}^{NV} \left[ (q_k/2R_{1k}) \left( \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{t_{ij}}^2 X_{ijk} \right) + (f_k/2R_{2k}) \left( \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{t_{ij}}^2 X_{ijk} \right) \right]
\]

Subject to:

\[
\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{d_{ij}} X_{ijk} + e_k \left( \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{d_{ij}}^2 X_{ijk} \right) \leq Q,
\]

\[
k = 1, \ldots, NV
\]

\[
AX \leq b
\]
for all $k \in \{1, 2, \ldots, NV\}$. The following theorem provides the optimum solution for problem (5.56) – (5.58).

**Theorem (5.1)**

If an optimal solution $X(R_{1k}, R_{2k})$ of problem (5.65) – (5.67) satisfies the conditions $R_{1k}$ and $R_{2k}$ as shown in (5.68) and (5.69), then $\hat{X}(R_{1k}, R_{2k})$ is also the optimal solution vector of problem (5.62) – (5.64).

**Proof**

For problem (5.62) – (5.64) the Lagrangian function is

\[
F_I(X, \lambda, \mu) = \sum_{k=1}^{NV} \left[ \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{1i} x_{ijk} + \sum_{i=1}^{TNS} \mu_{2i} x_{ijk} \right] - \sum_{i \neq j}^{NV} \sum_{k=1}^{TNS} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{1ij} x_{ijk}^2 + \sum_{i \neq j}^{NV} \sum_{k=1}^{TNS} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{1ij} x_{ijk}^2
\]

\[
+ \sum_{k=1}^{NV} \sum_{i \neq j}^{TNS} \sum_{k=1}^{NV} \left[ (Q - \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{d1} x_{ijk} - e_k (\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma_{d1} x_{ijk}^2))^2 \right] + \mu(b - AX).
\]

(5.70)
The vector \( \hat{X} = [X_{ijk}] \) for all \( i,j,k \) is an optimal solution for (5.62)–(5.64), if \( \hat{X} \) and \( \lambda = (\lambda_1, \ldots, \lambda_k) \) and \( \mu \) satisfy the following conditions:

\[
\frac{\partial F}{\partial X_{ijk}} = - \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \left[ \sum_{i \neq j}^{TNS} \mu_{t_{ij}} + \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{t_{ij}} \right] i \neq j
\]

\[
- \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} q_k \frac{(\sum_{i \neq j}^{TNS} \sigma^2 X_{ijk})}{(\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma^2 t_{ij})} i \neq j
\]

\[
- \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} f_k \frac{(\sum_{i \neq j}^{TNS} \sigma^2 X_{ijk})}{(\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma^2 t_{ij})} i \neq j
\]

\[
+ \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \lambda_k \left( \sum_{i \neq j}^{TNS} \mu_{d_i} - e_k \frac{(\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma^2 X_{ijk})}{(\sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma^2 t_{ij})} i \neq j \right)
\]

\[
= 0 \quad \text{for } X_{ijk} > 0
\]

\[\frac{\partial F}{\partial \lambda_k} = Q - \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{d_i} X_{ijk} - e_k \frac{(\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma^2 X_{ijk})}{(\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma^2 t_{ij})} i \neq j
\]

\[
\geq 0 \quad \text{for all } \lambda_k = 0
\]

\[
= 0 \quad \text{for } \lambda_k > 0
\]

\[k = 1, 2, \ldots, NV\]

\[
\frac{\partial F}{\partial \mu} = (b - AX)
\]

\[
\geq 0 \quad \text{for } \mu = 0
\]

\[
= 0 \quad \text{for } \mu > 0
\]
Similarly, the Kuhn-Tucker conditions of the problem (5.65) – (5.67) can be written as shown below. If \( \hat{x} = [X_{ijk}] \) \( \forall \ i, j, k \) and \( \hat{x}, \hat{\lambda}, \hat{\mu} \) satisfy the following conditions, then it is the optimum solution of problems (5.65) – (5.67).

\[
\begin{align*}
FII(X, \lambda, \mu) &= -\sum_{k=1}^{N} \left[ \sum_{i=1}^{T} \sum_{j=0}^{N} \mu_{t_{ij}} x_{ijk} + \sum_{i=1}^{T} \sum_{j=0}^{N} \mu_{t_{ij}} x_{ijk} \right]_{i \neq j} \\
&- \sum_{k=1}^{N} \left[ (q_k/2R_1) \left( \sum_{i=0}^{T} \sum_{j=0}^{N} \sigma_{t_{ij}}^2 x_{ijk}^2 \right) + (f_k/2R_2) \left( \sum_{i=1}^{T} \sum_{j=0}^{N} \sigma_{t_{ij}}^2 x_{ijk}^2 \right) \right]_{i \neq j} \\
- \sum_{k=1}^{N} \lambda_k \left[ \sum_{i=1}^{T} \sum_{j=0}^{N} \mu_{d_{ij}} x_{ijk} - e_k \left( \sum_{i=1}^{T} \sum_{j=0}^{N} \sigma_{d_{ij}}^2 x_{ijk} \right) \right]_{i \neq j} + \mu(b - AX) \\
\frac{\partial FII}{\partial x_{ijk}} &= -\sum_{k=1}^{N} \left[ \sum_{i=0}^{T} \sum_{j=0}^{N} \mu_{t_{ij}} + \sum_{i=1}^{T} \sum_{j=0}^{N} \mu_{t_{ij}} \right]_{i \neq j} \\
&- \sum_{k=1}^{N} \left[ (q_k/R_1) \left( \sum_{i=0}^{T} \sum_{j=0}^{N} \sigma_{t_{ij}}^2 x_{ijk} \right) - \sum_{k=1}^{N} \left( f_k/R_2 \right) \left( \sum_{i=1}^{T} \sum_{j=0}^{N} \sigma_{t_{ij}}^2 x_{ijk} \right) \right]_{i \neq j} \\
&+ \sum_{i=1}^{T} \lambda_k \left( \sum_{i=0}^{T} \sum_{j=0}^{N} \mu_{d_{ij}} - e_k \left( \sum_{i=1}^{T} \sum_{j=0}^{N} \sigma_{d_{ij}}^2 x_{ijk} \right) / \left( \sum_{i=0}^{T} \sum_{j=0}^{N} \sigma_{d_{ij}}^2 \right) \right)_{i \neq j} \\
&\leq 0 \text{ for } x_{ijk} = 0 \\
&= 0 \text{ for } x_{ijk} > 0
\end{align*}
\]
\[
\frac{\partial FII}{\partial \lambda_k} = Q - \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{d_i} X_{ijk} - \epsilon_k (\sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma^2_{d_i} X_{ijk})^{\frac{1}{2}} \\
\begin{cases} 
\geq 0, \lambda_k = 0 \\
= 0, \lambda_k > 0 
\end{cases}
\] (5.76)

\[
\frac{\partial FII}{\partial \mu} = (b - AX) \begin{cases} 
\geq 0, \mu = 0 \\
= 0, \mu > 0 
\end{cases}
\] (5.77)

Hence, if equations (5.68) and (5.69) exist, then the conditions of both problems are identical. Therefore, \( \hat{X}(R_1, R_2) \) is also an optimal solution of (5.62) – (5.64). Conversely, for any optimal solution \( \hat{X} \) of (5.62) – (5.64), if \( R_1, R_2 \) is set such that it satisfies (5.68) and (5.69), \( \hat{X} \) also satisfies the condition (5.65) – (5.67).

This theorem indicates that an optimal solution to the "F" type problems exists. Since this problem is nonlinear and decision variables are in 0-1 form, seeking the optimum solution by exact procedure is not efficient. Hence, heuristic approaches are considered for solving "E" and "F" type problems.

5.7 Distributions Other Than Normal

The deterministic constraints (5.19), (5.23), and (5.30) can be replaced with some other constraints in an easier form, if the time and demand distributions are of the same special forms. There are several distributions that satisfy the following condition:

\[
\sigma^2_1 = \psi \mu_1
\] (5.78)
This means that the variance is some constant multiple of the mean of that distribution. Distributions such as Poisson and chi-square satisfy the above condition. The value of $\Psi$ for these distributions are

1. Poisson
   \[ \sigma_i^2 = \mu_i \quad \Psi = 1 \]

2. Chi-square
   \[ \mu_i = \nu \]
   \[ \sigma_i^2 = 2\nu \quad \Psi = 2 \]

The following theorem shows the existence of a set of deterministic linear time and demand constraints which are equivalent to the nonlinear set of the time and demand constraints of the RIS problem.

**Theorem 5.2**

Under the following conditions:

1. The probability distributions of $t_{ij}$ are independent and stable [62, p. 84], and $\sigma_{t_{ij}}^2 = \Psi \mu_{t_{ij}}$

2. The probability distributions of $t_i$ are independent and stable, and $\sigma_{t_i}^2 = \Psi \mu_{t_i}$

3. The probability distributions of $d_i$ are independent and stable, and $\sigma_{d_i}^2 = \Psi \mu_{d_i}$

then there exist values $\bar{T}_1$, $\bar{T}_2$, and $\bar{Q}$ such that

\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} x_{ij} = \bar{T}_1 \]

\[ \sum_{i \neq j} i \neq j \] (5.79)
\[
\sum_{i \in I} \sum_{j \in I, j \neq i} \mu_{t_{ij}} x_{ij} = T_2 \tag{5.80}
\]

\[
\sum_{i \in I} \sum_{j \in I, j \neq i} \mu_{d_{ij}} x_{ij} = Q \tag{5.81}
\]

which are equivalent to the deterministic constraint (5.23), (5.19), and (5.30), respectively.

**Proof**

The proof is developed only for (5.79). One can prove similarly for (5.80) and (5.81). Since decision variable \(x_{ij}\) is either zero or 1, then,

\[x_{ij} = x_{ij}^2.\]

Therefore

\[
\sigma_{t_{ij}}^2 x_{ij} = \sigma_{t_{ij}}^2 x_{ij}^2
\]

\[
\sum_{i \in I} \sum_{j \in I, j \neq i} \sigma_{t_{ij}}^2 x_{ij} = \sum_{i \in I} \sum_{j \in I, j \neq i} \sigma_{t_{ij}}^2 x_{ij}^2
\]

or

\[
\left[ \sum_{i \in I} \sum_{j \in I, j \neq i} \sigma_{t_{ij}}^2 x_{ij} \right]^\frac{1}{2} = \left[ \sum_{i \in I} \sum_{j \in I, j \neq i} \sigma_{t_{ij}}^2 x_{ij}^2 \right]^\frac{1}{2} = \left[ \sum_{i \in I} \sum_{j \in I, j \neq i} \psi \mu_{t_{ij}} x_{ij} \right]^\frac{1}{2}
\]

\[
= \psi^\frac{1}{2} \left[ \sum_{i \in I} \sum_{j \in I, j \neq i} \mu_{t_{ij}} x_{ij} \right]^\frac{1}{2}.\]
Substitute (5.82) in equality (5.23), then

\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} X_{ij} + N^{-1}(1 - \alpha)\Psi^{\frac{1}{2}} \left[ \sum_{i \in I} \sum_{j \in I} \sigma_{t_{ij}} X_{ij} \right]^{\frac{1}{2}} = T_1. \]  
(5.83)

Let

\[ v = \left[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} X_{ij} \right]^{\frac{1}{2}} \quad \text{and} \quad N^{-1}(1 - \alpha) = \psi \]

then

\[ T_1 = v^2 + \phi \Psi^{\frac{1}{2}} v \]

\[ v^2 + \phi \Psi^{\frac{1}{2}} v - T_1 = 0. \]

Solve for \( v \)

\[ v = \left[ -\phi \Psi^{\frac{1}{2}} + (\phi^2 \Psi + 4T_1)^{\frac{1}{2}} \right]/2. \]

However,

\[ v^2 = \left[ (-\phi \Psi^{\frac{1}{2}} + (\phi^2 \Psi + 4T_1)^{\frac{1}{2}})/2 \right]^2 = \bar{T}_1. \]
(5.84)

Hence,

\[ v^2 = \left[ \left( \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} X_{ij} \right)^{\frac{1}{2}} \right]^2 = \bar{T}_1, \]

or

\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} X_{ij} = \bar{T}_1. \]
(5.85)

A similar analysis yields

\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} X_{ij} = \bar{T}_2. \]
(5.86)
\[
\sum_{i \in I} \sum_{j \in I, i \neq j} \mu_{d_i} x_{ij} = Q. \tag{5.87}
\]

5.8 Goal Programming Formulation

The GP approach is based on a priority structure of the established goals. In other words, the technique provides a solution according to the policy of the decision maker. The decision maker is thus required to determine the priority of the desired attainment of each goal and rank them in ordinal sequence for decision analysis.

The purpose of this section is to demonstrate the application of GP to decision problems in the area of SVRP. The model presented here has a priority of goals as follows:

1. All routes are feasible
2. Minimize the total cost of each vehicle route
3. Minimize the total travel and unload times
4. Minimize the route safety stock
5. Maximize the customer's satisfaction through the emergency service for the \(k\)th customer
6. Meet the dependency conditions such that station \(r\) follows station \(s\)

The Linear Integer Goal Programming (LIGP) formulation of Problem B is provided by utilizing the results of Theorem (5.2). Constraints (5.92) and (5.93) will change to nonlinear forms when random variables other than poisson and chi-square are utilized.
System constraints:

1. Route feasibility

\[ \sum_{j \in I} x_{ij} + n(1) - p(1) = 1 \text{ for all } i \in I \quad (5.88) \]
\[ i \neq j \]

\[ \sum_{i \in I} x_{ij} + n(2) - p(2) = 1 \text{ for all } j \in I \quad (5.89) \]
\[ i \neq j \]

\[ z_i - z_j + N x_{ij} + n(3) - p(3) = N - 1 \quad (5.90) \]
\[ \forall i, j \in I, i \neq j \]
\[ \text{and } i, j \neq 0. \]

Goal constraints:

2. Total cost of each vehicle route

\[ \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} + n(4) - p(4) = C \quad (5.91) \]
\[ i \neq j \]

3. Total elapsed time of each vehicle route

\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} x_{ij} + \sum_{i \in I} \sum_{j \in I} \mu_{t_i} x_{ij} + n(5) - p(5) = \overline{T1} + \overline{T2}. \quad (5.92) \]
\[ i \neq j \]
\[ i \neq j \]

When the minimization of the total travel and unload times of each vehicle route to target levels \( \overline{T1} \) and \( \overline{T2} \) are required, the goal constraint (5.92) can be divided into the following constraints, respectively:

\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} x_{ij} + n'(5) - p'(5) = \overline{T1} \text{ and } \]
\[ \sum_{i \in I} \sum_{j \in I} \mu_{t_{ij}} x_{ij} + n'(5) - p'(5) = \overline{T1} \]
\[ n'(5) = n(5) \]
\[ p'(5) = p(5) \]
\[ \sum \sum \mu_{ij} X_{ij} + n^\prime(5) - p^\prime(5) = T^2 \]

4. Route safety stock

\[ \sum \sum \mu_{ij} X_{ij} + n(6) - p(6) = [(Q - \bar{Q})/N^{-1}(1 - \eta)]^2 = \text{a constant} \] \hspace{1cm} (5.93)

5. Emergency service for the kth customer

\[ X_{0k} + n(7) - p(7) = 1 \] \hspace{1cm} (5.94)

where all \( X_{ij} \)'s are either 0 or 1.

6. Meet the dependency conditions such that station \( r \) follows station \( s \)

\[ X_{rs} + n(8) - p(8) = 1 \] \hspace{1cm} (5.95)

The system constraints, (5.88) – (5.90), are based on the logic and philosophy of the VRP in that only one station must follow station \( i \) on a given route. These constraints can be achieved by minimizing

\[ P_1 [n(1) + p(1) + n(2) + p(2) + p(3)] \]

where \( n(.) \) and \( p(.) \) indicate the vectors of the underachievement and overachievement for the set of system constraints and \( P_1 \) indicates the first goal priority. The goal constraints, (5.91) – (5.95), can be achieved through the minimization of \( P_2[p(4)] \), \( P_3[p(5)] \), \( P_4[p(6)] \), \( P_5[n(7) + p(7)] \), and \( P_6[n(8) + p(8)] \), respectively.

The trade-offs among the goals (2) – (5) can be easily made. For instance, if the goal of route safety stock is more important than the
total cost, total time of each vehicle route, and customer's satisfac­tion, it is necessary to minimize $P_2[p(6)]$ after minimizing $P_1[.]$. In this problem, it is assumed that neither underachievement nor over­achievement of the fifth and sixth goals are desirable. Therefore, the variables $n(1), p(1), n(2), p(2), p(3), n(7), p(7), n(8),$ and $p(8)$ are to be minimized. However, it is assumed that the total cost and total elapsed time must be less than predetermined levels $C$ and $T_1 + T_2$, respectively. Thus, only $p(4)$ and $p(5)$ are to be minimized for these two goals. Since the route safety stock cannot exceed the value of $[(Q - \bar{Q})/N^{-1}(1 - \eta)]^2$, then $p(6)$ is to be minimized.

5.9 Sensitivity of Time and Truck Capacity upon the Probability of Route Failures

In most cases, some of the problem data are not known exactly, but are estimated as accurately as possible. The probability of route failures $\alpha, \beta, \text{and } \eta$ might not exactly be known for the decision maker because travel and unload times, and customer demands are random variables. Therefore, the probability of route failures forces one to analyze the sensitivity of time and truck capacity.

Theorem 5.3

If $\alpha$, the probability of route failure, increases (up to 0.5), the value of $T_1$ will increase provided that $T_1$ is a fixed value.

Proof:

Let us reconsider equation (5.84) in the form of (5.96), where $\phi = Z - Z_{1-\alpha} = N^{-1}(1 - \alpha)$ and $T_1$ is fixed value:
\[ \bar{T}l = (2Z^2 \psi + 4Tl - 2Z\psi^{1/2} (Z^2 \psi + 4Tl)^{1/2})/4 \quad (5.96) \]

\[ \partial \bar{T}l/\partial \alpha = \partial \bar{T}l/\partial Z \cdot \partial Z/\partial \alpha. \quad (5.97) \]

\[ \partial Z/\partial \alpha < 0, \text{ and this is because when } \alpha \text{ increases, then } (1 - \alpha) \text{ and } Z(1 - \alpha) \text{ decrease. However, after some calculations, one has} \]

\[ \partial \bar{T}l/\partial Z = (Z\psi(2^2 \psi + 4Tl)^{1/4} - 3/2Z^2\psi^{3/2} - 2Tl\psi^{1/2})/(Z^2\psi + 4Tl)^{1/2}. \]

Notice that when \( a \) and \( b \) are two positive numbers, the following inequality exists:

\[ (a + b)^{1/2} < (a)^{1/2} + (b)^{1/2}. \quad (5.98) \]

Therefore,

\[ Z\psi(2^2 \psi + 4Tl)^{1/4} < Z\psi(2^2 \psi)^{1/4} + Z\psi(4Tl)^{1/4} = Z^2\psi^{3/2} + 2Z\psi Tl^{1/2}. \quad (5.99) \]

Hence, due to inequality (5.99),

\[ \partial \bar{T}l/\partial Z < (Z^2\psi^{3/2} + 2Z\psi Tl^{1/2} - 3/2 Z^2\psi^{3/2} - 2Tl\psi^{1/2})/(Z^2\psi + 4Tl)^{1/2} \]

\[ = (-1/2Z^2\psi^{3/2} Z^2 + 2Z\psi Tl^{1/2} - 2Tl\psi^{1/2})/(Z^2\psi + 4Tl)^{1/2}. \quad (5.100) \]

The numerator on the right hand side of (5.100) is

\[ - [2^{-1/2} Z^3/4 - (2Tl\psi^{1/2})^{1/2}]^2 \]

\[ \partial \bar{T}l/\partial Z < - [(\frac{1}{\sqrt{2}})Z^{3/4} - \sqrt{2}Tl\psi^{1/2}]^2/(Z^2\psi + 4Tl)^{1/2} < 0. \quad (5.101) \]

Hence

\[ \partial \bar{T}l/\partial \alpha = \partial \bar{T}l/\partial Z \cdot \partial Z/\partial \alpha > 0. \text{ Q.E.D.} \]
Corollary 1 to Theorem 5.3

If $\beta$, the probability of route failure, increases (up to 0.5) then the value of $T_2$ will increase provided that $T_2$ is a fixed value.

Corollary 2 to Theorem 5.3

If $\eta$, the probability of route failure, increases (up to 0.5) then the value of $Q$ will increase provided that $Q$ is a fixed value.

Theorem 5.4

Suppose that $\alpha_k$ and $\beta_k$, the probability of route failures, are set such that $\frac{d\beta_k}{d\alpha_k} > 0$, then by increasing $\alpha_k$ and $\beta_k$ (up to 0.5), the total elapsed time of the $k$th route will decrease.

Proof:

Let $Z_1 = Z(1 - \alpha_k) = N^{-1}(1 - \alpha_k)$ and $Z_2 = Z(1 - \beta_k) = N^{-1}(1 - \beta_k)$.

The total elapsed time of the $k$th route is

$$T_k = \sum_{i \in I} \sum_{j \in I, i \neq j} \mu_{t_{ij}} x_{ijk} + \sum_{i \in I} \sum_{j \in I, i \neq j} \mu_{t_{ij}} x_{ijk} + Z_1(\sum_{i \in I} \sum_{j \in I, i \neq j} \sigma^2_{t_{ij}} x_{ijk}^2)^{\frac{1}{2}}$$

$$+ Z_2(\sum_{i \in I} \sum_{j \in I, i \neq j} \sigma^2_{t_{ij}} x_{ijk}^2)^{\frac{1}{2}}. \quad (5.102)$$

Now, it is necessary to show that $\frac{\partial T_k}{\partial \alpha_k} < 0$ where

$$\frac{\partial T_k}{\partial \alpha_k} = \frac{\partial T_k}{\partial z_1} \cdot \frac{\partial z_1}{\partial \alpha_k} + \frac{\partial T_k}{\partial z_2} \cdot \frac{\partial z_2}{\partial \beta_k} \cdot \frac{\partial \beta_k}{\partial \alpha_k}. \quad (5.103)$$
\[ \frac{\partial Z_1}{\partial \alpha_k} \text{ and } \frac{\partial Z_2}{\partial \beta_k} \text{ are both less than zero because by increasing } \alpha_k \text{ and } \beta_k, \ (1 - \alpha_k) \text{ and } (1 - \beta_k) \text{ decrease and consequently } Z_1 = Z(1 - \alpha_k) \text{ and } Z_2 = Z(1 - \beta_k) \text{ decrease. However,} \]

\[ \frac{\partial T_k}{\partial Z_1} = \left( \sum_{i \in I} \sum_{j \in I} \sigma^2_{t_{ij}} x_{ijk}^2 \right)^{\frac{1}{2}} > 0 \]  
(5.104)

\[ \frac{\partial T_k}{\partial Z_2} = \left( \sum_{i \in I} \sum_{j \in I} \sigma^2_{t_{ij}} x_{ijk}^2 \right)^{\frac{1}{2}} > 0 \]  
(5.105)

\[ \frac{\partial T_k}{\partial \alpha_k} = (+) (-) + (+) (-) (+) = (-) + (-) < 0 \quad \text{Q.E.D.} \]

**Theorem 5.5**

If the condition of Theorem (5.4) exists for all NV truck routes, then the total elapsed time of the whole system decreases.

**Proof:**

According to Theorem (5.4) the total elapsed time of each vehicle route decreases and thus it can be concluded that the total elapsed time of the whole delivery system will decrease since NV remains unchanged.

**Theorem 5.6**

For a SVRP having only probabilistic customer demands, if \( \eta \), the probability of route failure, increases (up to 0.5), then the total travel distance of the whole delivery system will decrease.
Proof:

To prove this theorem, the following, which is a mathematical model for a SVRP having only probabilistic customer demand and no time restrictions, is considered:

\[
\text{Minimize:} \quad D = \sum_{k=1}^{\text{NV}} \sum_{i=0}^{\text{TNS}} \sum_{j=0}^{\text{TNS}} d_{ij} X_{ijk} \quad (5.106)
\]

Subject to:

\[
\sum_{i=1}^{\text{TNS}} \sum_{j=0}^{\text{TNS}} \mu_{d_{ij}} X_{ijk} + N^{-1}(1 - \eta_k) \left( \sum_{i=1}^{\text{TNS}} \sum_{j=0}^{\text{TNS}} \sigma_{d_{ij}}^2 X_{ijk}^2 \right)^{\frac{1}{2}} \leq Q \quad (5.107)
\]

\[X = [X_{ijk}] \in \mathcal{S}. \quad (5.108)\]

By setting

\[
Z = N^{-1}(1 - \eta_k) \quad \text{and} \quad Y = \left( \sum_{i=1}^{\text{TNS}} \sum_{j=0}^{\text{TNS}} \sigma_{d_{ij}}^2 X_{ijk}^2 \right)^{\frac{1}{2}} > 0
\]

it will be noticed that \(\partial Z / \partial \eta < 0\) because by increasing \(\eta_k\), \((1 - \eta_k)\) decreases and consequently \(N^{-1}(1 - \eta_k)\) decreases. Hence, \(Z \cdot Y\) decreases and consequently, \(Q = Q - Z \cdot Y\) will increase. However, \(\partial D / \partial \eta = \partial D / \partial Q \cdot \partial Q / \partial Z \cdot \partial Z / \partial \eta\). It is obvious that \(\partial Q / \partial Z = -Y < 0\). Hence, it remains to show that \(\partial D / \partial Q < 0\), this is because \(\partial Q / \partial Z \cdot \partial Z / \partial \eta > 0\).

Now, it is only necessary to prove that in the deterministic VRP where customer demands are equal to their demand's mean, by increasing the artificial capacity of truck the travelled distance will decrease. If the transportation cost depends linearly on the weight of goods delivered and the distance travelled, then the following equation can be used:
\[ C_{ij} = U_{ij} W_{ij} \, d_{ij} \]  

(5.109)

where

\[ U_{ij} = \text{cost per unit weight per unit distance from node } i \text{ to node } j, \]

\[ W_{ij} = \text{weight transported from node } i \text{ to node } j, \]

\[ d_{ij} = \text{the distance from node } i \text{ to node } j, \]

\[ r_j = \text{number of times that weight } W_{ij} \text{ can be fitted in } \bar{Q}. \]

However,

\[ d_{ij} = \frac{C_{ij}}{U_{ij} W_{ij}} \]  

(5.110)

and

\[ \bar{Q} = r_j \, W_{ij} \]

or

\[ W_{ij} = \frac{\bar{Q}}{r_j} \]

Since \( X_{ij} = \begin{cases} 1 \\ 0 \end{cases} \), then

\[ D = \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} d_{ij} X_{ijk} = \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} d_{ij} \]

\[ \sum_{i \neq j} \]

or

\[ D = \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \frac{C_{ij} \, r_j}{U_{ij} \, \bar{Q}}. \]

Hence,

\[ \frac{\partial D}{\partial \bar{Q}} = \frac{-1}{(\bar{Q})^2} \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \frac{C_{ij} \, r_j}{U_{ij}} < 0. \]

This result indicates that \( \frac{\partial D}{\partial \bar{Q}} < 0 \), which proves this theorem. Q.E.D.
The following theorem is concerned with the number of constructed vehicle routes for the SVRP having probabilistic customer demands. It indicates that when number of constructed vehicle routes increases, then the total demand to be served by the vehicles will increase. However, this situation will not happen when customer demands are deterministic.

**Theorem 5.7**

The larger number of routes is equivalent to the larger total demand to be served by all vehicles.

**Proof:**

Suppose that $\mu_i$ and $\sigma_i^2$ are the mean and variance of demand point $i$. It is clear that

$$\sum_{i=1}^{\text{TNS}} \mu_i = \mu_1 + \mu_2 + \ldots + \mu_{\text{TNS}}. \quad (5.112)$$

But

$$\sum_{i=1}^{\text{TNS}} \frac{2}{\sigma_i^2} < \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} + \ldots + \frac{1}{\sigma_{\text{TNS}}^2}. \quad (5.113)$$

If only one vehicle can be used to deliver all customer demands, then the following inequality is needed:

$$\sum_{i=1}^{\text{TNS}} \mu_i + \frac{1}{N-1}(1 - \eta) \sum_{i=1}^{\text{TNS}} \frac{2}{\sigma_i^2} \leq Q. \quad (5.114)$$

If two vehicles are used instead of one vehicle to deliver the customer demands, the following inequalities are needed;

$$\sum_{i=1}^{\text{m<TNS}} \mu_i + \frac{1}{N-1}(1 - \eta) \sum_{i=1}^{\text{m<TNS}} \frac{2}{\sigma_i^2} \leq Q. \quad (5.115)$$
and

\[
\sum_{i=m+1}^{TNS} \mu_i + N^{-1}(1 - \eta) \left( \sum_{i=m+1}^{TNS} \sigma_i^2 \right)^{\frac{1}{2}} \leq Q. \tag{5.116}
\]

Hence, after the addition of both sides of inequalities (5.115) and (5.116), the result is the following inequality:

\[
\sum_{i=1}^{TNS} \mu_i + N^{-1}(1 - \eta) \left[ (\sum_{i=1}^{m<TNS} \sigma_i^2)^{\frac{1}{2}} + (\sum_{i=m+1}^{TNS} \sigma_i^2)^{\frac{1}{2}} \right] \leq 2Q \tag{5.117}
\]

The left-hand side of inequality (5.117) represents the total generated demands to be served by two vehicles. Using the concept of inequality (5.113), inequality (5.117) can be written in the following form

\[
\sum_{i=1}^{TNS} \mu_i + N^{-1}(1 - \eta) \left( \sum_{i=1}^{TNS} \sigma_i^2 \right)^{\frac{1}{2}} \leq \sum_{i=1}^{m} \mu_i + N^{-1}(1 - \eta). \tag{5.118}
\]

\[
[ (\sum_{i=1}^{m} \sigma_i^2)^{\frac{1}{2}} + (\sum_{i=m+1}^{TNS} \sigma_i^2)^{\frac{1}{2}} ] < 2Q.
\]

The inequality (5.118) indicates that the total generated demand using two vehicles is larger than using one vehicle. However, one can extend the previous discussion for NV vehicles which are needed to satisfy all customer demands.

5.10 Summary

This chapter has presented the development of a multiple objective goal programming formulation of SVRP in which customer demand and travel and unload times are considered to be random variables having known distribution functions. The mathematical formulations of the problem were directly related to the model developments for the RCS and RIS of the
problem. The equivalent deterministic forms of Problems "C" and "D" were developed and presented by "E" and "F" type problems, respectively.

The existence of the optimum solution for the RCS of the problem is shown through Theory (5.1). The existence of a set of deterministic linear time constraints, which are equivalent to the nonlinear set of time constraints of the problem for distributions such as poisson and chi-square, is proved through Theory (5.2). The effects of route failure probabilities on the total elapsed time of the whole delivery system were proved through Theories (5.3) – (5.5). Theory (5.6) illustrates that the total travelled distance decreases when route failure probability increases. Additionally, Theory (5.7) is provided to demonstrate that the larger number of vehicle routes is equivalent to the larger customer demands.

The remainder of this research is divided into five chapters. Chapter VI is devoted to the development of the LIGP technique. Chapter VII demonstrates the development of an appropriate heuristic algorithm for obtaining favorable vehicle routes for "E" and "F" type problems. These heuristic approaches are new modifications of Clarke and Wright's algorithm. Chapter IX discusses the details of two interactive computer programs for the SVRP and LIGP techniques. Chapter X gives the conclusions and recommendations for future research in the area of SVRP.
CHAPTER VI
LINEAR INTEGER GOAL PROGRAMMING TECHNIQUE

6.1 Introduction

Linear Programming (LP) is a well known mathematical technique for optimizing a single objective function such as profit, total elapsed time, or total travelled distance, subject to stated constraints. The LP technique is employed in decision making situations in many real world problems. Due to the existence of conflicting objectives in many decision making situations, the area of multiple objective decision making has received a great deal of attention in recent years. In such decision making situations, the overall desire is that all objectives or goals be simultaneously met to as large an extent as possible.

One well known procedure which treats this problem is the Goal Programming (GP) technique. The GP programming technique assumes that the variables take continuous values within the feasible region. A Linear Integer Goal Programming (LIGP) problem is a goal programming problem in which the constraints and objective functions are linear, but the variables in the final solution are required to be integers.

Two distinct GP techniques, Preemptive Goal Programming (PREGP) and Partitioning Goal Programming (PARGP), are employed as a basis of algorithm routings. The PREGP and PARGP procedures are based on the simplex method of LP and Arthur and Ravindran's technique for solving linear goal programming problems. Arthur and Ravindran [4] have devised
a PARGP technique which consists of solving a series of linear programming subproblems with the solution to the higher priorities used as the initial solution of the lower priority problem. The major advantage of PARGP, relative to PREGP, is that one deals with the fewer constraints, fewer variables, and only one objective function at each stage of the problem.

The methodology for solving linear goal programs cannot be used to solve the linear integer goal program. Therefore, the PREGP and PARGP are extended to accomplish the handling of integer variables. However, two approaches are used for the integer algorithm developments of goal programs. The first approach deals with the development of the "cutting planes" (new objectives) that are added to the problem formulation when the continuous solution of the original problem has been obtained by the algorithm. The second approach concerns the development of the branch and bound algorithm for linear integer goal programs. However, the developed linear integer goal programs for GP techniques of PREGP and PARGP are called LIPREGP and LIPARGP, respectively.

6.2 Cutting Plane Method for Integer Goal Programming

The Cutting Plane (CP) method is a procedure which is used in literature [42] to solve the integer GP problems. The CP algorithms were originally developed in 1958 by Ralph Gomory [40, 56] for general Integer Linear Programming (ILP). The main difference between the Gomory procedure for Linear Integer Programming (LIP) and the CP of goal programming is the method in which these procedures handle the multidimensional priority weights.
The following sections are devoted to the development of the CP method for the All-Integer and Mixed-Integer LGP problems. The All-Integer, where all variables are required to be integer valued, will be discussed first, followed by a discussion of the Mixed-Integer case.

6.2.1 Development of All-Integer Cut

Suppose that the \( i \)th constraint is the selected source row and it appears in the final tableau of the related GP as

\[
\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = \text{source row}) \tag{6.1}
\]

where for simplicity \( x_j \) denotes any variable, whether decision or deviation; \( a_{ij} \) is the coefficient of variable \( x_j \) in the source row, and \( b_i \) is the right-hand-side value of the source row. Indicate the integer part of \( a \) as \([a]\), then, since \([a_{ij}] \leq a_{ij} \) and \( x_j \geq 0 \), one can write

\[
\sum_{j=1}^{n} [a_{ij}] x_j \leq b_i. \tag{6.2}
\]

Any integer vector \( x \) which satisfies (6.1) will also satisfy (6.2). For such an \( x \), the left-hand side of (6.2) is an integer. Hence, integer vector \( x \) must also satisfy the following constraint

\[
\sum_{j=1}^{n} [a_{ij}] x_j \leq [b_i] \tag{6.3}
\]

On the assumption that \( a_{ij} \) and \( b_i \) are not integer valued, one can write

\[
[a_{ij}] + f_{ij} = a_{ij} \tag{6.4}
\]
and
\[ [b_i] + f_i = b_i \] \hfill (6.5)

where \( 0 \leq f_{ij} < 1 \) and \( 0 \leq f_i < 1 \).

After substituting (6.4) and (6.5) in (6.3), then
\[ \sum_{j=1}^{n} (-f_{ij}) x_j \leq -f_i \] \hfill (6.6)

Inequality (6.6) can further be changed into
\[ \sum_{j=1}^{n} (-f_{ij}) x_j + n_i - p_i = -f_i \] \hfill (6.7)

Equation (6.7) is the cutting plane constraint to be added to the final tableau of the GP problem with the noninteger variables. In order to take care of the infeasibility resulting from the addition of constraint (6.7) into the final tableau of the GP problem, the dual simplex method may be employed to solve the new problem. Alternatively, the cutting plane (6.7) can be arranged as in (6.8):
\[ \sum_{j=1}^{n} f_{ij} x_i + n_i - p_i = f_i. \] \hfill (6.8)

In this case, the regular primal procedure is utilized.

In order to satisfy the cutting plane constraints (6.7) and (6.8), \( p_i \) and \( n_i \) should be minimized at the first priority level, respectively, and all other priorities are downgraded one level lower than their original assignment.
6.2.2 Development of the Mixed-Integer Cut

A Mixed-Integer Linear Goal Programming (MILGP) problem [40] can be developed in a manner similar to the pure IGP problem as previously described. In this case, only certain variables are to be integer-valued. The remaining ones take on feasible values on the continuous scale. Suppose that \( x_j \) is a variable which is required to be integral, then the \( i^{th} \) (source row) constraint can be written as

\[
x_j + \sum_{j \in \text{nonbasic}} f'_{ij} x_j = b_i
\]

(6.9)

By considering

\[
b_i = [b_i] + f_i
\]

(6.10)

then

\[
x_j + \sum_{j \in \text{nonbasic}} f'_{ij} x_j = [b_i] + f_i
\]

(6.11)

or

\[
\sum_{j \in \text{nonbasic}} f'_{ij} x_j + n_i - p_i = ([b_i] - x_j) + f_i
\]

(6.12)

where, the index \( i \) indicates the source row, \( f_i \) is the fractional part of the right-hand side value of the source row, and \( f'_{ij} \) is defined as follows:

\[
f'_{ij} = \begin{cases} a_{ij} & \text{if } a_{ij} \geq 0 \text{ and } x_j \text{ is a continuous variable} \\ (f_i/(f_i-1))a_{ij} & \text{if } a_{ij} < 0 \text{ and } x_j \text{ is a continuous variable} \\ f_i & \text{if } f_i < f_i \text{ and } x_j \text{ is an integer variable} \\ (f_i/(1-f_i))(1-f_{ij}) & \text{if } f_i > f_i \text{ and } x_j \text{ is an integer variable} \end{cases}
\]

(6.13)
Note that $f_{ij}$ is the nonnegative fractional part of $a_{ij}$. When the
cutting plane method (6.12) is chosen to be used, then $p_i$ is to be mini-
mized at the first priority level. All other priorities are to be down-
graded one level lower than their original assignment. However, using
either technique, the process of adding cutting plane methods and solv-
ing the new problem is repeated until an integer solution is reached.
This process is described in more detail below.

6.3 Algorithm for LIGP using
Cutting Plane Method

The proposed algorithm can be summarized as follows:

Step 1: Solve the initial GP problem by dropping the integerality
requirements. If the solution to this problem is integer, stop. Other-
wise, go to Step 2.

Step 2: Generate the cutting plane constraint as shown in (6.8)
or (6.12), depending on the type of problem (pure or mixed integer). A
most promising technique for choosing the source row is to choose the
constraint in the final simplex tableau which gives the largest $f_i$.

Step 3: Solve the new problem with the augmented cutting plane.
Use the regular method of the GP procedure. If the solution to this
problem is integer, stop. Otherwise, go to step 2.
Example 1

The following problem was taken from A. A. Abduelmagd [1] for illustration of this procedure. This problem is solved by the preemptive LIGP procedure using the cutting plane method where an integer solution to variables $x_1, x_2, x_3, x_4, x_5,$ and $x_6$ is required.

Minimize $P_1(n_1 + p_1 + n_2 + p_2) + P_2(n_3) + P_3(n_4)$

Subject to:

\[
\begin{align*}
8x_1 + x_2 + 3x_3 + 2x_4 + 3x_5 - 3x_6 + n_1 - p_1 & = 17 \\
3x_1 + 2x_3 + x_4 + x_5 - x_6 + n_2 - p_2 & = 5 \\
5x_1 + x_3 + 2x_4 + x_5 - 4x_6 + n_3 - p_3 & = 8 \\
12x_1 + x_2 + 2x_3 + 5x_4 + 4x_5 - 6x_6 + n_4 - p_4 & = 30 \\
\end{align*}
\]

$x_1 \geq 0$, $n_i$, $p_i \geq 0$ and $x_i$ are integer.

A continuous solution to this problem is obtained after five simplex iterations have been performed. The solution is

$x_1 = 0.40, x_2 = 7.0, x_4 = 4.60, x_6 = 0.80$

where the remainder of variables are zero and all three priorities have been achieved. Since only six variables out of 14 variables (number of decision variables plus deviations) were required to be integer, the mixed integer procedure was employed to obtain an optimal integer solution for this problem. After 23 more iterations and seven cuts, the following integer solution was obtained:

$x_1 = 0.0, x_2 = 6.0, x_3 = 0.0, x_4 = 4.0, x_5 = 1.0, x_6 = 0.0$

where all priority levels were achieved.
6.4 Branch and Bound Method for Integer Goal Programming

The technique of Branch and Bound was originally introduced by Land and Doing [40, 56]. Due to the inefficiency of Branch and Bound for computer coding, a modification of the algorithm was developed by Dakin [40, 44, 56]. This technique, unlike the CP methods, can be applied directly to both the pure and mixed integer LGP problems. In order to apply the Dakin algorithm to a LGP, one starts to solve the problem by a general LGP with the integer requirements ignored. If the result of this GP happens to be an integer solution according to the original integer requirements of the problem, then the optimal solution has been achieved. If the optimal solution is not an integer solution, then a noninteger variable should be selected from the list of the required integer variables. After such a variable, \( x_j \), is selected, one can write a range of the following form for that variable:

\[
[b_j] \leq x_j \leq [b_j] + 1
\]

(6.14)

where \([b_j]\) represents the largest integer that is less than the value of \(b_j\). Since \(x_j\) is required to be integer, the given range by (6.14) is infeasible for this variable. However, to avoid any solution in this range, the following conditions can be utilized as two objectives

\[
x_j \leq [b_j]
\]

\[
x_j \geq [b_j] + 1
\]

Each of these objectives is a presentation of a new problem which is branched from the previous problem. The GP problem which is associated
with each new branch consists of the GP problem of the previous problem from which this branch emanates and one of these two new objectives. The Branch and Bound method can be summarized in the following five steps:

Step 1 (Initial Solution): Solve the problem by a general method of LGP by treating all variables (decision variables and deviations) as continuous. Check the optimal solution, if this solution satisfies the integer requirements of the problem, then the optimal solution has been obtained, otherwise, go to Step 2.

Step 2 (Branching Variable Selection): Select a variable from the set of variables which are constrainted to be integer and its solution value is not integer. Using this variable, develop two new objectives as follows:

\[ x_j \leq \lfloor b_i \rfloor \quad (6.15) \]

and

\[ x_j \geq \lfloor b_i \rfloor + 1 \quad (6.16) \]

where \( x_j \) is the basic variable located in the \( i \)th row and \( b_i \) is the right-hand-side value of this row (or the solution value of \( x_j \)).

The objectives of (6.15) and (6.16) can be written in terms of nonbasic variables of the optimal tableau from which \( x_j \) was chosen. More clearly, objective (6.15) can be written as

\[ x_j = b_i - \sum_{j=1}^{n} a_{ij} x_j \leq \lfloor b_i \rfloor \quad (6.17) \]
or
\[ b_i - [b_i] \leq \sum_{j=1}^{n} a_{ij} x_j \]  
(6.18)

By setting \( f_i = b_i - [b_i] \), (6.18) can be written as
\[ f_i \leq \sum_{j=1}^{n} a_{ij} x_j \]

Therefore,
\[ \sum_{j=1}^{n} a_{ij} x_j + n_i - P_i = f_i \]  
(6.19)

where \( n_i \) is to be minimized at the first priority level. Similarly, the objective \( x_j \geq [b_i] + 1 \) can be written as
\[ \sum_{j=1}^{n} (-a_{ij}) x_j + n_i - P_i = (1 - f_i) \]  
(6.20)

where \( n_i \) should be minimized again at the priority level one. Furthermore, it should be noted that each of the equations (6.19) and (6.20) will be treated separately as a new constraint and a new objective function in the goal programming formulation of new problems which emanate from the previous problem.

Step 3 (Formation of New Nodes): Add these new constraints to the goal programming problem by the node under consideration in Step 2. One subproblem is formed by augmenting constraint (6.19) and the other by
augmenting constraint (6.20). Solve each of these subproblems as a linear goal programming using the simplex method to obtain two new solutions to these problems. Determine the degree of goal attainments (the optimal value of the objective function) for each subproblem separately.

Step 4 (Test for Terminal Node): Each of the nodes formed in Step 3 may be a terminal node for one of the following reasons:

1. The problem represented by the node may have no feasible solution.

2. The value of \( x_j, j \in I \) are all integers (\( I \) indicates the set of all required integer variables).

In both cases, the node under consideration should be terminated. In the second case, the value of the objective function should be compared with current best available value. By defining vector \( \overline{R}_i = (r_1, r_2, \ldots) \) as the value of priority levels at node \( i \), then for any two solutions, say \( \overline{R}_k \) and \( \overline{R}_m \), \( \overline{R}_k \) is preferred to \( \overline{R}_m \) if a priority level of \( \overline{R}_k \) is lower in value than the corresponding priority level in \( \overline{R}_m \) and all preceding priority levels are equal in both \( \overline{R}_k \) and \( \overline{R}_m \) [38, pp. 130].

The priority level \( \overline{R}_k \) is preferred to priority level \( \overline{R}_r \) if \( \overline{R}_k = (0, 100, 16, 300, 0) \) and \( \overline{R}_r = (0, 100, 19, 401, 5) \).

Step 5 (Node Selection): If both nodes at Step 4 were terminated then select the next node from the list of nodes which are in the waiting list for further branching. If exactly one node in Step 4 was terminated, then use the nonterminal node and go to Step 2.

If both nodes in Step 4 were nonterminal, then choose the more promising one. A node with the smallest value of the objective function is considered to be more promising. However, the other node should be
added to the list of waiting nodes for further branching which are considered later. Figure 1 depicts the branch and bound procedure for LIGP.

The selection of variables from which to generate new constraints is obviously one of the most important steps to be taken in the solution process of LIGP problem. The $i^{th}$ constraint where the basic variable $x_j$ assumes a noninteger value is considered to be the source row. The easiest way of selecting the source row is to pick up a basic variable with the largest fractional part. It is important to note that this rule is not an absolute one but it works well.

Example 2

Let us reconsider the problem which is given in example 1. Now an integer solution to this problem by the LIPARGP technique along with the branch and bound procedure is required.

A continuous solution to this problem was obtained after 5 iterations which is given below:

$$x_1 = 0.40 \quad x_2 = 7.0 \quad x_4 = 4.6 \quad x_6 = 0.80$$

where all priorities are achieved and the remainder of variables are zero. The integer procedure has started on the sixth iteration and has stopped on the sixteenth iteration with the result given in Figure 2. Hence, the optimal integer solution to this problem by branch and bound technique is $x_1 = 0, x_2 = 7, x_3 = 0, x_4 = 5, x_5 = 1, x_6 = 1$ where priorities 1 and 3 have been achieved and the underachievement of priority 2 is equal to 1.
Solve the original problem

Are the integer requirements satisfied?

Select variable \( x_j (j \in I) \) whose value is noninteger

Construct two new subproblems: one with \( x_j \leq [b_j] \), the other with \( x_j \geq [b_j] + 1 \) added to the original problem

Select the subproblem with the lexicographically lowest value \( R_t \) for further branching. \( R_t \) is the lowest value of the priority levels for subproblem \( t \).

Figure 1. Flowchart for the Branch and Bound Procedure for LIGP
Figure 2. A Tree Diagram Presentation of the Solution of Example 2
Example 3

The problem which is presented in Table I [44] is a 10 variable problem with 15 constraints and 6 priority levels. Therefore, the total number of variables (decision and deviation) are 40. First, an attempt was made to find only an integer solution to this problem using the branch and bound method and LIPREGP. Therefore, an integer solution to this problem after 11.10 seconds and 115 iterations was obtained. The solution is

$$X_1 = 2, X_2 = 2, X_3 = 1, X_4 = 0, X_5 = 0,$$
$$X_6 = 1, X_7 = 0, X_8 = 0, X_9 = 1, X_{10} = 0$$

where all priorities have been achieved at the levels of

$$r_1 = r_2 = r_3 = r_4 = r_5 = 0$$ and $$r_6 = 320.54.$$

Next, an attempt was made for the evaluation of 0-1 integer solution to the above problem. Therefore, constraints of the following type:

$$x_i + n_i - p_i = 1, \forall i = 1,2,\ldots,10$$ and an absolute priority level

$$P_0 = \sum_{i=1}^{10} p_{i+m}$$ were added into the original problem. Now this problem is composed of 60 variables, 25 constraints, and 7 priorities. The absolute priority level was considered as the first priority and the original priorities of the problem were downgraded by one level. This problem was solved by the LIPREGP using branch and bound method. Three integer solutions of the following were obtained after 68 iterations and 11.37 seconds. The first set of 0-1 integer solutions are:

(I) $$X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 0, X_5 = 1,$$
$$X_6 = 1, X_7 = 0, X_8 = 0, X_9 = 1, X_{10} = 0$$
TABLE I
A 10-VARIABLE TEST PROBLEM FOR EXAMPLE 3*

Minimize: 
\[ Z = P_1(n_{14} + n_{15}) + P_2(p_1 + p_2 + p_3) + P_3n_4 + \]
\[ P_4(n_5 + n_6 + n_7) + P_5(n_8 + n_9 + n_{10}) + \]
\[ P_6(p_{11} + p_{12} + p_{13}) \]

Subject to:

<table>
<thead>
<tr>
<th></th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>X_4</th>
<th>X_5</th>
<th>X_6</th>
<th>X_7</th>
<th>X_8</th>
<th>X_9</th>
<th>X_10</th>
<th>b_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>20</td>
<td>100</td>
<td>75</td>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>250</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td></td>
<td>150</td>
<td>90</td>
<td>100</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td>259</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>315</td>
<td>173</td>
<td>160</td>
<td>680</td>
<td>475</td>
<td>95</td>
<td>65</td>
<td>120</td>
<td>115</td>
<td>300</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>75</td>
<td>20</td>
<td>250</td>
<td>10</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>275</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>150</td>
<td>25</td>
<td>170</td>
<td>250</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>150</td>
<td>100</td>
<td>115</td>
<td>250</td>
<td>50</td>
<td>70</td>
<td>25</td>
<td>50</td>
<td>100</td>
<td>590</td>
</tr>
<tr>
<td>8</td>
<td>46</td>
<td>56</td>
<td>15</td>
<td></td>
<td>185</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>187</td>
</tr>
<tr>
<td>9</td>
<td>74</td>
<td>111</td>
<td>16</td>
<td>126</td>
<td>199</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>242</td>
</tr>
<tr>
<td>10</td>
<td>75</td>
<td>111</td>
<td>74</td>
<td>79</td>
<td>186</td>
<td>32</td>
<td>27</td>
<td>14</td>
<td>29</td>
<td>74</td>
<td>267</td>
</tr>
<tr>
<td>11</td>
<td>31</td>
<td>43</td>
<td>29</td>
<td>37</td>
<td>21</td>
<td>17</td>
<td>6</td>
<td>33</td>
<td>18</td>
<td>9</td>
<td>81</td>
</tr>
<tr>
<td>12</td>
<td>33</td>
<td>41</td>
<td>29</td>
<td>41</td>
<td>27</td>
<td>17</td>
<td>6</td>
<td>31</td>
<td>19</td>
<td>27</td>
<td>84</td>
</tr>
<tr>
<td>13</td>
<td>35</td>
<td>40</td>
<td>31</td>
<td>42</td>
<td>27</td>
<td>24</td>
<td>13</td>
<td>21</td>
<td>19</td>
<td>41</td>
<td>84</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

*The data of this test problem is taken from Lewis [44, p. 83].
where priorities 1 through 6 have been achieved at the level of 0, 0, 0, 125, 0, and 163.2, respectively. The second set of 0-1 integer solution is

\[(II) \quad x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0,
\quad x_6 = 1, x_7 = 1, x_8 = 1, x_9 = 0, x_{10} = 1\]

where priorities 1 through 6 have been achieved at the level of 0, 95, 309, 45, 78 and 343, respectively. Finally, the third set of 0-1 integer solution is

\[(III) \quad x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0, x_5 = 1,
\quad x_6 = 1, x_7 = 0, x_8 = 0, x_9 = 1, x_{10} = 0\]

where priorities 1 through 6 have been achieved at the level of 0, 83, 0, 105, 0, and 254, respectively.

Obviously, the optimal integer solution to this problem is the first set of integer solutions using the method of preference as previously described. Lewis [44, p. 82] solved this problem by the zero-one goal programming code. The same optimal integer solution was obtained by the 0-1 GP code after 12 seconds and 490 solution combinations.

6.5 Summary

The purpose of this chapter was to develop the linear integer goal programming techniques which could efficiently be used in the goal oriented problems. Two goal programming techniques, PREGP and PARGP, were used as the basis of the algorithm routings. The Cutting Plane method and Branch and Bound techniques were employed for solving the integer goal programming problems. Two cases of all integer variables and mixed
integer variables were discussed for the Cutting Plane methods. Similar situations were investigated for the IGP based on the Branch and Bound procedure. The applicability of IGP methods were demonstrated through the solution of three example problems.
CHAPTER VII

DEVELOPMENT OF THE HEURISTIC ALGORITHM
FOR STOCHASTIC VRP

7.1 Introduction

This chapter is concerned with the description and evaluation of the appropriate solution procedures for the "E" and "F" type problems. The proposed approaches for solving the "E" and "F" type problems are based on the Clarke and Wright "saving" approach to construct feasible vehicle routes which in turn satisfy the probabilistic customer demands at each station and the probabilistic travel and unload times constraints.

7.2 Clarke and Wright Algorithm

The Clarke and Wright algorithm [17] (saving approach) is the most widely known of the heuristics developed for solving delivery problems. In the saving approach, it is assumed that every two distinct demand points i and j are supplied individually by two vehicles (Figure 3).

Figure 3. Initial Set-up
Figure 3 illustrates an initial set-up where one vehicle is assigned to one demand point. However, if instead of two vehicles, one uses only one vehicle (Figure 4), the saving in travelled time (cost or distance) is:

\[ S_{ij} = 2T_{0i} + 2T_{0j} - (T_{0i} + T_{0j} + T_{ij}) = T_{0i} + T_{0j} - T_{ij} \]  (7.1).

![Diagram of NODES i and j linked by using the Savings Approach concept.]

The user calculates the "savings" associated with all pairs of locations to be serviced and then sorts these savings in decreasing order beginning with the list of pairs with positive values. Starting at the top of the list, the demand points are combined provided that the resulting tour is feasible and truck capacity is not violated. Using this method, increasingly larger and better tours are formed until the list of savings is exhausted. The chief deficiency of this method is that once an arc is added to a route, it is never removed.

7.3 Development of the Heuristic Approach

For "E" Type Problem

The "E" type problem, as developed in Chapter V, is reproduced here.
Minimize \[ \sum_{k=1}^{NV} \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} c_{ij} x_{ijk} \]  

Subject to:

\[ \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \mu_{c_{ij}} x_{ijk} + N^{-1}(1 - \alpha_k) \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{c_{ij}}^2 x_{ijk}^{1.5} \leq TR_k \]  

\[ \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{d_{ij}} x_{ijk} + N^{-1}(1 - \beta_k) \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma_{d_{ij}}^2 x_{ijk}^{1.5} \leq UT_k \]  

\[ \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_{e_{ij}} x_{ijk} + N^{-1}(1 - \eta_k) \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma_{e_{ij}}^2 x_{ijk}^{1.5} \leq Q \]  

\[ X = [X_{ijk}] \in S \quad \text{for } k = 1, 2, \ldots, NV \]

To adapt the Clarke and Wright algorithm, the following rule previously described in Section 7.2 must be employed,

\[ S_{ij} = C_{0i} + C_{0j} - C_{ij} \]

where \( C_{ij} \) is the cost of moving from station \( i \) to station \( j \). In this problem, constraints (7.3), (7.4), and (7.5) should be checked for feasibility before the addition of any new node to an existing route or before combining two routes together. However, the procedure for solving the "E" type problem is completely identical to the Clarke and Wright algorithm with the exception that some additional checks must be made for new constraints. The "E" type problem algorithm uses the objective function \[ \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sum_{k=1}^{NV} C_{ij} X_{ijk} \] and nonlinear constraints (7.3), (7.4), and (7.5) in contrast to the deterministic form of the Clarke and
Wright algorithm where only linear constraints must be checked for feasibility. These evaluations make the procedure more complex, and consequently more memory allocation and computer time will be required.

The calculations in all three constraints, (7.3), (7.4), and (7.5), are carried out by using values \( \tau_1 = N^{-1}(1 - \alpha) \), \( \tau_2 = N^{-1}(1 - \beta) \), and \( \tau_3 = N^{-1}(1 - \eta) \) rather than using \( \alpha \), \( \beta \), and \( \eta \), respectively. The evaluation of \( \tau_1 \), \( \tau_2 \), and \( \tau_3 \) is not a very hard job. For instance, for normal distribution which has been assumed in equations (7.3), (7.4), and (7.5), \( \tau_1 \), \( \tau_2 \), and \( \tau_3 \) will be the standard normal deviate \( z \).

The flowchart shown in Figure 5 outlines the procedural steps for the method developed for the "E" type problem.

7.4 Development of Heuristic Approach

For "F" Type Problem

The "F" type problem, as shown in Chapter V, is reproduced here.

\[
\begin{align*}
\text{Minimize} & \quad \sum_{k=1}^{NV} \left( \sum_{i=0}^{TNS} \mu_{tij} X_{ijk} + \sum_{i=1}^{TNS} \mu_{t1j} X_{ijk} + N^{-1}(1 - \alpha_k) \right) \\
\text{Subject to:} & \quad \sum_{i=0}^{TNS} \sum_{j=0}^{TNS} \sigma_{t1j}^2 X_{ijk}^{2} \cdot N^{-1}(1 - \beta_k) \left( \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma_{t1i}^2 X_{ijk}^{2} \cdot \frac{1}{2} \right) \\
& \quad \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \mu_d X_{ijk} + N^{-1}(1 - \eta_k) \left( \sum_{i=1}^{TNS} \sum_{j=0}^{TNS} \sigma_d X_{ijk}^{2} \cdot \frac{1}{2} \right) \leq Q \\
& \quad X = [X_{ijk}] \in S, \quad k = 1, \ldots, NV
\end{align*}
\]

As mentioned before, it is almost impossible to solve a large scale problem such as the "F" type problem by the exact procedure. Therefore,
Enter number of Nodes and terminal, distance matrix, travel time, variance of travel time, mean unload time, variance of unload time, mean and variance of demand, capacity of truck, and TRk and UTk.

Calculate the saving in terms of costs for all pairs of nodes.

Order the savings from the largest to smallest.

If all pairs of savings are not being used, drop the previous best pair and consider the next best pair.

Is this link feasible?

Can a link be added in front or back of route?

Can two routes be joined?

Create new route.

Add the node into the route.

Is upper bound of total unload time exceeded?

Is upper bound of total travel time exceeded?

Is capacity exceeded?

Figure 5. Algorithmic Flowchart of Procedural Steps for "E" Type Problem
heuristic approach is considered for solving this problem. In the "F" type problem the objective is to minimize total elapsed, travel, and unload times where these times are random variables. Two modifications of the Clarke and Wright algorithm as they pertain to the "F" type problem, are shown in Sections 7.4.1 and 7.4.2.

7.4.1 Algorithm (I)

To adapt the Clarke and Wright algorithm to handle the random travel and unload times when the minimization of the total elapsed time of the whole delivery system is the criterion, the saving function must be modified. A saving function of the following form can be used for evaluation of savings when travel time is random variable having a known distribution function:

\[ S_{ij} = \gamma (\mu_{t_{0i}} + \mu_{t_{0j}} - \mu_{t_{ij}}) + (1 - \gamma)(\sigma_{t_{0i}}^2 + \sigma_{t_{0j}}^2 + \sigma_{t_{ij}}^2)^{1/2} \]  

provided that \( \mu_{t_{0i}} + \mu_{t_{0j}} - \mu_{t_{ij}} > 0 \) and \( 0 < \gamma \leq 1 \). The saving function shown in (7.10) can be developed very easily. Consider Figures 3 and 4 which have been used for the Clarke and Wright saving algorithm: by considering the fact that the travel time between demand points \( i \) and \( j \) are random variables with means of \( \mu_{t_{0i}} ', \mu_{t_{0j}} ', \) and \( \mu_{t_{ij}} ' \) and variances \( \sigma_{t_{0i}}'^2, \sigma_{t_{0j}}'^2, \) and \( \sigma_{t_{ij}}'^2 \). Also, if one vehicle is used instead of using two, the saving in terms of these random variables is \( S_{T} = t_{0i} + t_{0j} - t_{ij} \), where \( t_{0i}, t_{0j}, \) and \( t_{ij} \) are assumed to be independent random variables. Therefore,

\[ E(S_T) = E(t_{0i} + t_{0j} - t_{ij}) = E(t_{0i}) + E(t_{0j}) - E(t_{ij}) \]

or
\[ E(ST) = \mu_{t_{0i}} + \mu_{t_{0j}} - \mu_{t_{ij}} \]  
\( (7.11) \)

and

\[ \text{Var}(ST) = \sigma^2_{t_{0i}} + \sigma^2_{t_{0j}} + \sigma^2_{t_{ij}} \]  
\( (7.12) \)

The total saving in terms of mean and standard deviation of random variable \( ST \) can be related in one expression with more emphasis in mean of saving than on the standard deviation of saving, as shown by Equation (7.10). In this equation, if \( \gamma = 1 \), then all emphasis is placed on the mean of saving \( ST \) which involves the basic concept of Clarke and Wright's algorithm. On the other hand, if \( 0 < \gamma < 1 \), then a combination of mean and standard deviation of saving \( ST \) will be used.

7.4.2 Algorithm (II)

To adapt the Clarke and Wright algorithm, the following saving function can be used:

\[ F_{\text{saving}} = \mu_{\text{saving}} + \left(\sigma^2/\delta \ast (\sigma^2_{\text{saving}})^{1/2}\right) \]  
\( (7.13) \)

where \( \mu_{\text{saving}} = E(ST) \), \( \sigma^2_{\text{saving}} = \text{var}(ST) \), \( M \) is the total number of \( \sigma^2_{t_{ij}} \), \( \delta > 0 \) and

\[ \overline{\sigma^2} = (\Sigma \sigma^2_{t_{ij}}/M). \]  
\( (7.14) \)

Maximizing function (7.13) provides a station which can be added to the vehicle route. When \( \delta \) approaches to infinitive, then \( F_{\text{saving}} = \mu_{\text{saving}} \), which is the basic concept of the Clarke and Wright algorithm. On the other hand, when \( \delta \to 0 \), then a great emphasis is placed on the standard deviation rather than on the mean. Equation (7.14) indicates
that \( \sigma^2 \) is a constant value for each specific problem. Hence, \( F_{\text{saving}} \) for each pair of demand points depends upon the value of saving in mean and on the amount of standard deviation between these two stations.

Since great emphasis will more often be placed on the mean of saving rather than on the variance of saving, one can logically design a larger coefficient for saving in mean than on the variance. However, one may employ both algorithms to solve a SVRP and then accept the solution with the lowest total travel time for the whole system. The flowchart shown in Figure 6 outlines the procedural steps for the method of solution for the "F" type problem.

To compare saving function (7.10) and (7.13), consider the following new notations for simplicity: \( x = \mu_{\text{saving}} \), \( y = (\sigma_{\text{saving}})^2 \), \( z = S_{ij} \), \( c = \sigma^2 = \text{constant} \) and \( z' = F_{\text{saving}} \). Hence, the saving functions (7.10) and (7.13) can be written in terms of new notations as:

\[
\begin{align*}
z &= \gamma x + (1 - \gamma)y, \\
z' &= x + \frac{c}{\delta y}
\end{align*}
\]

respectively. To obtain a relationship between \( \gamma \) and \( \delta \), let \( z = z' \). Therefore,

\[
\gamma x + (1 - \gamma)y = x + \frac{c}{\delta y} \tag{7.15}
\]

or

\[
\gamma = 1 + \frac{c}{\delta y(x-y)} \tag{7.16}
\]

Equation (7.16) indicates that \( \gamma \) approaches 1 when \( \delta \to \infty \) regardless of the values of \( x \), \( y \), and \( c \). However, one can expect to obtain similar results by algorithms (I) and (II) of the "F" type problem when
Enter (1) Number of nodes and terminal
(2) mean and variance of travel time
(3) mean and variance of unload time
(4) mean and variance of demand
(5) capacity of truck

Use Algorithm I or II to calculate the savings in terms of travel time
(mean and variance)

Order the savings from the largest to smallest

Have all pairs of savings being used

Drop the previous best pair and consider the next best pair

Is this link feasible

Can add a link in front or back of route

Can two routes be joined

Add this node into the route

Is capacity exceeded

Create a new route

Figure 6. Algorithmic Flowchart of Procedural Steps for "F" Type Problem
\( \gamma \) and \( \delta \) accept large values in ranges zero to one and zero to infinite, respectively.

A flowchart summarizing the general structure of the analysis and solution of the SVRP is shown in Figure 7. The analysis begins with the determination of the type of the objective function which is to be minimized as was shown by problem C or D. The next step is the determination of the equivalent deterministic forms of problems C or D as were presented by the "E" or "F" type problem. The "E" or "F" type problem is used for the purpose of constructing routes. The routes are determined using the appropriate heuristic approach of "E" or "F" type problem.

If the decision maker is willing to accept the vehicle routes constructed by the RCS of the problem without change, then the procedure halts. Otherwise, the DM should consider the output information from the RCS of the problem and provide a set of goals for the RIS. Two GP models for the RIS of the problem are shown by problems A and B.

Prior to the formulation of each problem (each vehicle route is called a problem) as a 0-1 integer GP problem, values \( T_1 \) and \( T_2 \) for problems A and \( T \) for problem B should be evaluated. This 0-1 integer CP problem can be solved using either LIPARGP or LIPREGP techniques. The route improvement stage should be applied to those vehicle routes that do not satisfy the customer and decision maker's requirements after RCS of the problem has been applied.

7.5 Example Problem

The algorithm for the multiple objective GP model of the SVRP is illustrated by a simple example problem. The following small problem is involved with a single depot and 15 locations to be served by vehicles.
Formulate the SVRP as:
(1) "C" problem to minimize total cost (distance), or
(2) "D" problem to minimize total elapsed time of whole delivery system.

Determine the equivalent deterministic form of the selected problem which is referred to as an "E" or "F" type problem obtained from "C" or "D" respectively.

"E" type problem

Apply the developed heuristic approach for the "E" type problem

"F" type problem

Apply algorithm (I) or (II) for "F" type problem

Is solution satisfactory?
Yes → Stop
No → 1

Figure 7. General Structure of the Analysis and Solution of SVRP
Use the $k$th vehicle route as input data for RIS

Choose problem "A" as GP model

Choose problem "B" as GP model

Evaluate $\bar{T}_1$ and $\bar{T}_2$ by using equations:

$$\bar{T}_1 = \left( -\phi \Psi^k + (\phi^2 \Psi + 4T_1)^k / 2 \right)^2$$

and

$$\bar{T}_2 = \left( -\phi \Psi^k + (\phi^2 \Psi + 4T_2)^k / 2 \right)^2,$$

respectively.

Evaluate the artificial capacity of truck by equation:

$$Q = \left( -\phi \Psi^k + (\phi \Psi + 4Q)^k / 2 \right)^2$$

Formulate the problem as 0-1 integer GP problem and use LIPARGP or LIPREGP techniques to solve the problem.

Have all vehicle routes been considered?

Yes → Stop

No → $k = k + 1$

Figure 7: Continued
TABLE II

SUMMARY OF DISTANCES BETWEEN LOCATION IN MILES

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>60</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>63</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>36</td>
<td>45</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>87</td>
<td>75</td>
<td>63</td>
<td>90</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>48</td>
<td>42</td>
<td>33</td>
<td>48</td>
<td>52</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>102</td>
<td>45</td>
<td>99</td>
<td>81</td>
<td>78</td>
<td>66</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>90</td>
<td>54</td>
<td>78</td>
<td>81</td>
<td>29</td>
<td>45</td>
<td>39</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>72</td>
<td>39</td>
<td>57</td>
<td>60</td>
<td>39</td>
<td>27</td>
<td>45</td>
<td>21</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>51</td>
<td>63</td>
<td>27</td>
<td>60</td>
<td>36</td>
<td>21</td>
<td>57</td>
<td>48</td>
<td>42</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>36</td>
<td>30</td>
<td>18</td>
<td>72</td>
<td>30</td>
<td>78</td>
<td>69</td>
<td>48</td>
<td>42</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>48</td>
<td>39</td>
<td>69</td>
<td>21</td>
<td>105</td>
<td>66</td>
<td>81</td>
<td>93</td>
<td>72</td>
<td>81</td>
<td>39</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>63</td>
<td>21</td>
<td>75</td>
<td>36</td>
<td>96</td>
<td>60</td>
<td>57</td>
<td>75</td>
<td>60</td>
<td>81</td>
<td>45</td>
<td>24</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>45</td>
<td>15</td>
<td>48</td>
<td>27</td>
<td>69</td>
<td>30</td>
<td>57</td>
<td>54</td>
<td>36</td>
<td>48</td>
<td>21</td>
<td>39</td>
<td>30</td>
<td>00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>66</td>
<td>60</td>
<td>45</td>
<td>69</td>
<td>21</td>
<td>21</td>
<td>72</td>
<td>39</td>
<td>27</td>
<td>21</td>
<td>51</td>
<td>87</td>
<td>78</td>
<td>48</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>81</td>
<td>27</td>
<td>78</td>
<td>63</td>
<td>66</td>
<td>48</td>
<td>21</td>
<td>33</td>
<td>27</td>
<td>66</td>
<td>57</td>
<td>66</td>
<td>45</td>
<td>36</td>
<td>48</td>
<td>00</td>
</tr>
</tbody>
</table>

### TABLE III

**MEAN TRAVEL TIME BETWEEN LOCATIONS IN MINUTES**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>78</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>38</td>
<td>92</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>47</td>
<td>59</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>111</td>
<td>104</td>
<td>80</td>
<td>115</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>62</td>
<td>56</td>
<td>40</td>
<td>62</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>146</td>
<td>66</td>
<td>132</td>
<td>99</td>
<td>146</td>
<td>66</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>125</td>
<td>69</td>
<td>95</td>
<td>101</td>
<td>125</td>
<td>69</td>
<td>56</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>96</td>
<td>58</td>
<td>73</td>
<td>75</td>
<td>96</td>
<td>58</td>
<td>25</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>61</td>
<td>80</td>
<td>35</td>
<td>77</td>
<td>61</td>
<td>80</td>
<td>52</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>35</td>
<td>43</td>
<td>41</td>
<td>25</td>
<td>35</td>
<td>43</td>
<td>95</td>
<td>98</td>
<td>68</td>
<td>59</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>65</td>
<td>51</td>
<td>88</td>
<td>36</td>
<td>65</td>
<td>51</td>
<td>106</td>
<td>133</td>
<td>90</td>
<td>103</td>
<td>56</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>78</td>
<td>30</td>
<td>90</td>
<td>49</td>
<td>79</td>
<td>30</td>
<td>71</td>
<td>98</td>
<td>82</td>
<td>103</td>
<td>66</td>
<td>34</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>58</td>
<td>19</td>
<td>62</td>
<td>38</td>
<td>58</td>
<td>19</td>
<td>73</td>
<td>79</td>
<td>47</td>
<td>61</td>
<td>27</td>
<td>51</td>
<td>39</td>
<td>00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>90</td>
<td>73</td>
<td>55</td>
<td>96</td>
<td>90</td>
<td>73</td>
<td>100</td>
<td>51</td>
<td>37</td>
<td>29</td>
<td>61</td>
<td>119</td>
<td>94</td>
<td>64</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>108</td>
<td>34</td>
<td>93</td>
<td>90</td>
<td>108</td>
<td>34</td>
<td>28</td>
<td>40</td>
<td>37</td>
<td>79</td>
<td>68</td>
<td>94</td>
<td>56</td>
<td>46</td>
<td>68</td>
<td>00</td>
</tr>
</tbody>
</table>

*The original data has been multiplied by 60 and rounded off to the nearest integer value.

TABLE IV
SUMMARY OF DEMAND AND UNLOAD TIME
FOR EACH DEMAND POINT

<table>
<thead>
<tr>
<th>Demand Point</th>
<th>Demand Mean</th>
<th>Demand Variance</th>
<th>Unload Time (minutes) Mean</th>
<th>Unload Time (minutes) Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>30</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>29</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>37</td>
<td>37</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>17</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>22</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>20</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>33</td>
<td>33</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>37</td>
<td>37</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>25</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
Tables II, III, and IV summarize the necessary data for whole stations. Tables II and III give the summary of distance between locations in miles and travel time between locations in minutes, respectively. Table IV gives the mean and variance of demand and unload time at each demand point. Additional conditions are as follows:

1. Total unload time for each vehicle route is restricted to 120 minutes,
2. Total travelled time for each route is limited to 480 minutes,
3. Capacity of each truck is 80 units,
4. $\alpha_k = 0.1$, $\beta_k = \eta_k = 0.05$, and
5. The DM's requirements are given in details in Section 7.5.2.

This is because the DM can set the goal priority levels based on the results of the RCS of the problem.

The process of solution for this example problem is divided into four parts. In the first part, this problem is treated as an "E" type problem where minimization of the total travelled distance of the whole delivery system is considered as a criterion. The second part treats this problem as a "F" type problem considering the minimization of the total elapsed time of the whole delivery system as a criterion. The third part deals with the utilization of the developed LIGP technique to improve the sequence of stations on the constructed vehicle routes to meet the DM's requirements. The fourth part of the solution process is concerned with the sensitivity analysis of the results as were theoretically investigated in Chapter V.

The first and second parts of the solution process are illustrated in Section 7.5.1 while the third and fourth parts are discussed in Sections 7.5.2 and 7.6, respectively.
7.5.1 Route Construction Stage

If the above conditions along with all other assumptions being employed in this research apply to this example problem, then the "E" and "F" type problems can be solved by applying the proposed algorithms in order to determine the most satisfactory solution.

The solution to the "E" type problem, as formulated in Chapter V, is obtained via the computer program where constructed vehicle routes are \((0, 4, 7, 6, 15, 0), (0, 9, 14, 8, 13, 0), (0, 1, 12, 0), (0, 10, 3, 11, 0), \) and \((0, 2, 5, 0)\). The total distance, travel time, unload time, and customer demand of each vehicle route are shown in Table V. The total travelled distance is 810 miles and total travel and unload times are 1,240 and 215 minutes, respectively.

The next attempt was to solve the example problem (7.5) using the concept of the "F" type problem, as described in Chapter V. Two algorithms, (I) and (II), were employed with \(\gamma = 0.90\) and \(\delta = 0.50\) for these procedures, respectively. The constructed vehicle routes by algorithm (I) are \(\{0, 14, 7, 6, 15, 0\}, \{0, 1, 12, 0\}, \{0, 11, 4, 5, 0\}, \{0, 9, 8, 13, 0\}, \{0, 2, 0\}, \) and \(\{0, 3, 10, 0\}\) while the constructed vehicle routes by algorithm (II) are \(\{0, 14, 7, 6, 15, 0\}, \{0, 1, 12, 0\}, \{0, 9, 4, 10, 0\}, \{0, 2, 8, 13, 0\}, \) and \(\{0, 5, 11, 3, 0\}\). The details of the results in the case of the "F" type problem using algorithm (I) are given in Table VI. The results obtained in the case of "F" type problem using algorithm (II) are given in Table VII. The results show that the number of constructed vehicle routes in the case of the "F" type problem using algorithm (II) is smaller than that obtained from the algorithm (I). Therefore, one can verify Theorem (5.7) by comparing the total
TABLE V
SUMMARY OF RESULTS FOR "E" TYPE PROBLEM USING
$\alpha_k = 0.1$, $\beta_k = \eta_k = 0.05$ for all $k$

<table>
<thead>
<tr>
<th>Route Number</th>
<th>Distance (Mile)</th>
<th>Travel Time (minute)</th>
<th>Unload Time (minute)</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>267</td>
<td>454</td>
<td>43</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>180</td>
<td>251</td>
<td>48</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>144</td>
<td>203</td>
<td>42</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>111</td>
<td>177</td>
<td>40</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>155</td>
<td>42</td>
<td>79</td>
</tr>
<tr>
<td>Total</td>
<td>810</td>
<td>1,240</td>
<td>215</td>
<td>382</td>
</tr>
</tbody>
</table>

demand, travel, and unload times that are generated by algorithms (I) and (II) of the "F" type problem.

The results obtained from this simple example by the heuristic approaches of the "E" and "F" type problems can be compared using tables V, VI, and VII. The number of constructed vehicle routes using the "E" type problem is equal to that obtained by the algorithm (II) of the "F" type problem. However, the number of constructed vehicle routes by algorithm (I) is larger than that obtained by algorithm (II). Tables VI and VII indicate that the total travel and unload times and total generated demand by algorithms (I) and (II) are not equal. This is mainly because of the difference in the number of vehicle routes. On the other hand, the total travel and unload times and total demands obtained by algorithm (I) of the "F" type problem is larger than that obtained by the heuristic approach of the "E" type problem and algorithm (II) of the "F" type problem.
TABLE VI
SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING ALGORITHM (I) WHERE $\alpha_k = 0.1$, $\beta_k = \eta_k = 0.05$ ($\gamma = 0.9$)

<table>
<thead>
<tr>
<th>Route</th>
<th>Travel Time (minute)</th>
<th>Unload Time (minute)</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>356</td>
<td>44</td>
<td>74</td>
</tr>
<tr>
<td>2</td>
<td>203</td>
<td>42</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>274</td>
<td>40</td>
<td>72</td>
</tr>
<tr>
<td>4</td>
<td>236</td>
<td>40</td>
<td>63</td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>87</td>
<td>21</td>
<td>37</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1,264</strong></td>
<td><strong>219</strong></td>
<td><strong>386</strong></td>
</tr>
</tbody>
</table>

TABLE VII
SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING ALGORITHM (II) WHERE $\alpha_k = 0.10$, $\beta_k = \eta_k = 0.05$ ($\delta = 0.5$)

<table>
<thead>
<tr>
<th>Route Number</th>
<th>Travel Time (minute)</th>
<th>Unload Time (minute)</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>356</td>
<td>44</td>
<td>74</td>
</tr>
<tr>
<td>2</td>
<td>203</td>
<td>42</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>209</td>
<td>40</td>
<td>78</td>
</tr>
<tr>
<td>4</td>
<td>234</td>
<td>47</td>
<td>73</td>
</tr>
<tr>
<td>5</td>
<td>202</td>
<td>42</td>
<td>77</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1,204</strong></td>
<td><strong>215</strong></td>
<td><strong>382</strong></td>
</tr>
</tbody>
</table>
7.5.2 Route Improvement Stage

When the arrangement of stations on one route do not exactly or even partially meet the decision maker's needs, then the route improvement technique should be employed. This stage of the problem is required to sequence the stations on each route for the purpose of meeting the customer's and decision maker's criteria. The stations on each vehicle route are sequenced by using the LIGP technique. Prior to the utilization of this technique, the decision maker should consider the following information from the RCS:

1. constructed routes,
2. total demand of each route,
3. total expected cost, time or distance for the whole delivery system,
4. unload and travel times for each route, and
5. number of required vehicles.

In order to demonstrate the application of the route improvement stage of the problem that involves multiple conflicting goals, routes number 1 and 2 from the set of routes of the "E" type problem have been selected. The set of goals and priorities assigned to these routes are described in Sections 7.5.2.1 and 7.5.2.2, respectively. It is assumed that the sequence of stations on routes 3, 4, and 5 meet the decision maker's criteria.
7.5.2.1 Goals and Priorities for Route 1 of "E" Type Problem

Of course the primary objective is to reconstruct a new feasible route. Therefore, route feasibility has been given the first priority. Other priorities are as listed below (Problem B from Section 5.4.2):

P₂ - To minimize total travelled distance to 267 miles
P₃ - To minimize total travel time to 450 minutes and total unload time to 50 minutes
P₄ - To minimize the route safety stock
P₅ - To meet the dependency condition for station 7 which is to be served after station 6.

Before the formulation of Multiple Objective GP begins, one should use equation (5.84) from Theorem 5.2 in order to calculate the artificial capacity of truck, $\overline{Q}$:

$$\overline{Q} = \left[ (\phi \Psi^4 + (\phi^2 \Psi + 4Q)^4)/2 \right]^2$$

(7.17)

where $\phi = N^{-1}(1 - \eta_r)$ and $\Psi$ is as previously defined. By substituting $Q = 80$ units, $\Psi = 1$, and $\phi = 1.645$ [34, pp. 592-593] in the above equation, an artificial truck capacity of 66 units will be obtained.

The problem can be formulated as:

$$\text{Min} \quad P_1 \left[ \sum_{i=1}^{10} (n_i + p_i) + \sum_{i=11}^{22} p_i \right] + P_2 (p_{23}) + P_3 (p_{24} + p_{27}) + P_4 (p_{25}) + P_5 (p_{26} + n_{26})$$
Subject to:

\[\begin{align*}
X_{0,4} + X_{0,7} + X_{0,6} + X_{0,15} + n_1 - p_1 &= 1 \\
X_{4,0} + X_{4,7} + X_{4,6} + X_{4,15} + n_2 - p_2 &= 1 \\
X_{7,0} + X_{7,4} + X_{7,6} + X_{7,15} + n_3 - p_3 &= 1 \\
X_{6,0} + X_{6,4} + X_{6,7} + X_{6,15} + n_4 - p_4 &= 1 \\
X_{15,0} + X_{15,6} + X_{15,7} + X_{15,4} + n_5 - p_5 &= 1 \\
X_{4,0} + X_{7,0} + X_{6,0} + X_{15,0} + n_6 - p_6 &= 1 \\
X_{0,4} + X_{7,4} + X_{6,4} + X_{15,4} + n_7 - p_7 &= 1 \\
X_{0,7} + X_{4,7} + X_{6,7} + X_{15,7} + n_8 - p_8 &= 1 \\
X_{0,6} + X_{4,6} + X_{7,6} + X_{15,6} + n_9 - p_9 &= 1 \\
X_{0,15} + X_{4,15} + X_{7,15} + X_{6,15} + n_{10} - p_{10} &= 1
\end{align*}\]

\[\begin{align*}
z_1 - z_2 + 5X_{4,7} + n_{11} - p_{11} &= 4 \\
z_1 - z_3 + 5X_{4,6} + n_{12} - p_{12} &= 4 \\
z_1 - z_4 + 5X_{4,15} + n_{13} - p_{13} &= 4 \\
z_2 - z_1 + 5X_{7,4} + n_{14} - p_{14} &= 4 \\
z_2 - z_3 + 5X_{7,6} + n_{15} - p_{15} &= 4 \\
z_2 - z_4 + 5X_{7,15} + n_{16} - p_{16} &= 4 \\
z_3 - z_1 + 5X_{6,4} + n_{17} - p_{17} &= 4 \\
z_3 - z_2 + 5X_{6,7} + n_{18} - p_{18} &= 4 \\
z_3 - z_4 + 5X_{6,15} + n_{19} - p_{19} &= 4 \\
z_4 - z_1 + 5X_{15,4} + n_{20} - p_{20} &= 4 \\
z_4 - z_2 + 5X_{15,7} + n_{21} - p_{21} &= 4 \\
z_4 - z_3 + 5X_{15,6} + n_{22} - p_{22} &= 4
\end{align*}\]

\[\begin{align*}
87X_{0,4} + 90X_{0,7} + 102X_{0,6} + 81X_{0,15} + 87X_{4,0} + 39X_{4,7} + 39X_{4,15} + 90X_{7,0} + 39X_{7,4} + 39X_{7,6} + 33X_{7,15} = 4
\end{align*}\]
An attempt was made to evaluate the 0-1 integer solution of the above problem. Therefore, constraints of the following type

\[ x_i + n_i - p_i = 1 \quad \forall i, i = 1, \ldots, 20, \]

and an absolute priority level of

\[ P_0 = \sum_{i=1}^{20} P_{i+20} \]

were added to the original problem. The resulting problem consisted of 47 constraints, 118 variables (decision and deviational variables), and 6 priority levels. The absolute priority level was considered as the first priority and the original priorities of the problem were downgraded by one level. This problem was solved by the LIPARGP technique.
using the branch and bound method. Two new integer solutions, illustrated in the form of routes \((0, 15, 6, 7, 4, 0)\) and \((0, 6, 7, 15, 4, 0)\), were obtained. Other computer results are illustrated in Table VIII. Route \((0, 15, 6, 7, 4, 0)\) which satisfies the decision maker's criteria is the final solution to this problem.

### Table VIII

**SUMMARY OF RESULTS FOR ROUTE 1 OF "E" TYPE PROBLEM AFTER EMPLOYMENT OF THE RIS OF THE PROBLEM**

<table>
<thead>
<tr>
<th>Solution Number</th>
<th>Sequence of locations on each new route</th>
<th>Distance</th>
<th>Unload Time (minute)</th>
<th>Travel Time (minute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((0, 15, 6, 7, 4, 0))</td>
<td>267</td>
<td>34</td>
<td>428</td>
</tr>
<tr>
<td>2</td>
<td>((0, 6, 7, 15, 4, 0))</td>
<td>327</td>
<td>34</td>
<td>461</td>
</tr>
</tbody>
</table>

7.5.2.2 Goals and Priorities for Route 2 of "E" Type Problem

As expected, the route feasibility is given the first priority level where other priorities are (Problem A from Section 5.4.1):

- \(P_2\) - To minimize total travelled distance of each vehicle route to 180 miles,
- \(P_3\) - To minimize the unload time of vehicle route to \(T_2\) minutes and travel time of vehicle route to \(T_1\) minutes
- \(P_4\) - To meet the dependency conditions such that Station 8 follows Station 9.
The third priority level requires the minimization of travel and unload times to the levels of $T_1$ and $T_2$, respectively; hence, prior to the formulation of this Multiple Objective GP, $T_1$ and $T_2$ must be calculated. Equation (7.17) can be used again for the calculation of $T_1$. $Q$ and $\phi$ are substituted by 480 and 1.285, respectively. Similarly, in the evaluation of $T_2$, $Q$ and $\phi$ can be substituted with 120 and 1.645, respectively. The overall formulation of the problem for Route 2 of the "E" type problem is:

$$\text{Min} \quad P_1 \left[ \sum_{i=1}^{10} (n_i + p_i) + \sum_{i=11}^{22} (p_i) \right] + P_2 (p_{23}) + P_3 (p_{24} + p_{25}) + P_4 (n_{26} + p_{26})$$

Subject to:

$$X_{0,13} + X_{0,8} + X_{0,14} + X_{0,9} + n_1 - p_1 = 1$$
$$X_{13,0} + X_{13,8} + X_{13,14} + X_{13,9} + n_2 - p_2 = 1$$
$$X_{8,0} + X_{8,13} + X_{8,14} + X_{8,9} + n_3 - p_3 = 1$$
$$X_{14,0} + X_{14,13} + X_{14,8} + X_{14,9} + n_4 - p_4 = 1$$
$$X_{9,0} + X_{9,13} + X_{9,8} + X_{9,14} + n_5 - p_5 = 1$$

$$X_{13,0} + X_{8,0} + X_{14,0} + X_{9,0} + n_6 - p_6 = 1$$
$$X_{0,13} + X_{8,13} + X_{14,13} + X_{9,13} + n_7 - p_7 = 1$$
$$X_{0,8} + X_{13,8} + X_{14,8} + X_{9,8} + n_8 - p_8 = 1$$
$$X_{0,14} + X_{13,14} + X_{8,14} + X_{9,14} + n_9 - p_9 = 1$$
$$X_{0,9} + X_{13,9} + X_{8,9} + X_{14,9} + n_{10} - p_{10} = 1$$

$$Z_1 - Z_2 + 5X_{13,8} + n_{11} - p_{11} = 4$$
$$Z_1 - Z_3 + 5X_{13,14} + n_{12} - p_{12} = 4$$
$Z_1 - Z_4 + 5X_{13,9} + n_{13} - p_{13} = 4$

$Z_2 - Z_1 + 5X_{8,13} + n_{14} - p_{14} = 4$

$Z_2 - Z_3 + 5X_{8,14} + n_{15} - p_{15} = 4$

$Z_2 - Z_4 + 5X_{8,9} + n_{16} - p_{16} = 4$

$Z_3 - Z_1 + 5X_{14,13} + n_{17} - p_{17} = 4$

$Z_3 - Z_2 + 5X_{14,8} + n_{18} - p_{18} = 4$

$Z_3 - Z_4 + 5X_{14,9} + n_{19} - p_{19} = 4$

$Z_4 - Z_1 + 5X_{9,13} + n_{20} - p_{20} = 4$

$Z_4 - Z_2 + 5X_{9,8} + n_{21} - p_{21} = 4$

$Z_4 - Z_3 + 5X_{9,14} + n_{22} - p_{22} = 4$

$45X_{0,13} + 72X_{0,8} + 66X_{0,14} + 51X_{0,9} + 45X_{13,0} +$

$36X_{13,9} + 48X_{13,14} + 48X_{13,9} + 72X_{8,0} + 36X_{8,13} +$

$27X_{8,14} + 42X_{8,9} + 15X_{14,0} + 48X_{14,13} + 27X_{14,8} +$

$21X_{14,9} + 51X_{9,0} + 48X_{9,13} + 42X_{9,8} + 21X_{9,14} +$

$n_{23} - p_{23} = 180$

$11X_{0,13} + 11X_{8,13} + 11X_{14,13} + 11X_{9,13} + 11X_{0,8} +$

$11X_{13,8} + 11X_{14,8} + 11X_{9,8} + 7X_{0,14} + 7X_{13,14} +$

$7X_{8,14} + 7X_{9,14} + 9X_{0,9} + 9X_{13,9} + 9X_{8,9} + 9X_{14,9} +$

$n_{24} - p_{24} = 111$

$58X_{0,13} + 96X_{0,8} + 90X_{0,14} + 61X_{0,9} + 58X_{13,0} +$

$47X_{13,8} + 64X_{13,14} + 61X_{13,9} + 96X_{8,0} + 47X_{8,13} +$

$37X_{8,14} + 52X_{8,9} + 90X_{14,0} + 64X_{14,13} + 37X_{14,8} +$

$29X_{14,9} + 61X_{9,0} + 61X_{9,13} + 52X_{9,8} + 29X_{9,14} +$

$n_{25} - p_{25} = 452$

$X_{9,8} + n_{26} - p_{26} = 1$
A similar procedure for calculation of 0-1 integer solution for this problem was used. After the addition of all necessary constraints and the absolute priority level of $P_0$, the total number of constraints, variables, and priorities became 46, 116, and 5, respectively. This problem was solved by LIPARGP technique and a new solution illustrated in forms of route $\{0, 14, 9, 8, 13, 0\}$ was obtained.

The characteristics of this vehicle route are:

- total travelled distance = 210 miles
- total unload time = 38 minutes
- total travel time = 276 minutes

These results indicate that all decision maker's criteria, except the total travelled distance, have been achieved. However, it can be conclude that an increase of 30 miles in the total travelled distance has been sacrificed for the achievement of other goals set by the decision maker.

The final solution to this example problem, according to the decision maker's criteria and customer's requirements, is summarized in Table IX.

7.6 Sensitivity of Elapsed Time Upon The Probability of Route Failures

Thus far, the basic concepts of the SVRP and the derivation of solution methods have been the main objective of this research. However, an important part of any solution process is the analysis of the parameter changes after the final solution has been determined. This technique is defined as the sensitivity analysis of the procedure. The
degree of uncertainty in real world problems such as demands, travel and unload times, shipment, and costs has increased the utilization of sensitivity analysis in the decision making environments. Obviously, forecasting techniques can be used to predict the future values of the important parameters of the problem when the final solution is relatively sensitive to these factors.

The purpose of this section is to introduce some ideas concerning the analysis of elapsed time in the SVRP due to route failure probabilities. To illustrate this idea, it is necessary to review the example problem presented in the previous section. In the example, the route failure probabilities were considered to be $\alpha_k = 0.1$, $\beta_k = \eta_k = 0.05$. Now, what will be the number of vehicle routes, travel time, unload time, and travelled distance using different values of route failure probabilities by employing the "E" and "F" type problems?

<table>
<thead>
<tr>
<th>Route Number</th>
<th>Distance (mile)</th>
<th>Travel Time (minute)</th>
<th>Unload Time (minute)</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>267</td>
<td>428</td>
<td>34</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>210</td>
<td>276</td>
<td>38</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>144</td>
<td>203</td>
<td>42</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>111</td>
<td>177</td>
<td>40</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>155</td>
<td>42</td>
<td>79</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>840</strong></td>
<td><strong>1,239</strong></td>
<td><strong>196</strong></td>
<td><strong>382</strong></td>
</tr>
</tbody>
</table>

TABLE IX
THE SUMMARY OF FINAL RESULTS OF EXAMPLE PROBLEM BASED ON THE "E" TYPE PROBLEM ANALYSIS
The data and all necessary information of the example problem has been used in the solution process of the "E" type problem. The results are shown in Tables X and XI. Table X gives the summary of results when \( \beta_k = \eta_k \) and \( \alpha_k \) accept different values. On the other hand, Table XI illustrates the results for the "E" type problem when \( \eta_k \) is fixed and \( \alpha_k = \beta_k \) accepts different probability levels. The number of vehicle routes for this example using the "E" type problem under these probability levels is 5.

After solving the "E" type problem, the next objective was to solve the example problem (7.5) using the "F" type problem solution procedure where \( \gamma = 0.90 \). Table XII provides the travel time, unload time, and total elapsed time of each vehicle route developed by algorithm (I), where \( \alpha_k \) accepts different values and \( \beta_k = \eta_k \) are fixed. Table XIII provides the travel time, unload time, and total elapsed time of each vehicle route developed by algorithm (I) for fixed \( \eta_k \) and various probability levels for \( \alpha_k = \beta_k \) where \( \gamma = 0.90 \).

The example problem (7.5) was solved by the "F" type problem using the algorithm (II) where \( \delta = 0.50 \). These results are shown in Tables XIV and XV. However, Tables X, XII, and XIV indicate that if \( \alpha_k \) increases, then the travel time of each vehicle route decreases. For instance, Table XI shows that by increasing \( \alpha_k \) from 0.05 to 0.1, then the travel time of routes 1 and 2 decrease from 462 to 454 and 257 to 251, respectively. Tables XI, XIII, and XV support the results of Theorem 5.5, which has been proved in Chapter V. This means, for example, that if \( \alpha_k \) and \( \beta_k \) increase such that \( \alpha_k = \beta_k \), then the travel and unload times of each vehicle route will decrease.
TABLE X

SUMMARY OF RESULTS FOR THE "E" TYPE PROBLEM
FOR $\beta_k = \eta_k = 0.05$

<table>
<thead>
<tr>
<th>Route Number</th>
<th>$\alpha_k$</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05</td>
<td>462</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>257</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>208</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>181</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>159</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>0.10</td>
<td>454</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>251</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>203</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>177</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>155</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>0.30</td>
<td>438</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>239</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>193</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>167</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>146</td>
<td>42</td>
</tr>
<tr>
<td>Route Number</td>
<td>$\alpha_k$</td>
<td>$\beta_k$</td>
<td>Travel Time (minutes)</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>462</td>
</tr>
<tr>
<td>2</td>
<td>257</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>208</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>181</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>159</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>454</td>
</tr>
<tr>
<td>2</td>
<td>251</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>177</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>155</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.30</td>
<td>0.30</td>
<td>438</td>
</tr>
<tr>
<td>2</td>
<td>239</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>193</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>167</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>146</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE XII

SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING
ALGORITHM (I) WHERE $\beta_k = \eta_k = 0.05$
AND $\gamma = 0.90$

<table>
<thead>
<tr>
<th>Route Number</th>
<th>$\alpha_k$</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
<th>Total Elapsed Time of Each Route (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.025</td>
<td>368</td>
<td>44</td>
<td>412</td>
</tr>
<tr>
<td>2</td>
<td>212</td>
<td></td>
<td>42</td>
<td>254</td>
</tr>
<tr>
<td>3</td>
<td>285</td>
<td></td>
<td>40</td>
<td>325</td>
</tr>
<tr>
<td>4</td>
<td>246</td>
<td></td>
<td>40</td>
<td>286</td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td></td>
<td>32</td>
<td>147</td>
</tr>
<tr>
<td>6</td>
<td>93</td>
<td></td>
<td>21</td>
<td>114</td>
</tr>
<tr>
<td>1</td>
<td>0.10</td>
<td>356</td>
<td>44</td>
<td>400</td>
</tr>
<tr>
<td>2</td>
<td>203</td>
<td></td>
<td>42</td>
<td>245</td>
</tr>
<tr>
<td>3</td>
<td>274</td>
<td></td>
<td>40</td>
<td>314</td>
</tr>
<tr>
<td>4</td>
<td>236</td>
<td></td>
<td>40</td>
<td>276</td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td></td>
<td>32</td>
<td>140</td>
</tr>
<tr>
<td>6</td>
<td>87</td>
<td></td>
<td>21</td>
<td>108</td>
</tr>
<tr>
<td>1</td>
<td>0.30</td>
<td>342</td>
<td>44</td>
<td>386</td>
</tr>
<tr>
<td>2</td>
<td>193</td>
<td></td>
<td>42</td>
<td>235</td>
</tr>
<tr>
<td>3</td>
<td>262</td>
<td></td>
<td>40</td>
<td>302</td>
</tr>
<tr>
<td>4</td>
<td>225</td>
<td></td>
<td>40</td>
<td>265</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td></td>
<td>32</td>
<td>133</td>
</tr>
<tr>
<td>6</td>
<td>80</td>
<td></td>
<td>21</td>
<td>101</td>
</tr>
</tbody>
</table>
TABLE XIII

SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING ALGORITHM (II) WHERE $\eta_k = 0.05$ AND $\gamma = 0.90$

<table>
<thead>
<tr>
<th>Route Number</th>
<th>$\alpha_k$</th>
<th>$\beta_k$</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
<th>Total Elapsed Time of Each Route (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.025</td>
<td>0.025</td>
<td>368</td>
<td>46</td>
<td>414</td>
</tr>
<tr>
<td>2</td>
<td>212</td>
<td>44</td>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>285</td>
<td>41</td>
<td>326</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>246</td>
<td>41</td>
<td>287</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td>33</td>
<td>148</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>93</td>
<td>22</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.10</td>
<td>0.10</td>
<td>356</td>
<td>42</td>
<td>398</td>
</tr>
<tr>
<td>2</td>
<td>203</td>
<td>40</td>
<td>243</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>274</td>
<td>38</td>
<td>312</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>236</td>
<td>38</td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>30</td>
<td>138</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>87</td>
<td>19</td>
<td>106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.30</td>
<td>0.30</td>
<td>342</td>
<td>38</td>
<td>380</td>
</tr>
<tr>
<td>2</td>
<td>193</td>
<td>36</td>
<td>229</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>262</td>
<td>33</td>
<td>295</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>225</td>
<td>33</td>
<td>258</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>26</td>
<td>127</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>80</td>
<td>17</td>
<td>97</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE XIV

SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING ALGORITHM (II) WHERE $\beta_k = \eta_k = 0.05$ AND $\delta = 0.50$

<table>
<thead>
<tr>
<th>Route Number</th>
<th>$\alpha_k$</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
<th>Total Elapsed Time of Each Route (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.025</td>
<td>368</td>
<td>44</td>
<td>412</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>212</td>
<td>42</td>
<td>254</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>219</td>
<td>40</td>
<td>259</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>244</td>
<td>47</td>
<td>291</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>211</td>
<td>42</td>
<td>253</td>
</tr>
<tr>
<td>1</td>
<td>0.10</td>
<td>356</td>
<td>44</td>
<td>400</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>203</td>
<td>42</td>
<td>245</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>209</td>
<td>40</td>
<td>249</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>234</td>
<td>47</td>
<td>281</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>202</td>
<td>42</td>
<td>244</td>
</tr>
<tr>
<td>1</td>
<td>0.30</td>
<td>342</td>
<td>44</td>
<td>386</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>193</td>
<td>42</td>
<td>235</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>199</td>
<td>40</td>
<td>239</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>223</td>
<td>47</td>
<td>270</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>192</td>
<td>42</td>
<td>234</td>
</tr>
</tbody>
</table>
### TABLE XV

**SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING ALGORITHM (II) WHERE \( \eta_k = 0.05 \) AND \( \delta = 0.5 \)**

<table>
<thead>
<tr>
<th>Route Number</th>
<th>( \alpha_k )</th>
<th>( \beta_k )</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
<th>Total Elapsed Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.025</td>
<td>0.025</td>
<td>368</td>
<td>46</td>
<td>414</td>
</tr>
<tr>
<td>2</td>
<td>212</td>
<td>41</td>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>244</td>
<td>48</td>
<td>292</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>211</td>
<td>44</td>
<td>255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.10</td>
<td>0.10</td>
<td>356</td>
<td>42</td>
<td>398</td>
</tr>
<tr>
<td>1</td>
<td>203</td>
<td>40</td>
<td>243</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>209</td>
<td>38</td>
<td>247</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>234</td>
<td>44</td>
<td>278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>202</td>
<td>40</td>
<td>242</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.30</td>
<td>0.30</td>
<td>342</td>
<td>38</td>
<td>380</td>
</tr>
<tr>
<td>1</td>
<td>193</td>
<td>36</td>
<td>229</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>199</td>
<td>33</td>
<td>232</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>223</td>
<td>40</td>
<td>263</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>192</td>
<td>36</td>
<td>228</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.7 Summary

A heuristic algorithm based on the concept of Clarke and Wright's saving approach for the "E" type problem has been developed. The development of the heuristic approach for the "F" type problem was related to two new algorithms, (I) and (II). Two functions, one for each algorithm, have been developed and used as the basis of the saving evaluations in the "F" type problem. Two algorithms which consist the saving rules are used for partitioning a set of stations into feasible subsets using the concept of the Clarke and Wright procedure. Algorithms (I) and (II), which were developed in this chapter, have the capability of evaluating the savings for the SVRP where travel times are random variables. Hence, any SVRP with probabilistic customer demand, travel time, and unload time can be solved by employing the proposed heuristic approaches.

A simple example problem (7.5) is employed to illustrate the algorithm procedure. The process of solution of this example were divided into four sections. In the first part, the problem was treated as an "E" type problem, and in the second part as a "F" type problem, as discussed in Chapter V. The appropriate heuristic approaches of these types of problems were employed to design the vehicle routes. The third part of the analysis is related to the utilization of the developed LIGP technique for improving the sequence of stations on the constructed vehicle routes by the RCS of the problem for the "E" type problem. Finally, the example problem (7.5) was analyzed through the sensitivity analysis, as theoretically investigated in Chapter V. The results of this example problem fully support the theoretical background of the sensitivity analysis in relation to the route failure probabilities.
The most important characteristic of the developed algorithms for "E" and "F" type problems is to take into account the decision maker's and the customer's requirements. These procedures allow the decision maker to investigate and make good trade-off decisions concerning any possible criteria in the problem's environment.
CHAPTER VIII

ANALYSIS OF RESULTS

8.1 Introduction

The objective of this chapter is to analyze the results obtained from the "E" and "F" type problems developed in Chapter VII and the SVRP having only probabilistic customer demands. These results will be used to validate the new procedure for the SVRP. Three numerical examples demonstrate the performance of these algorithms. The validity of these procedures is evaluated by comparing the results with those of the existing saving methods for the SVRP having only probabilistic customer demands. A saving method developed by Stewart [54] is selected for the purpose of comparison of the results for the SVRP with only probabilistic customer demand. However, the lack of research in the area of SVRP with probabilistic customer demand and travel and unload times has made a comparison of results for this type of problem impossible. Therefore, the computational results obtained by Algorithms (I) and (II) of the "F" type problem, as described in Chapter VII, are only compared to each other.

Before entering into the analysis of the test problems, it is important to discuss difficulties which may arise due to the utilization of CCP in the SVRP. Specifically, the major difficulty with CCP is the determination of appropriate probability levels for constraints. Obviously, a reasonable approach is to provide a specific range for each
probability level for the important constraints and several probability levels for other constraints, then determine the corresponding results. The reason for considering a specific range for probability levels is two-fold. First, a specific range will prevent the problem from becoming too large. Second, the decision maker might not be interested in the whole range of the probability level which is from 0 to 1. The general assumption, of course, is that the manager of a delivery system is able to determine the value of these probability levels because of his or her familiarity, experience, and utilization of the CCP in the SVRP.

Another point that needs to be mentioned concerns the utilization of LIGP in the route improvement stage of the problem. The goal program developed in this research can only solve the linear and linear integer GP problems. For this reason, whenever a GP problem with nonlinear constraints appears to be a feasible option, a nonlinear integer goal program regarding the minimization of the priority levels must be employed. In this case, the GP model should be able to solve a nonlinear problem with 0-1 type decision variables.

In the examples given in this chapter for comparison purposes, one or more of the random variables are considered to be normally distributed. This is done for ease of comparison of the new model with the other models.

8.2 Validity of the New Model

To validate the new model, the results are first verified through hand computations to assure that the results satisfy all specified conditions. Specifically, such verification consists of determining that (1) the total demand of each route and truck capacity agree, (2) each
customer is served by only one truck, (3) the total travel time of each vehicle route satisfies the predetermined travel time level, and (4) there is no disagreement with the total unload time of each route and its predetermined level.

Next, the results obtained from this model are compared with those obtained from the saving method developed by Stewart [54]. Stewart's model is selected as a basis for comparison because the new model has similar characteristics provided that customer demands are probabilistic and travel and unload times are deterministic.

8.3 Comparison with the Stewart Model

In this section, two test problems proposed by Stewart [54] are used for the purpose of comparison. The detailed data for these problems are reproduced in Appendix C. The first test problem consists of fifty demand points where customer demands are considered to be normally distributed. The second test problem consists of 75 demand points with normally distributed customer demands. The objective of these two problems is the minimization of the total travelled distance between the stations.

Table XVI compares the results of Clarke and Wright's algorithm for the CCP problem, where \( \eta = 0.01, 0.025, 0.10, \) and 0.15 are considered to be the probability of route failure for customer demand. These results are based on the fifty node problem with truck capacity of 160 units and where customer demands are considered to be normally distributed. The proposed model produced routes requiring the same number as those derived by the Stewart algorithm. The total travel distance of
both algorithms are almost identical. Most likely, the small differences between the generated total travelled distance by these two procedures is due to round-off errors in integer calculations.

Table XVII summarizes the computational results of Clark and Wright's algorithm for the CCP problem of 75 demand points (second test problem) for $\eta$ values of 0.025, 0.05, and 0.10, where the capacity of each truck is considered to be 140 units. The customer demands were assumed to be normally distributed. The detailed data for this problem is shown in Appendix C. The results indicate that the same number of vehicle routes and nearly identical travel distances are obtained by both procedures. It is expected that the route distance will decrease with the increase in the $\eta$ probability level.

8.4 Validity of the Developed Heuristic Approaches

In this section the validity of the developed heuristic approaches for solving the "E" and "F" type problems is proven. A numerical problem (third test problem) is furnished in order to demonstrate some important points when "E" and "F" type problems are used. The model is solved with the total distance ("E" type problem) and total elapsed time ("F" type problem) as two separate objective functions. The data for this problem is randomly generated [51] with the following characteristics:

1. this problem is an extension of Steward's 50-node problem,
2. customer demands are normally distributed,
3. unload times are poisson distributed such that mean unload time is equal to mean customer demand,
### TABLE XVI

Comparison of Results of the 50 Demand Points with the Stewart Algorithm for the Chance-Constrained VRP with Normally Distributed Customer Demand

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>( \eta )</th>
<th>Stewart Distance (Mile)</th>
<th>Stewart Number of Routes</th>
<th>Proposed Procedure Distance (Mile)</th>
<th>Proposed Procedure Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.010</td>
<td>606</td>
<td>7</td>
<td>607</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>0.025</td>
<td>596</td>
<td>6</td>
<td>590</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>0.100</td>
<td>621</td>
<td>6</td>
<td>622</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>0.150</td>
<td>623</td>
<td>6</td>
<td>623</td>
<td>6</td>
</tr>
</tbody>
</table>

Truck Capacity = 160 units

### TABLE XVII

Comparison of Results for 75 Demand Points with the Stewart Algorithm for Chance-Constrained VRP with Normally Distributed Customer Demand

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>( \eta )</th>
<th>Stewart Distance (Mile)</th>
<th>Stewart Number of Routes</th>
<th>Proposed Procedure Distance (Mile)</th>
<th>Proposed Procedure Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.025</td>
<td>975</td>
<td>13</td>
<td>973</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>0.050</td>
<td>---</td>
<td>--</td>
<td>948</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>0.100</td>
<td>923</td>
<td>12</td>
<td>923</td>
<td>12</td>
</tr>
</tbody>
</table>

Truck Capacity = 140 units
4. mean travel time between stations i and j is considered to be a linear function of distance between stations i and j. It is evaluated through the following equation, mean of 
\[ t_{ij} = 1.2 \, d_{ij} + 3, \] and

5. the standard deviation, \( \sigma_{t_{ij}} \), of travel times was randomly generated using a uniform random number generator so that \( \sigma_{t_{ij}} \) fall between zero and 1/4 of the mean travel time of \( t_{ij} \).

In this case, the customer demands and travel and unload times are assumed to be independent of each other. The detailed data for characteristics 1 through 5 are shown in Appendix C.

The following conditions for solving this problem are shown in Table XVIII:

1. truck capacity with values of 140, 160, and 200 units,
2. unload time with values 60, 90, and 120 minutes for each vehicle route in order to solve the "E" type problem,
3. travel time with values 420, 390, and 360 minutes per route for solving the "E" type problem,
4. route failure probabilities of \( \alpha_k \), \( \beta_k \), and \( \eta_k \) can accept ranges \( 0 < \alpha_k < 0.20 \), \( 0 < \beta_k < 0.10 \), and \( 0 < \eta_k < 0.10 \),
5. in Algorithm (I) of "F" type problem, \( \gamma \) can accept values \( 0.70 < \gamma < 0.99 \), and
6. in Algorithm (II) of "F" type problem, \( \delta \) can accept values \( 0.50 < \delta < 4.0 \).

The amount of truck capacity, maximum value of travel and unload times per each vehicle route, value of probability levels, and other
factors such as $\gamma$ and $\delta$ are chosen arbitrarily. A set of nine subproblems considering different combinations and using previous conditions are designed and illustrated in Table XVIII.

<table>
<thead>
<tr>
<th>Subproblem Number</th>
<th>Truck Capacity</th>
<th>Mean Unload Time (minutes)</th>
<th>Travel Time (minutes)</th>
<th>$\alpha_k$</th>
<th>$\beta_k$</th>
<th>$\eta_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140</td>
<td>60</td>
<td>420</td>
<td>0.05</td>
<td>0.10</td>
<td>0.025</td>
</tr>
<tr>
<td>2</td>
<td>140</td>
<td>60</td>
<td>420</td>
<td>0.10</td>
<td>0.10</td>
<td>0.025</td>
</tr>
<tr>
<td>3</td>
<td>140</td>
<td>60</td>
<td>420</td>
<td>0.20</td>
<td>0.10</td>
<td>0.025</td>
</tr>
<tr>
<td>4</td>
<td>160</td>
<td>90</td>
<td>390</td>
<td>0.05</td>
<td>0.10</td>
<td>0.025</td>
</tr>
<tr>
<td>5</td>
<td>160</td>
<td>90</td>
<td>390</td>
<td>0.10</td>
<td>0.10</td>
<td>0.025</td>
</tr>
<tr>
<td>6</td>
<td>160</td>
<td>90</td>
<td>390</td>
<td>0.20</td>
<td>0.10</td>
<td>0.025</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>120</td>
<td>360</td>
<td>0.05</td>
<td>0.025</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>200</td>
<td>120</td>
<td>360</td>
<td>0.10</td>
<td>0.050</td>
<td>0.05</td>
</tr>
<tr>
<td>9</td>
<td>200</td>
<td>120</td>
<td>360</td>
<td>0.20</td>
<td>0.10</td>
<td>0.10</td>
</tr>
</tbody>
</table>

The purpose of this section is to solve these nine subproblems by treating them in the following categories:

Category 1. "E" type problem,

Category 2. "F" type problem using Algorithm (I) with $\gamma = 0.90$,

Category 3. "F" type problem using Algorithm (II) with $\delta = 0.50$. 
Eighteen additional subproblems are solved by treating subproblem 9 as

Category 4. "F" type problem using Algorithm (I) with \( \gamma = 0.70, 0.80, 0.82, 0.85, 0.87, 0.92, 0.95, 0.97 \) and 0.99,

and

Category 5. "F" type problem using Algorithm (II) with \( \delta = 0.40, 1.0, 1.5, 1.7, 2.0, 2.3, 2.5, 3.0, \) and 4.0.

As can be seen, a grand total of 45 subproblems will be solved using the data and information of this example problem.

Two procedures have been employed to solve this test problem. First, it is treated as an "E" type problem where the minimization of the total travelled distance is the criterion. Second, it is treated as a "F" type problem using the minimization of total elapsed time of the whole delivery system as a criterion. It is important to note that the upper bounds for travel and unload times for each route, as given in Table XVIII, are not used in the solution process of the problem when the "F" type procedure is employed. Therefore, the parameters to be considered at the time of analysis of results of the "F" type problem are truck capacity, probability levels \( \alpha_k, \beta_k, \) and \( \eta_k, \) and other factors such as \( \gamma \) and \( \delta. \)

8.4.1 Results of Category 1

Table XIX illustrates the minimization of the total travel distance of the delivery system under the existence of travel and unload time constraints and truck capacity (Category 1). The results indicate that by increasing the probability levels of \( \alpha, \beta, \) and \( \eta \) the travel and unload times of each vehicle route and consequently the total elapsed time of the whole delivery system decreases. The number of vehicle
routes and total travelled distance of the whole delivery system decreased from 17 to 8 routes and 1036 to 675 miles as the vehicle capacity increased from 140 to 200, respectively.

8.4.2 Results of Category 2

Table XX demonstrates the results of the nine subproblems for the "F" type problem using Algorithm (I) where $\gamma = 0.90$ (Category 2). The table shows that a better solution can be obtained in terms of minimum number of vehicles and minimum travel and unload times when this procedure is selected. It also shows that increasing truck capacity decreases the number of vehicle routes. The final observation is that when $\alpha_k$ (the route failure probability for travel time) increases, the total travel time of the whole delivery system decreases.

8.4.3 Results of Category 3

Table XXI, which is self explanatory, describes the improvement process in detail. The results show that the objective function which measures the total elapsed time of the whole delivery system has improved in all cases. It is important to note that the number of vehicle routes decreased as the truck capacity increased.

8.4.4 Results of Category 4

Table XXII shows the results of subproblem 9 where $\gamma$ has increased from 0.70 to 0.99. The number of vehicle routes and total unload times is fixed for all cases. The total travel times decrease as the value of $\gamma$ increases from 0.70 to 0.99. It is interesting to note
# TABLE XIX

**SUMMARY OF RESULTS OF "E" TYPE PROBLEM**

(CATEGORY 1)

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Distance (Mile)</th>
<th>Travel Time (Minutes)</th>
<th>Unload Time (Minutes)</th>
<th>Total Elapsed Time (Minutes)</th>
<th>Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1036</td>
<td>1474</td>
<td>918</td>
<td>2392</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>1036</td>
<td>1456</td>
<td>918</td>
<td>2374</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>1036</td>
<td>1436</td>
<td>918</td>
<td>2354</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>783</td>
<td>1140</td>
<td>891</td>
<td>2031</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>783</td>
<td>1128</td>
<td>891</td>
<td>2019</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>783</td>
<td>1113</td>
<td>891</td>
<td>2004</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>675</td>
<td>994</td>
<td>876</td>
<td>1870</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>695</td>
<td>1010</td>
<td>903</td>
<td>1913</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>675</td>
<td>973</td>
<td>876</td>
<td>1849</td>
<td>8</td>
</tr>
</tbody>
</table>

# TABLE XX

**SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM**

USING ALGORITHM (I) WHERE $\gamma = 0.90$

(CATEGORY 2)

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Total Travel Time (minutes)</th>
<th>Total Unload Time (minutes)</th>
<th>Total Elapsed Time (minutes)</th>
<th>Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>964</td>
<td>867</td>
<td>1831</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>953</td>
<td>867</td>
<td>1820</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>942</td>
<td>867</td>
<td>1809</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>959</td>
<td>861</td>
<td>1820</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>949</td>
<td>861</td>
<td>1810</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>939</td>
<td>861</td>
<td>1800</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>902</td>
<td>896</td>
<td>1798</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>878</td>
<td>876</td>
<td>1754</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>861</td>
<td>853</td>
<td>1714</td>
<td>5</td>
</tr>
</tbody>
</table>
### TABLE XXI

**SUMMARY OF RESULTS FOR THE "F" TYPE PROBLEM USING ALGORITHM (II) WHERE \( \delta = 0.50 \)**

*(CATEGORY 3)*

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Total Travel Time (minutes)</th>
<th>Total Unload Time (minutes)</th>
<th>Total Elapsed Time (minutes)</th>
<th>Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>995</td>
<td>872</td>
<td>1867</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>986</td>
<td>872</td>
<td>1858</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>972</td>
<td>872</td>
<td>1844</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>965</td>
<td>861</td>
<td>1826</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>955</td>
<td>861</td>
<td>1816</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>946</td>
<td>861</td>
<td>1807</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>944</td>
<td>853</td>
<td>1797</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>911</td>
<td>876</td>
<td>1787</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>895</td>
<td>854</td>
<td>1749</td>
<td>5</td>
</tr>
</tbody>
</table>

### TABLE XXII

**SOLUTION OF SUBPROBLEM 9 USING ALGORITHM (I) WITH VARIOUS VALUES OF \( \gamma \)**

*(CATEGORY 4)*

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
<th>Total Elapsed Time (minutes)</th>
<th>Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.70</td>
<td>877</td>
<td>853</td>
<td>1730</td>
<td>5</td>
</tr>
<tr>
<td>0.80</td>
<td>886</td>
<td>854</td>
<td>1740</td>
<td>5</td>
</tr>
<tr>
<td>0.82</td>
<td>877</td>
<td>853</td>
<td>1730</td>
<td>5</td>
</tr>
<tr>
<td>0.85</td>
<td>877</td>
<td>853</td>
<td>1730</td>
<td>5</td>
</tr>
<tr>
<td>0.87</td>
<td>877</td>
<td>853</td>
<td>1730</td>
<td>5</td>
</tr>
<tr>
<td>0.90</td>
<td>861</td>
<td>853</td>
<td>1714</td>
<td>5</td>
</tr>
<tr>
<td>0.92</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
<tr>
<td>0.95</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
<tr>
<td>0.97</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
<tr>
<td>0.99</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
</tbody>
</table>
that the total travelled time remains constant when $\gamma$ increases from 0.82 to 0.87 and from 0.92 to 0.99.

8.4.5 -Results of Category 5

Table XXIII illustrates the results of subproblem 9 where $\delta$ increases from 0.40 to 4.0. The amount of travel time remains constant as the value of $\delta$ increases from 1.0 to 2.3 and from 2.5 to 4.0. A total travel time of 887 minutes is obtained as $\delta$ increases from 1.0 to 2.3. No change in the number of vehicle routes occurred as $\delta$ increased from 0.40 to 4.0. The results from Tables XXII and XXIII indicate that the amount of total elapsed time provided by Algorithms (I) and (II) of the "F" type problem equalize as $\gamma$ and $\delta$ both are assigned large values. It is therefore concluded that Algorithms (I) and (II) are closely related and that the results of these algorithms can be used for the purpose of comparison.

8.5 Summary

Several aspects of the SVRP have been analyzed in this chapter. The computational experience of the proposed procedure on three test problems has been presented, and the computational results of a SVRP having only probabilistic customer demands on two test problems has been compared with the available procedure from the literature. It has been shown that a SVRP can be treated as both "E" and "F" type problems. Usually the "E" type problem is expected to produce a larger number of vehicle routes because the objective function measures the total travelled distance with restrictions on travel and unload times and truck capacity. On the other hand, the "F" type problem measures the total
<table>
<thead>
<tr>
<th>δ</th>
<th>Travel Time (minutes)</th>
<th>Unload Time (minutes)</th>
<th>Total Elapsed Time (minutes)</th>
<th>Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.40</td>
<td>899</td>
<td>853</td>
<td>1752</td>
<td>5</td>
</tr>
<tr>
<td>0.50</td>
<td>895</td>
<td>854</td>
<td>1749</td>
<td>5</td>
</tr>
<tr>
<td>1.00</td>
<td>887</td>
<td>854</td>
<td>1741</td>
<td>5</td>
</tr>
<tr>
<td>1.50</td>
<td>887</td>
<td>854</td>
<td>1741</td>
<td>5</td>
</tr>
<tr>
<td>1.70</td>
<td>887</td>
<td>854</td>
<td>1741</td>
<td>5</td>
</tr>
<tr>
<td>2.00</td>
<td>887</td>
<td>854</td>
<td>1741</td>
<td>5</td>
</tr>
<tr>
<td>2.30</td>
<td>887</td>
<td>854</td>
<td>1741</td>
<td>5</td>
</tr>
<tr>
<td>2.50</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
<tr>
<td>3.00</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
<tr>
<td>4.00</td>
<td>860</td>
<td>853</td>
<td>1713</td>
<td>5</td>
</tr>
</tbody>
</table>
elapsed time of the whole delivery system with no restrictions on travel and unload times for each vehicle route. The computational results of the experiments on the 45 subproblems show that the algorithm is capable of solving different types of SVRP considering different conditions. Additionally, sensitivity analysis on the final result can be performed by changing the probability levels, upper bounds on travel and unload times, truck capacity, and by using different values of $\gamma$ and $\delta$ for Algorithms (I) and (II) of the "F" type problem.
CHAPTER IX

USING INTERACTIVE COMPUTER PROGRAMS

9.1 Introduction

This chapter describes two interactive computer programs which primarily implement the route construction and route improvement stages of the SVRP. The computer program for the route construction stage of the problem, whether deterministic or probabilistic VRP, provides the user a tool for designing vehicle routes. The computer program for the PREGP and PARGP, whether the final solution for the decision variables is required to be continuous or integer, supply a good procedure for solving any types of LGP.

Both computer programs are interactive in such a way that the computer alerts the user for the necessary inputs. These two programs are coded in FORTRAN. The LIGP program is shown in Appendix A and the SVRP program is shown in Appendix B. For the user, the more important parameters are provided by the program in order to make the procedure faster and save system operating time. For instance, the values of $\gamma$ and $\delta$ are provided by the SVRP program and presented to the user for selection. Additionally, these two interactive computer programs are designed to investigate the user's input data and then prompt the user to correct probable errors or inconsistencies. As a safeguard, the program will move into a new stage only when the input has been checked by the program and verified by the user. These interactive procedures are coded
in such a way that any user, with or without previous knowledge about
the computer and/or the mathematical structure of these models, can eas-
ily operate the system to determine the most desireable solution. Fur­
thermore, the values to be entered are questioned and explained by the
program. For instance, when several values must be entered, the program
explains how to enter the data and instructs the operator to leave a
blank space between the entries.

The remainder of this chapter explains these two interactive pro-
cedures in more detail. The next two sections deal with the description
of the interactive computer program for the LIGP and for the SVRP,
respectively.

9.2 Interactive Linear Integer
Goal Programming

This section is concerned with the analysis of the linear integer
goal programming procedures, LIPREGP and LIPARGP, as coded in FORTRAN
and presented in Appendix A. The interactive procedure for the PREGP is
based on the simplex method approach described by Zeleny [69], while the
PARGP technique uses the concept of partitioning goal programming devel­
oped by Ravindaran [4]. These two techniques can be used to obtain both
continuous and integer solutions, and additionally, the interactive
integer PREGP technique can be used for the sensitivity analysis of the
problem. The two integer programming methods discussed in Chapter VI
are incorporated into these two goal programming procedures.

The interactive PREGP can be employed to

1. derive a solution to a general goal programming problem,
2. perform a sensitivity analysis,
3. derive an integer solution (pure or mixed integers) by the cutting plane method, and
4. derive an integer solution (pure or mixed integers) by the branch and bound procedure.

A need for the post optimality analysis comes after the solution of the problem has been obtained. This is because a number of changes may be necessary after the problem is solved. For instance, the goal attainment levels may increase or decrease, or the technological coefficients may need to be changed. This computer program can be used when it is desirable to evaluate a set of new solutions because of one or more of the following changes:

1. change one or more right-hand side values,
2. add one or more new decision variables,
3. add one or more new objective functions, or
4. change the coefficient of a nonbasic variable associated with the \( i^{th} \) row, \( j^{th} \) nonbasic column.

The PARGP performs exactly the same as the PREGP except that it does not consider the best feature of the PREGP, which is the sensitivity analysis. This is because when the optimal solution to the \( k^{th} \) subproblem is obtained, the PARGP procedure requires the deletion of all the nonbasic columns which have a negative value of \((Z_j - C_j)\) from the optimal tableau of the \( k^{th} \) problem for further consideration (before the addition of new goal constraints of such problem \( K+1^{th} \)). However, a continuous or integer solution to any problem with linear constraints and objective function terms can be obtained by each of these procedures. The algorithmic flowchart presented in Figure 8 gives the general idea of these computer programs.
Figure 8. Algorithmic Flowchart of the Computer Program for LIGP (Continued)
To add a New decision variable
To add a new objective function
To change the coeff. associated with the i\textsuperscript{th} row and j\textsuperscript{th} nonbasic column
To do no changes

Call subroutine \texttt{SENSTY}

Print the final results

Display of Menu 3

Choose the option

Integer solution by \texttt{PREGP} using Gomory procedure
Integer solution by \texttt{PARGP} using Gomory procedure
Integer solution by \texttt{PREGP} using \texttt{B & B}
Integer solution by \texttt{PARGP} using \texttt{B & B}

To keep the continuous solution

Call \texttt{FACT 1}
Call \texttt{FACT 2}
Call \texttt{BOUND}

Print the final Solution

All required problems solved

Yes
No

Stop
The first question asked by this program is the number of problems that are to be solved. After the user has input data and pressed the RETURN key, the program asks for verification of the operator's response to the question, then continues by displaying the options from Menu 1 as presented below. An input of "1" (PREGP) or "2" (PARGP) for this menu indicates that the LPREGP or LPARGP is to be used as a selected procedure for solving the desired problem.

DISPLAY OF MENU 1

CONTINUOUS SOLUTION BY PREGP PROCEDURE

*** ENTER 1 ***

CONTINUOUS SOLUTION BY PARGP PROCEDURE

*** ENTER 2 ***

*** CHOOSE THE OPTION ***

The user is allowed to choose a printing procedure for the intermediate computer calculation. The options are to

1. print all calculations in the tableau format, or
2. print only the basic variables and their values including the level of achievement of all priority goals.

After a continuous solution for the original problem is obtained, the program displays a menu of sensitivity analysis, Menu 2. One or more of the changes of the same type which are given in this menu can be performed at the same time. For instance, one can change one or more right hand side values or add one or more new decision variables of the problem.
*** DISPLAY OF MENU 2 ***

*** MENU FOR SENSITIVITY ANALYSIS ***

TO DO NO CHANGES ENTER 5

CHANGE THE RHS VALUES

** ENTER 1 **

TO ADD A NEW DECISION VARIABLE

** ENTER 2 **

TO ADD A NEW OBJECTIVE FUNCTION

** ENTER 3 **

TO CHANGE THE COEFFICIENT ASSOCIATED WITH THE

i^{th} row, j^{th} NONBASIC COLUMN

** ENTER 4 ***

*** CHOOSE THE OPTION ***

When no changes are required, the user enters 5 in order to move into the next stage. In this case, the program displays Menu 3, which allows the selection of choices for an integer method, or the option to terminate with a continuous solution only. In order to evaluate the integer solution of the required decision variables, the user can elect to use the final result of the original problem, or continue with the results associated with the last sensitivity analysis.

DISPLAY OF MENU 3

INTEGER SOLUTION BY PREGP USING GOMORY GP

*** ENTER 3 ***

INTEGER SOLUTION BY PREGP USING B & B

*** ENTER 4 ***
The method of data arrangement for this program is described in Section 9.2.1.

9.2.1 The Data Input Procedure

For the sake of time and quick data input, the operator is advised to arrange the data before the logon process begins. To use the \textit{PREGP} technique, the following data arrangement is necessary:

1. number of constraints, number of variables, and total number of priority levels,
2. number of original decision variables, number of positive and negative deviational variables,
3. number of nonzero elements in the left hand side of the constraints,
4. right hand sides values,
5. basis which is the list of the negative deviation variables, and
6. number of nonzero elements in all priority levels.

Care should be taken in the arrangement of the input data for the \textit{PARGP} procedure. The following steps should be followed before the arrangement of data for \textit{PARGP} starts:
1. break down the original problem into as many subproblems as the total number of priorities,

2. the data arrangement for the first subproblem is exactly the same as the data arrangement for PREGP,

3. the number of constraints and variables for the $k^{th}$ subproblem should be calculated as below:

   Number of constraints for the $k^{th}$ subproblem = (number of goal and rigid constraints used in all $K$ subproblems) – (number of goal and rigid constraints used in all $(k-1)$ subproblems), and,

   Number of variables for the $k^{th}$ subproblem = (number of variables (decisions and deviations) used in all $k$ subproblems) – (number of variables (decisions and deviations) used in all $(K-1)$ subproblems), and

4. enter "0" if no new constraint or no new variable has been used in the new subproblem.

9.3 Interactive Stochastic Vehicle Routing Problem

The main objective of this section is to describe the interactive computer program for the SVRP. Because deterministic VRP is a special case of the SVRP, the program is designed to solve any of these types of problems. A SVRP can be categorized as a VRP

1. with only probabilistic customer demand,

2. with probabilistic travel time, unload time, and customer demand when the total cost of whole system is expected to be minimized ("E" type problem), and
3. with probabilistic customer demand, travel and unload times when total elapsed time of whole delivery system is expected to be minimized ("F" type problem).

The algorithmic flowchart (Figure 9) gives the general idea about this interactive computer program.

The number of problems needed to be solved by the system is the first question the user responds to. Following the verification of this response by the operator, the program displays Menu 1 and expects a response of "1", "2", "3" or "4" as presented below:

DISPLAY OF MENU 1

SELECT ONE OF THE FOLLOWING

TO SOLVE THE DETERMINISTIC VRP

** ENTER 1 **

TO SOLVE A SVRP WITH PROBABILISTIC DEMAND

** ENTER 2 **

TO SOLVE SVRP OF "E" TYPE PROBLEM

** ENTER 3 **

TO SOLVE SVRP OF "F" TYPE PROBLEM

** ENTER 4 **

9.3.1 The Data Input Procedure

VRP: To prevent errors at the time of entering the data, the user needs to arrange the data before the logon process begins. The arrangement of data for the VRP is as shown below:

1. the distance type to be used (euclidean or linear),
2. number of demand points + depot,
3. truck capacity,
To solve 'VRP Call DETERM
Number of problem to be solved
Display of Menu 1
Choose the option

1

To solve 'VRP
Call DETERM
Print the final routes and other available information

To solve SVRP with probabilistic cust. demand
Call STATS

To solve SVRP of "E" type problem
Call PROB

To solve SVRP of "F" type problem

Use Algorithm I
Use Algorithm II

Print the final routes and other available information

Figure 9. Algorithmic Flowchart of Computer Program for SVRP (Continued)
Display of Menu 2

Choose the option

Change TCAP

Change UTIME or TTTime

Change $\alpha, \beta$ or $\eta$

Change the customer demands

Change the unload time

Change the coordinate locations

Change travel time between $i^{th}$ & $j^{th}$ location

Construct new routes by using the available savings

Construct new routes start from the beginning

Print the final routes and other available information

Figure 9. Continued
4. coordinate of depot first and then customer's locations, and
5. customer demands.

**SVRP:** Steps 1 through 4 of the data arrangement for the VRP can be used when one deals with the SVRP having only probabilistic customer demands. The remainder of steps for SVRP are:

5. the \( Z \) value (from the normal table) of the route failure probability when customer demand is probabilistic, and
6. mean and variance of customer demands.

The arrangement of data for the SVRP of the "E" type problem is shown below:

1. the type of distribution function for customer demand,
2. number of customer demands + depot,
3. capacity of truck,
4. total expected unload and travel times for each vehicle route,
5. the \( Z \) values for \( \alpha , \beta \), and \( \eta \) probability levels,
6. mean travel time,
7. variance of travel time,
8. mean and variance of unload time,
9. mean and variance of customer demand, and
10. type of distance (euclidean or linear).

The arrangement of data for the "F" type problem is very similar to the "E" type problem. However, in this case items 1, 4, and 10 are excluded from the data arrangement.

After all necessary data have been entered into the system, the computer will perform all necessary calculations and print the following outputs:
1. total cost, distance, or time depending on the nature of criteria,
2. set of all constructed vehicle routes,
3. total demand of each vehicle route,
4. total travel and unload times for each vehicle route (for "E" and "F" type problems), and
5. number of required vehicles.

In many cases the user will find it necessary to evaluate a set of new solutions by incorporating one or more of the following modifications:

1. change the truck capacity,
2. change the total unload and travel times,
3. change the value of $\alpha$, $\beta$, and $\eta$,
4. change one or more customer demands,
5. change the unload time at one or more of stations,
6. change the coordinate of locations, and/or
7. change the travel time between the $i^{th}$ and $j^{th}$ customer.

One or more of these changes can be made when the algorithm displays Menu 2 as shown below:

DISPLAY OF MENU 2

SELECT ONE OF THE FOLLOWING

TO CHANGE THE CAPACITY OF TRUCK

** ENTER 1 **

TO CHANGE THE "UTIME" OR "TTTIME"

** ENTER 2 **

TO CHANGE "ALPHA", "BETA" AND "ETA"

** ENTER 3 **
TO CHANGE THE COORDINATE OF LOCATIONS
** ENTER 4 **

TO CHANGE THE CUSTOMER DEMAND
** ENTER 5 **

TO CHANGE THE UNLOAD TIME
** ENTER 6 **

TO CHANGE THE TRAVEL TIME
** ENTER 7 **

** TO DO NO CHANGES ENTER 8 **

When no changes are expected, the user should enter "8". In this case, the user is allowed to do one of the following:

1. enter "1" to change an "E" type problem to a "F" type problem,
2. enter "2" to change a "F" type problem to an "E" type problem, or
3. enter "3" to terminate.

Changes 1 through 5 of Menu 2 do not require recalculation and ranking of the savings which are associated with each of the two demand points. When the appropriate information for any specific changes has been entered, the program restarts the process of route construction, considering all new and old restrictions of the problem. Although these types of changes save operation as well as computer time, the 6th and 7th changes from list of Menu 2 save only operation time. For these two cases, the program restarts all necessary calculations for savings evaluations and ranks these savings from the largest to the smallest, as described in Chapter VII.
This program is well structured for considering more than one change at a time. For instance, it is possible to change truck capacity, values $\alpha$, $\beta$ and $\eta$, customer demands, and/or coordinates of locations before moving toward the calculation process. However, the program's structure demands that when making such changes, the coordinates of locations and travel time be considered last.

9.4 Summary

In this chapter, the important features of the interactive computer programs for SVRP and linear integer goal programming are described. Also, methods in which the decision maker can use the interactive LIPREGP for sensitivity analysis and determination of integer solutions using one of the LIGP techniques are demonstrated. A LIPARGP technique which can be used for deriving a continuous or an integer solution is also given. The methods of data arrangements for these two procedures are fully described.

The interactive SVRP is described in detail and it is shown that three different categories of these types of problems can be solved by this procedure. Without termination from the program, the user can change the type of problem which was being used previously. Also, the interactive SVRP can be used when the operator desires to evaluate a set of new solutions by changing the truck capacity, total travel and unload times, and/or probability levels.
CHAPTER X

CONCLUSIONS AND RECOMMENDATIONS

10.1 Conclusions

This dissertation has presented a study of a GP model of the general SVRP in which travel time, unload time, and customer demands are random variables. This research extends the state of the art in multiple objectives SVRPs by fulfilling the primary and secondary objectives and all the subobjectives of Chapter I. That is:

1. A GP model of the problem within the framework of the SVRP has been mathematically formulated,

2. A SVRP in which travel time, unload time, and customer demands may be represented as random variables has been developed,

3. An equivalent deterministic form of the SVRP for RCS and RIS of the problem has been formulated,

4. The existence of a new set of deterministic linear time constraints which are equivalent to the nonlinear set of time constraints of the problem for distributions such as poison, binomial, negative binomial, gamma, chi-square and exponential has been proven through Theory 5.2,

5. A linear GP formulation of the RIS of the problem where conflicting multiple objectives are treated explicitly has been developed,

6. The effects of the route failure probabilities of \( \alpha_k \) and \( \beta_k \) on the total elapsed time of the system with \( 0 \leq \alpha_k \leq 1 \) and
0 ≤ β_k ≤ 1 for all k have been proven through Theories 5.3, 5.4, and 5.5,

7. The existence of the optimum solution for the route construction stage of the problem has been proven through Theory 5.1,

8. A heuristic algorithm which is a modification of the Clarke and Wright heuristic procedure has been developed in order to solve the "E" type problem,

9. Two new heuristic approaches based on the concepts of the Clarke and Wright algorithm have been originated in order to solve the "F" type problem. The computational experiments show that these two procedures are closely related,

10. A comprehensive, interactive computer program for the SVRP has been developed and described. This program is capable of solving a VRP, a SVRP having only probabilistic customer demands, and a SVRP with "E" and "F" type problems. This program allows the decision maker's involvement in the solution process of the problem. The decision maker can evaluate the solution by changing the value of probability levels, truck capacity, customer demands, and other important parameters of the problem to fully analyze the sensitivity of the final solution, and

11. An interactive Computer Program for the LIGP using the concepts of preemptive and partitioning GP has been developed and described to determine the most favorable vehicle routes of the multiple objective SVRP's where the decision policies and customer requirements need to be fully considered. The interactive procedure allows a decision maker to provide an integer solution for the problem, and to understand the behavior of the system through the utilization of the sensitivity analysis of the optimal solution.
10.2 Recommendations

There are several directions in which additional research should be conducted in the area of the SVRP and the LIGP technique. Some possible considerations for future research are presented below:

1. To develop an interactive computer program as a link between the interactive SVRP and LIGP programs in order to eliminate the operator's time for mathematical formulation of the GP problem which is based on the constructed vehicle routes from the RCS of the problem.

2. To develop new heuristic approaches for solving the "E" and "F" type problems.

3. To develop an iterative procedure that solves the "F" type problem optimally.

4. To apply a computer graphic system to the interactive procedure developed for the SVRP to help the decision maker visualize the constructed vehicle routes.

5. To consider other stochastic elements such as vehicle breakdowns together with the SVRP which is developed in this research.

6. To develop a heuristic approach for solving a GP problem where decision variables are required to be 0-1.

7. To develop an interactive nonlinear integer goal program that can solve the nonlinear constraints of the type generated by the CCP with 0-1 decision variables.
BIBLIOGRAPHY


APPENDIX A

INTERACTIVE COMPUTER PROGRAM FOR LINEAR INTEGER GOAL PROGRAMMING
FORTRAN PROGRAM TO SOLVE THE LINEAR INTEGER PREEMPTIVE GOAL PROGRAMMING (LIPREGL) AND LINEAR INTEGER PARTITIONING GOAL PROGRAMMING (LIPARPG) PROBLEMS

AUTHOR: YAHYA ZARE-MEHRJERDI
ADVISOR: DR. M. P. TERRELL
COMPUTER: IBM 3061D
DATE: NOVEMBER 1986

SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT
OKLAHOMA STATE UNIVERSITY
STILLWATER, OK 74078

THIS PROGRAM ALLOWS THE USER TO FIND A CONTINUOUS OR INTEGER SOLUTION OF A LINEAR GOAL PROGRAMMING PROBLEM USING PREEMPTIVE GOAL PROGRAMMING (PREGP) OR PARTITIONING GOAL PROGRAMMING (PARGP) METHODS.

AN INTEGER SOLUTION OF PROBLEM CAN BE OBTAINED USING EITHER CUTTING PLANE OR BRANCH AND BOUND TECHNIQUES.

ADDITIONALLY, USER CAN OBTAIN A MIXED INTEGER SOLUTION OF THE PROBLEM BY EMPLOYING EITHER OF THESE GP METHODS.

FINALLY, PREGP CAN BE USED FOR THE PURPOSE OF SENSITIVITY ANALYSIS OF THE PROBLEM.

THE FOLLOWING SUBROUTING ARE USED IN THIS PROGRAM.

- INTERS = TO READ THE INPUT DATA
- PIVCOL = TO FIND THE PIVOT COLUMN
- PIVROW = TO FIND THE PIVOT ROW
- CALC = TO UPDATE THE NEW TABLEAU
- PARTG = IS USED FOR PARTITIONING GP.
- ACHECK = TO DETERMINE THE # OF ALTERNATIVE SOLUTIONS FOR PARGP
- INILS = TO PREPARE THE INITIAL TABLEAU FOR SUBPROBLEMS
OTHER THAN THE FIRST ONE FOR PARGP
TSORT = IT SORTS THE BASIC AND NONBASIC VARIABLES WHOSE VALUES ARE REQUIRED TO BE INTEGER
BOUND = TO CONTROL THE PROGRAM FOR THE BRANCH AND BOUND PROCEDURE
BRNCH = TO DETERMINE THE SOURCE ROW AND DEVELOPE THE NEW CONSTRAINTS FOR THE BRANCH AND BOUND PROCEDURE
SUMMARY = IT STORE ALL INTEGER SOLUTIONS IN ORDER TO PROVIDE THE USER WITH THE LIST OF ALL INTEGER VARIABLES
MEMOH = CONSISTS THE CONTENTS OF MENU 1, 2 AND 3.
ADUP = TO ENTER THE DATA FOR NEW PRIORITY LEVEL
DUALS = STANDS FOR DUAL SIMPLEX METHOD
PIVROW= TO FIND THE PIVOT ROW
FACT1 = TO CONTROL THE PROGRAM FOR PREEMPTIVE LIGP TECHNIQUE
FACT2 = TO CONTROL THE PROGRAM FOR PARTITIONNING LIGP TECHNIQUE
INTGR = TO DEVELOP THE GOMORY CUTTING PLANE FOR PURE AND MIXED INTEGER VARIABLES FOR BOTH PREGP AND PARGP TECHNIQUES
SADD= TO PROVIDE THE OBJECTIVE FUNCTION FOR THE FINAL TABLEAU AND TO DOWNGRADE THE PREVIOUS OBJECTIVES BY ONE LEVEL
GOMORY= TO FIND THE SOURCE ROW AND COUNT THE # OF AVAILABLE INTEGER VARIABLES
BINVRS= TO CALCULATE THE INVERSE OF MATRIX B
SENSTY= TO PERFORM THE SENSITIVITY ANALYSIS FOR PREGP.

DEFINITION OF VARIABLES

STANDS FOR PREEMPTIVE GOAL PROGRAMMING PROCEDURE
STANDS FOR PARTITIONNING GOAL PROGRAMMING PROCEDURE
TOTAL NUMBER OF PRIORITIES
NUMBER OF PRIORITIES. NOTE THAT NOPRO= NPR0 FOR PREEMPTIVE PROCEDURE. NOPRO=1 FOR SGL.
NUMBER OF VARIABLES
NUMBER OF CONSTRAINTS
IS THE NUMBER OF CONSTRAINTS PLUS ONE
IS THE NUMBER OF CONSTRAINTS PLUS PRIORITIES
IS THE NUMBER OF ALL VARIABLES PLUS ONE
C* NORS = NUMBER OF ORIGINAL DECISION VARIABLES
C* LND1 = NUMBER OF ORIGINAL NEGATIVE DEVIATIONS
C* LPD1 = NUMBER OF ORIGINAL POSITIVE DEVIATIONS
C* NPRNT = 1 MEANS PRINT ALL INTERMEDIATE TABLEAUES
C* = 2 MEANS TO PRINT THE LIST OF ALL VARIABLES AND
C*  PRIORITY LEVELS AND THEIR VALUES
C* IBOUND = 0 USE LIPREGP TOGETHER WITH THE CUTTING PLANE METHOD
C* = 1 USE LIPARGP TOGETHER WITH THE CUTTING PLANE METHOD
C* = 3 USE THE BRANCH AND BOUND TECHNIQUE
C* INTGP = 0 KEEP THE FINAL SOLUTION CONTINUOUS
C* = 1 FIND THE INTEGER SOLUTION OF THE PROBLEM
C* NXREAL = NUMBER OF INTEGER VARIABLES
C* INTRANT = NUMBER OF ITERATIONS
C* KING = COUNTS THE NUMBER OF VARIABLES FOR THE PARGP PROCEDURE
C* NNZRD = NUMBER OF NONZERO ELEMENTS IN THE NEW CONSTRAINTS
C* IPZRD = NUMBER OF NONZERO ELEMENTS IN THE NEW PRIORITY
C* KINPRD = IS THE NUMBER OF PRIORITIES INCLUDING THE ABSOLUTE ONE
C* IPROBL = NUMBER OF PROBLEMS TO BE SOLVED
C* NONZRD = NUMBER OF NONZERO ELEMENTS IN THE MATRIX OF
C*  TECHNOLOGICAL COEFFICIENTS
C* FRACT = IS THE FRACTION PART OF THE VALUE OF A VARIABLE
C* KNOC = NUMBER OF NEW CONSTRAINTS TO BE ADDED IN PARGP
C* KNOV = NUMBER OF NEW VARIABLES TO BE ADDED IN PARGP
C* IPVC = STANDS FOR PIVOT COLUMN
C* IPROW = INDICATES PIVOT ROW
C* INO = IS A COUNTER
C* XMAX = STANDS FOR THE MAXIMUM
C* IB(I) = ARRAY OF BASIC VARIABLES
C* ID(I) = ARRAY OF ALL VARIABLES(DECISION PLUS DEVIATION)
C* IN(I) = ARRAY OF GOAL ACHIEVEMENT LEVELS
C* IV(I) = ARRAY OF VALUE OF RIGHT HAND SIDE VALUES
C* IBOR(I) = ARRAY OF BASIC VARIABLES (USED FOR SENSITIVITY ANAL.)
C* IVZAR(I) = ARRAY OF VALUE OF RHS(USED FOR THE SENSITIVITY ANAL.)
C* ISDL(I) = ARRAY OF ORIGINAL LIST OF VARIABLES(USED FOR
C*  SENSITIVITY ANAL.)
C* SSIN(I) = ARRAY OF GOAL ACHIEVEMENT LEVELS(USED FOR SENSITIVITY
C* ANAL.)
C* LDECS(I) = ARRAY OF DECISION VARIABLES
C* LPDEV(I) = ARRAY OF POSITIVE DEVIATIONS
C* LNDEV(I) = ARRAY OF NEGATIVE DEVIATIONS
C* IREAL(I) = LIST OF REQUIRED INTEGER VARIABLES
C* IPM(I) = 0 INDICATES THAT VARIABLE I IS NONBASIC
C* = OTHERWISE THE NONBASIC VARIABLE HAS ALREADY BEEN
C* DELETED FROM THE TABLEAU
C* DUM(I) = ARRAY OF CUTTING PLANE
C* NONBAS(I) = THE LIST OF NONBASIC VARIABLES FROM THE REQUIRED
C* INTEGER VALUED VARIABLES
C* KBASE(I) = THE LIST OF BASIC VARIABLES FROM THE LIST OF THE
C* REQUIRED INTEGER VALUED VARIABLES
C* XDOM(I) = ARRAY OF NEW CONSTRAINTS USED IN THE BRANCH AND
C* BOUND TECHNIQUE
C* XMDD(I) = IT IS EQUAL TO THE -XDOM(I)
C* SIV(I) = TO SAVE THE VALUE OF THE RHS VALUES
C* ISIN(I) = TO SAVE THE LIST OF THE BASIC VARIABLES
C* SIN(I) = TO SAVE THE VALUE OF THE PRIORITY LEVELS
C* ISID(I) = TO SAVE THE LIST OF THE DECISION VARIABLES
C* AIV(I) = TO SAVE THE VALUE OF PRIORITY LEVELS OF THE OPTIMAL
C* TABLEAU FOR SOLVING SUBPROBLEM TWO
C* AIN(I) = TO SAVE THE RHS VALUE OF THE OPTIMAL TABLEAU FOR
C* SOLVING SUBPROBLEM TWO
C* IDVAR(I) = TO SAVE THE LIST OF THE VARIABLES OF THE OPTIMAL
C* TABLEAU FOR SOLVING SUBPROBLEM TWO
C* IABASE(I) = TO SAVE THE LIST OF THE BASIC VARIABLES OF THE
C* OPTIMAL TABLEAU FOR SOLVING SUBPROBLEM TWO
C* KBB(I) = TO SAVE THE LIST OF THE BASIC VARIABLES OF OPTIMAL
C* TABLEAU OF SUBPROBLEM 1
C* RHV(I) = TO SAVE THE VALUE OF THE BASIC VARIABLES OF THE
C* OPTIMAL TABLEAU OF SUBPROBLEM 1
C* ARHN(I) = TO SAVE THE VALUE OF THE PRIORITY LEVELS OF THE
C* OPTIMAL TABLEAU OF SUBPROBLEM 1
C* KDD(I) = TO SAVE THE LIST OF THE DECISION VARIABLES OF THE
C* OPTIMAL TABLEAU OF SUBPROBLEM 1
C* F(I) = TO SAVE THE VALUE OF THE PRIORITY LEVEL OF SUB-
C* PROBLEM 1
C* FF(I) = TO SAVE THE VALUE OF THE PRIORITY LEVELS OF SUB-
C* PROBLEM 2
C* SARRY(I) = TO SAVE THE INTEGER SOLUTION OF VARIABLES
C* XARRY(I) = TO SAVE THE LIST OF THE INTEGER VARIABLES
C* IBU(I) = TO SAVE THE LIST OF BASIC VARIABLES OF THE OPTIMAL
C* TABLEAU OF SUBPROBLEM 2
C* IUN(I) = TO SAVE THE VALUE OF THE PRIORITY LEVELS OF THE
C* OPTIMAL TABLEAU OF SUBPROBLEM 2
C* IUD(I) = TO SAVE THE LIST OF THE DECISION VARIABLES OF THE
C* OPTIMAL TABLEAU OF SUBPROBLEM 2
C* IUU(I) = TO SAVE THE VALUE OF THE BASIC VARIABLES OF THE
C* OPTIMAL TABLEAU OF SUBPROBLEM 2
C* TAB(I,J) = ARRAY OF TABLEAU OF THE ORIGINAL PROBLEM
C* ZZ(I,J) = ARRAY OF TABLEAU OF THE JTH VARIABLE OF PRIORITY 1
C* C(I,J) = ARRAY OF THE PRIORITY WEIGHTS FOR THE JTH VARIABLE
C* AND ITH PRIORITY LEVEL
C* Z(I,J) = ARRAY OF TABLEAU OF PRIORITY VALUES
C* SSV(I,J) = AN ARRAY USED FOR SENSITIVITY ANALYSIS
C* STOF(I,J) = AN ARRAY USED FOR THE SENSITIVITY ANALYSIS
C* ATAB(I,J) = ARRAY OF THE OPTIMAL TABLEAU USED FOR THE BRANCH AND
C* BOUND TECHNIQUE
C* AZE(I,J) = THIS ARRAY IS USED IN THE BRANCH AND BOUND TECHNIQUE
C* ATT(I,J) = ARRAY OF OPTIMAL TABLEAU OF SUBPROBLEM 1
C* AZZ(I,J) = ARRAY OF PRIORITY COEFFICIENTS OF OPTIMAL TABLEAU
C* OF SUBPROBLEM 1
C* TUT(I,J) = ARRAY OF OPTIMAL SOLUTION OF SUBPROBLEM 2
C* ZUZ(I,J) = ARRAY FROM THE OPTIMAL TABLEAU OF SUBPROBLEM 2
C*SENS(I,J) = GIVES THE MATRIX OF B INVERS
C* SZZ(I,J) = THIS ARRAY IS USED FOR THE SENSITIVITY ANALYSIS
C* STABB(I,J) = THIS TABLEAU IS USED IN THE SENSITIVITY ANALYSIS
C
C*********************************************************************************
C******************************************************************************
C******************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************

C*********************************************************************************
C*********************************************************************************
C*********************************************************************************
WRITE(6,8) IPROBL
WRITE(10,8) IPROBL
8 FORMAT(//2X,'NUMBER OF PROBLEMS=',2X,I2)
IF(IPROBL.LE.0) THEN
WRITE(6,6)
WRITE(10,6)
6 FORMAT(//5X,'REENTER AGAIN')
GO TO 7
ENDIF
DO 3001 I=1,IPROBL

C*** TO COUNT THE NUMBER OF ITERATIONS
ITRATN=0
IBUND=0
ICOUNT=0
C* TO KEEP THE NUMBER OF VARIABLES.
KAT=0
C* TO DISPLAY MENU 1
C*

JOYL=1
CALL MEMOH(JOYL,ISEN)
IF (IPART.EQ.1) CALL PARTG
IF(IPEMPT.EQ.1) GO TO 1000
GO TO 3003
C**** CALL FOR INPUT DATA
1000 ICHECK=0
WRITE (6,151)
WRITE(10,151)
151 FORMAT(//10X,'ALGORITHM IS USING GP PREEMTIVE PROCEDURE')
INTUR=1
CALL INTERS (INTUR)
C*** CALL FOR PIVOT COLUMN AND PIVOT ROW
C***

IND=1
DO 20 IND=1,NPRO
10 CALL PIVCOL
IF(XMAX.EQ.0) GO TO 20
IF(IPVC.EQ.0) GO TO 20
CALL PIVROW(IPROW)
CALL CALC(IPROW)
CALL PTRG(NDV)
C**
IF(IND.EQ.1) GO TO 10
IF(IN(IND).EQ.0) GO TO 20
GO TO 10
20 CONTINUE
ISSS=0
C TO SPECIFY THE TYPE OF SENSITIVITY ANALYSIS
502 WRITE(6,500)
WRITE(10,500)
500 FORMAT(5X,'DO YOU WISH TO DO ANY SENSITIVITY ANALYSIS')
WRITE(6,501)
WRITE(10,501)
501 FORMAT(5X,'** ENTER 1 FOR YES **'/5X,'** ENTER 2 FOR NO **')
READ(5,*), NOYES
IF(NODYES.EQ.2) GO TO 3003
C-
C TO DISPLAY MENU 3
C-
JOYL=2
CALL MEMOH(JOYL,ISEN)
IF(ISEN.EQ.5) GO TO 3003
ISSS=ISSS+1
IF(ISSS.EQ.1) THEN
IAM=1
ELSE
WRITE(6,503)
WRITE(10,503)
503 FORMAT(/5X,'DO YOU LIKE TO WORK WITH THE FINAL TABLEAU'/'
+5X,'OF THE ORIGINAL PROBLEM')
WRITE(6,504)
WRITE(10,504)
504 FORMAT(/5X,'ENTER',2X,'1: YES',2X,'2: NO')
READ(5,*), NOYYSS
IF(NODYSS.EQ.1) THEN
IAM=2
CALL SENSTY(ISEN,IAM)
IAM=3
ELSE
IAM=3
ENDIF
ENDIF
CALL SENSTY(ISEN,IAM)
ISUMMY=3
CALL SUMMRY(IB,IN,NOPRO,IV,JZJJ,NOC,ISUMMY)
GO TO 502
C TO DETERMINE AN INTEGER PROCEDURE, CUTTING PLANE METHOD OR BRANCH
C AND BOUND TECHNIQUE
3003 JOYL=3
CALL MEMOH(JOYL,ISEN)
IF(IPART.EQ.1) GO TO 3000
IF(INTGP.EQ.1) THEN
IAM=2
CALL SENSTY(ISEN, IAM)
IF(BOUND.EQ.1 .AND. IPEMPT.EQ.1) THEN
CALL FACT1
GO TO 3001
ENDIF
C* TO DISPLAY MENU 2
3000 IF(BOUND.EQ.1 .AND. IPART.EQ.1) THEN
CALL FACT2
GO TO 3001
ENDIF
IF(BOUND.EQ.2) THEN
CALL PTRG(NOV)
IBUND=1
CALL BOUND
GO TO 3001
ENDIF
CONTINUE
STOP
END
SUBROUTINE INTENS(INTUR)
DIMENSION IBOR(100), IVZAR(100)
DIMENSION STOF(100, 100), SZV(100, 100), ISDD(100), SSIN(100)
DIMENSION TAB(100, 100), Z(100, 100), ID(100), IV(100)
DIMENSION IB(100), IREAL(50), IN(100), C(100, 100)
DIMENSION TABB(100, 100), LEVATT(50), ZZ(1, 100), IPM(1000)
DIMENSION LDECS(100), LPDEV(100), LNDEV(100)
REAL IV, IN, ISM, LEVATT
COMMON/B1/TAB, IV, ID, IB, LIT
COMMON/B2/Z, IN, IP, IC, IW
COMMON/B3/IPVC, XMAX, NOV, INQ
COMMON/B4/NPRNT, NDC, NPRO
COMMON/B5/NPRO, IPM, TABB, ZZ
COMMON/B6/IPEMPT, IPART, IBOUND
COMMON/B7/ICHECK, INTGP, IREAL, NXREAL
COMMON/B8/C
COMMON/B12/ITRATN, KINPRO, LEVATT
COMMON/B20/LDECS, LPDEV, LNDEV, LTOT1, LTOT2, LTOT3
COMMON/SS1/IBOR, IVZAR, STOF, ISDD, SZV, SSIN
GO TO (2, 4), INTUR
2 WRITE(6, 9)
WRITE(10, 9)
9 FORMAT(10X,'-->','ENTER 1 FOR PRINTING ALL TABLEAU'/ 
+13X,'ENTER 2 FOR PRINTING VARIABLES AND THEIR VALUES'/)
100 CONTINUE
READ(5.*,END=100) NPRNT
101 CONTINUE
WRITE(6,10)
WRITE(10,10)
10 FORMAT(5X,'-->',2X,'ENTER NO. OF CONTS.'/5X,'-->',2X, 
+ 'ENTER NO. OF VARIABLES'/5X,'-->',2X,'ENTER NO. OF PRIORITIES'/) 
READ(5.*,END=101) NDC,NOV,NPRO 
IF(IPEMPT.EQ.1) NPRO=NPRO 
IF(IPART.EQ.1) NPRO=1 
WRITE(6,20) NDC,NOV,NPRO 
WRITE(10,20) NDC,NOV,NPRO 
20 FORMAT(5X, 'NOC=',I2//5X, 'NOV=',I2//5X, 
+NPRO=',I2) 
WRITE(6,5) 
WRITE(10,5) 
READ(5.*) ICORR 
IF(ICORR.EQ.2) THEN 
WRITE(6,7) 
WRITE(10,7) 
GO TO 101 
ENDIF 
WRITE(6,21) 
WRITE(10,21) 
21 FORMAT(5X,'-->',2X,'ENTER THE NUMBER OF ALL VARIABLES' 
+/10X,'ENTER THE # OF POSITIVE DEVIATIONS'/10X,'ENTER THE # OF 
+ALL NEGATIVE DEVIATIONS') 
READ(5.*) NDS,LPD1,LND1 
WRITE(6,600) 
WRITE(10,600) 
READ(5.*) (LDECS(I),I=1,NDS) 
IF(LPD1.EQ.0) GO TO 151 
WRITE(6,601) 
WRITE(10,601) 
READ(5.*) (LPDEV(J),J=1,LPD1) 
151 IF(LND1.EQ.0) GO TO 152 
WRITE(6,602) 
WRITE(10,602) 
600 FORMAT(5X,'-->',2X,'ENTER THE LIST OF DECISION VARIABLES') 
601 FORMAT(5X,'-->',2X,'ENTER THE LIST OF POS-DEV VARIABLES') 
602 FORMAT(5X,'-->',2X,'ENTER THE LIST OF NEG-DEV VARIABLES') 
READ(5.*) (LNDEV(K),K=1,LND1) 
152 LDT1=NDS 
LDT2=LPD1 
LDT3=LND1
C
DO 22 I=1,NOV
ID(I)=I
ISDD(I)=I
22 CONTINUE
4 IF(INTGP.EQ.1) THEN
30 WRITE(6,170)
WRITE(10,170)
170 FORMAT(//10X,'--->',2X,'ENTER THE NUMBER OF INTEGER VARIABLES')
READ(5,*) NXREAL
WRITE(6,171) NXREAL
WRITE(10,171) NXREAL
171 FORMAT(//10X,'--->',2X,'ENTER',2X,I3,2X,'VARIABLES NAMES')
READ(5,*) (IREAL(I),I=1,NXREAL)
WRITE(6,605)
WRITE(10,605)
605 FORMAT(//20X,'LIST OF THE REQUIRED INTEGER VARIABLES')
DO 604 I=1,NXREAL
WRITE(6,603) IREAL(I)
WRITE(10,603) IREAL(I)
603 FORMAT(//20X,I5)
604 CONTINUE
WRITE(6,5)
WRITE(10,5)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,7)
WRITE(10,7)
GO TO 30
ENDIF
RETURN
ENDIF
DO 41 I=1,NOV
DO 41 J=1,NOV
STOF(I,J)=0.0
41 TAB(I,J)=0.0
WRITE(6,42)
WRITE(10,42)
42 FORMAT(5X,'-->',2X,'ENTER NUMBER OF NONZERO ELEMENTS IN' +/10X,'THE TECHNOLOGICAL MATRIX')
READ(5,*) NONZRO
WRITE(6,23)
WRITE(10,23)
23 FORMAT(5X,'-->',2X,'ENTER THE TECHNOLOGICAL COEFFICIENTS')
WRITE(6,31)
WRITE(10,31)
FORMAT(SX, '->', 2X, 'ENTER ROW I, COLUMN J AND THEN ITS'/' +10X, 'VALUE', 2X, 'LEAVE ONE SPACE BETWEEN ENTRIES')

CONTINUE
DO 43 I=1, NONZRO
READ(5, *, END=102) L, M, VALUE
WRITE(6, 32) L, M, VALUE
WRITE(10, 32) L, M, VALUE
WRITE(6, 5)
WRITE(10, 5)
READ(E, *) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6, 7)
WRITE(10, 7)
GO TO 33
ENDIF
TAB(L, M)=VALUE
STOF(L, M)=VALUE
CONTINUE
DO 40 I=1, NOPRO
DO 40 J=1, NOV
SZV(I, J)=0
Z(I, J)=0
WRITE(6, 50)
WRITE(10, 50)
FORMAT(SX, '->', 2X, 'ENTER THE RIGHT HAND SIDE VALUES')
CONTINUE
DO 60 I=1, NODC
READ(5, *, END=103) IV(I)
WRITE(6, 34) IV(I)
WRITE(10, 34) IV(I)
FORMAT(5X, 'RHS=', 2X, F9.4)
WRITE(6, 5)
WRITE(10, 5)
READ(E, *) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6, 7)
WRITE(10, 7)
GO TO 35
ENDIF
IVZAR(I)=IV(I)
WRITE(6, 70)
WRITE(10, 70)
FORMAT(SX, '->', 'ENTER THE INITIAL BASIC VARIABLES')
CONTINUE
DO 80 I=1, NODC
READ(5,*,END=104) IB(I)
WRITE(6,36) IB(I)
WRITE(10,36) IB(I)

FORMAT(5X,'BAISC VARIABLE=',2X,I4)
WRITE(6,5)
WRITE(10,5)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,7)
WRITE(10,7)
END IF
GO TO 37
ENDIF

IBOR(I)=IB(I)
CONTINUE
DO 1201 I=1,NOPRO
DO 1201 J=1,NOV
C(I,J)=0.
WRITE(6,1202)
WRITE(10,1202)
FORMAT(5X,'-->',2X,'ENTER THE NUMBER OF NONZERO ELEMENTS'
+/10X,'IN THE PRIORITY WEIGHT MATRIX')
READ(5,*,END=106) NZROP
WRITE(6,38)
WRITE(10,38)

FORMAT(5X,'-->',2X,'ENTER PRIORITY NUMBER I',VARIABLE '
+/10X,'THEN ITS PRIORITY WEIGHT.' ,2X,'LEAVE A SPACE BETWEEN ENTRIES')
DO 1203 I=1,NZROP
READ(5,*) J,KK,PL
WRITE(6,39) J,KK,PL
WRITE(10,39) J,KK,PL
FORMAT(10X,'PRIORITY#=',2X,I3,2X,'VARIABLE#=',2X,I3,2X,
+'PRIORITY WEIGHT=.',2X,F8.4)
WRITE(6,5)
WRITE(10,5)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,7)
WRITE(10,7)
GO TO 45
ENDIF
C(J,KK)=PL
IF(IHECK.EQ.1) RETURN
IROUTY=1
IF(IROUTY.EQ.1) GO TO 186
DO 1200 J=1,NOPRO
DO 1200 I=1,NOC
K = IB(I)
Z(J.K) = 0
1200 CONTINUE
JJ = IB(1)
JJ1 = JJ - 1
DO 150 I = 1, NOPRO
   DO 150 J = 1, JJ1
      SUM = 0
      ISM = 0
      DO 160 K = JJ, NOV
         KK = K - JJ1
         ISM = ISM + C(I, K) * IV(KK)
         SUM = SUM + C(I, K) * TAB(KK, J)
      160
      Z(I, J) = SUM - C(I, J)
      SZV(I, J) = Z(I, J)
   150
   IN(I) = ISM
   GO TO 1888
186 CONTINUE
   DO 180 I = 1, NOPRO
      DO 180 K = 1, NOV
         DO 182
            KK = 1, NOC
            IF(IB(KK).EQ.ID(K)) THEN
               Z(I, K) = 0.
               GO TO 180
            END IF
         182
         SUM = SUM + C(I, LOK) * TAB(KK, K)
       183
         ISM = ISM + C(I, LOK) * IV(KK)
         Z(I, K) = SUM - C(I, K)
      180
      IN(I) = ISM
188 FORMAT(2X, 'CORRECT', 5X, 'ENTER', 5X, '1: YES', 5X, '2: NO')
190 FORMAT(2X, 'REENTER AGAIN')
RETURN
END

SUBROUTINE PIVCOL
DIMENSION IN(100)
DIMENSION Z(100, 100), IPINK(100)
REAL IN, IV, LEVATT
COMMON/B2/Z, IN, IP, IC, IW
COMMON/B3/IPVC,XMAX,NOV,IND
COMMON/B19/ICOUNT,IFLAG
COMMON/B19/IBUND

C SAVE THE NAME OF ENTERING VARIABLE INTO THE BASIS
C

DO 5 I=1,NOV
   IPINK(I)=0
   IPVC=0
   XMAX=0
5   IF(IBUND.EQ.0) THEN
      DO 10 J=1,NOV
         IF(XMAX.GE.Z(IND,J)) GO TO 10
         XMAX=Z(IND,J)
         IPVC=J
      10 CONTINUE
   ELSE
      DO 20 J=1,NOV
         IF(Z(IND,J).LE.0) GO TO 20
         XMAX=Z(IND,J)
         IPVC=J
      20 CONTINUE
   END IF
   IPINK(IPVC)=IPVC
   END IF

IF(ABS(XMAX).LE.1E-4) XMAX=0.

60 IND1=IND1-1
IF(IND1.EQ.0) RETURN
IF(IPVC.EQ.0) RETURN
DO 40 I=1,IND1
   IF(Z(I,IPVC).LE.-9999.0) THEN
      IG=1
   GO TO 80
   ELSE
      IG=0
   ENDIF
40 CONTINUE
IF(IG.EQ.0) RETURN

C PURPOSE TO FIND A PIVOT COLUMN OR AN ENTERING VARIABLE
C WHICH DOES NOT DESTROYED THE PREVIOUS PRIORITY LEVELS.
C

80 IPV=0
   XMAX=0
   IF(IBUND.EQ.0) THEN
DO 50 J=1,NOV
IF(XMAX.GE.Z(INO,J)) GO TO 50
IF(J.EQ.IPINK(J)) GO TO 50
XMAX=Z(INO,J)
IPV=J
50 CONTINUE
ELSE
DO 85 J=1,NOV
IF(Z(INO,J).LE.O.) GO TO 85
IF(J.EQ.IPINK(J)) GO TO 85
XMAX=Z(INO,J)
IPV=J
GO TO 86
85 CONTINUE
ENDIF
86 IPVC=IPV
IF(IPVC.NE.O) THEN
IPINK(IPVC)=IPVC
ENDIF
IF(XMAX.EQ.O) GO TO 70
GO TO 60
70 RETURN
END

C**************************************************************************
C SUBROUTINE PIVROW
C**************************************************************************
SUBROUTINE PIVROW(IPROW)
DIMENSION TAB(100,100),IV(100)
DIMENSION IREAL(50),ID(100),IB(100)
REAL IN,IV,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B3/IPVC,XMAX,NOV,INO
COMMON/B4/NPRNT,NOC,NOPRO
COMMON/B7/ICHECK,INTGP,IREAL,NXREAL
COMMON/B15/IPAT,KRAZY
C
IF(INTGP.EQ.1) THEN
C CHECK FOR FEASIBILITY
KRAZY=0
DO 100 I=1,NOC
IF(IV(I).LT.O.) THEN
KRAZY=KRAZY+1
RETURN
ENDIF
100 CONTINUE
ENDIF

C TEST FOR THE PIVOT ROW

IPAT=0
IFSBL=0
IDGN=0
RAT=200000000.

310 DO 300 I=1,NOC
IF(TAB(I,IPVC)) 300,300,312
312 RATIO=IV(I)/TAB(I,IPVC)
IF(RAT.EQ.RATIO) IDGN=1
IF(RAT.LT.RATIO) GO TO 300
RAT=RATIO
IPROW=I
300 CONTINUE
DO 500 I = 1, NOC
IF(TAB(I,IPVC).LE.0.) THEN
IFSBL=IFSBL+1
ENDIF
500 CONTINUE
IF(IFSBL.EQ.NOC) THEN
IPAT=1
RETURN
ENDIF
RETURN
END

C***************************************************************************
C SUBROUTINE CALC
***************************************************************************
SUBROUTINE CALC(IPROW)
DIMENSION TAB(100,100),Z(100,100),IV(100),LEVATT(50),IN(100)
DIMENSION TT(200,101),IB(100),ID(100),B(200,101)
REAL IV,IN,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NOV,IND
COMMON/B4/NPRNT,NOC,NOPRO
COMMON/B6/IPEMPT,IPART,IBOUND
COMMON/B12/ITRATN,KINPRO,LEVATT
ITRATN=ITRATN+1

IB(IPROW)=ID(IPVC)
IP=NOC+1
IC=NOC+NOPRO
IW=NOV+1
LIT=0
LOOP=0
DO 5 K=1,NOC
  IF(IV(K).GE.O) LOOP=LOOP+1
DO 10 I=1,IC
DO 10 J=1,IW
10  TT(I,J)=O
DO 20 I=1,NOC
DO 20 J=1,NOV
20  TT(I,J)=TAB(I,J)
DO 30 I=1,NOC
30  TT(I,IW)=IV(I)
K=O
DO 40 I=IP,IC
K=K+1
DO 40 J=1,NOV
40  TT(I,J)=Z(K,J)
K=O
DO 50 I=IP,IC
K=K+1
50  TT(I,IW)=IN(K)
DO 313 I=1,IC
DO 314 J=1,IW
IF(I.EQ.IPROW) GO TO 315
IF(J.EQ.IPVC) GO TO 1111
B(I,J)=TT(I,J)-TT(IPROW,J)*TT(I,IPVC)/TT(IPROW,IPVC)
GO TO 314
315  B(I,J)=TT(I,J)/TT(IPROW,IPVC)
GO TO 314
1111  DO 1112 K=1,IW
IF(K.EQ.IPROW) GO TO 1113
B(K,J)=O
GO TO 1112
1113  B(IPROW,IPVC)=1.
1112  CONTINUE
314  CONTINUE
313  CONTINUE
DO 100 I=1,IC
DO 100 J=1,IW
100  IF(ABS(B(I,J)).LE.1E-4) B(I,J)=O.
DO 70 I=1,NOC
DO 70 J=1,NOV
70  TAB(I,J)=B(I,J)
DO 80 I=1,NOC
80  IV(I)=B(I,IW)
K=O
DO 90 I=IP,IC
K=K+1
DO 90 J=1,NOV
Z(K,J)=B(I,J)
IN(K)=B(I,IW)
IF(ABS(IN(K)) .LE. 0.01) IN(K)=0.
90 CONTINUE
RETURN
END

C**********************************************************************
C* SUBROUTINE PTRG
**********************************************************************
C
SUBROUTINE PTRG(NOV)
DIMENSION TAB(100, 100), IB(100), IV(100), ID(100), IN(100)
DIMENSION Z(100, 100), LEVATT(50), C(100, 100)
DIMENSION IPM(1000), TABB(100, 100), ZZ(1, 100)
REAL IN, IV, LEVATT
COMMON/B1/TAB, IV, ID, IB, LIT
COMMON/B2/Z, IN, IP, IC, IW
COMMON/B4/NPRNT, NDC, NOPRO
COMMON/B5/NPRQ, IPM, TABB, ZZ
COMMON/B6/PEMPT, IPART, IBOUND
COMMON/B12/ITRATN, KINPRO, LEVATT

WRITE(6,10) ITRATN
WRITE(10,10) ITRATN
10 FORMAT(//10X,'TABLE'/10X,'ITERATION',2X,I5)
WRITE(6,5)
WRITE(10,5)
5 FORMAT(//)
GO TO (1,2), NPRNT
1 WRITE(6,20)
WRITE(10,20)
20 FORMAT(10X,'BASIS',35X,'VARIABLES',40X,'VALUES')
WRITE(6,100)
WRITE(10,100)
WRITE(6,30) (ID(I), I=1, NOV)
WRITE(10,30) (ID(I), I=1, NOV)
30 FORMAT(4X,17(5X,I2))
WRITE(6,100)
WRITE(10,100)
100 FORMAT(10X,'----------------------------------------------------')
DO 50 I=1,NDC
WRITE(6,60) IB(I), (TAB(I,J), J=1, NOV), IV(I)
WRITE(10,60) IB(I), (TAB(I,J), J=1, NOV), IV(I)
60 FORMAT(//X,12,1X,5(F16.6,1X),F16.6)
50 CONTINUE
SUBROUTINE PARTG

DIMENSION TAB(100,100), IV(100), IB(100), ID(100), Z(100,100)
DIMENSION IN(100), IREAL(50), TAA(100,100)
DIMENSION IPM(1000), TABB(100,100), C(100,100), ZZ(1,100)
DIMENSION LEVATT(50), SAVE(50,100), CC(100,100)
DIMENSION LDECS(100), LPDEV(100), LNDEV(100)
REAL IN, IV, LEVATT
COMMON/B1/TAB, IV, ID, IB, LIT
COMMON/B2/Z, IN, IP, IC, IW
COMMON/B3/IPVC, XMAX, NOV, INO
COMMON/B4/NPRNT, NDC, NOPRO
COMMON/B5/NPRO, IPM, TABB, ZZ
COMMON/B6/IPEMP, IPART, IBOUND
COMMON/B7/ICHECK, INTGP, IREAL, NXREAL
COMMON/B8/C
COMMON/B10/KING
COMMON/B11/KAT, KIT
COMMON/B12/ITRATN, KINPRO, LEVATT
COMMON/B15/IPAT, KRAZY
COMMON/B17/ICALL, CC, SAVE, ISIGN
COMMON/B20/LDECS, LPDEV, LNDEV, LTOT1, LTOT2, LTOT3

C
WRITE(6, 10)
WRITE(10, 10)
10 FORMAT(/10X, 'ALGORITHM IS USING THE PARGP PROCEDURE')
ICHECK=0
ISIGN=0
KINPRO=1
INTUR=1
CALL INTERS (INTUR)
C
C TO SAVE THE VECTOR OF PRIORITY WEIGHTS
C
DO 18 I=1, 100
DO 18 J=1, 50
18 SAVE(J, I)=0.
DO 19 I=1, NOV
19 SAVE(1, I)=C(1, I)
ICALL=1
C** CALL FOR PIVOT COLUMN AND PIVOT ROW
40 INC=1
C TO COUNT THE NUMBER PRIORITIES
C
50 CALL PIVCOL
IF(XMAX.EQ.0) GO TO 20
IF(IPVC.EQ.0) GO TO 20
CALL PIVROW(IPROW)
CALL CALCI(IPROW)
C
IF(LIT.NE.O OR LIT.NE.NOC) GO TO 20
CALL PTRG(NOV)
GO TO 50
20 KIND=IND+1
100 IF(KIND.GT.NPRO) GO TO 30
LEVATT(KINPRO)=IN(1)
KINPRO=KINPRO+1
IOMID=1
CALL ACHECK(ICHECK, IOMID)
IF(ISIGN.EQ.1) RETURN
IF(ICHECK.EQ.1) THEN
CALL INILS(ICHECK, LX)
ICALL=ICALL+1
DO 17 I=1, NOV
SAVE(ICALL, I)=CC(1, I)
CONTINUE
ENDIF

CALL PIVCOL
IF(XMAX.EQ.0 OR IPVC.EQ.O) GO TO 80
CALL PIVROW(IPROW)
CALL CALC(IPROW)
CALL PTRG(NOV)
GO TO 60

80 KIND=KIND+1
GO TO 100

30 RETURN
END

C* SUBROUTINE ACHECK

SUBROUTINE ACHECK(ICHECK, IOMID)
DIMENSION TAB(100,100),IV(100),IB(100),ID(100),IN(100)
DIMENSION Z(100,100),ZZ(1,100),TA8B(100, 100),TAA(100,100)
DIMENSION IPM(1000),LEVATT(50),SAVE(50,100),IDD(100)
DIMENSION CC(100,100)
REAL IN,IV,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NOV,IND
COMMON/B4/NPRNT,NDC,NOPRO
COMMON/B5/NPRO,IPM,TABB,ZZ
COMMON/B6/IPEMPT,IPART,IBOUND
COMMON/B10/KING
COMMON/B11/KAT,KIT
COMMON/B12/ITRATN,KINPRO,LEVATT
COMMON/B17/ICALL,CC,SAVE,ISIGN
GO TO (1,2),IOMID

C** PURPOSE TO DETERMINE THE NUMBER OF ALTERNATIVE SOLUTIONS
1 K=0
DO 10 I=1,NOV
10 IF(Z(1,I).EQ.0.) K=K+1

C** TO CHECK FOR ALTERNATIVE SOLUTIONS
C ICHEK=1 INDICATES THAT THERE EXIST AT LEAST ONE ALTER SOLUTION.
NZRO=K-NOC
IF(NZRO.GE.1) THEN
ICHECK=1
ISIGN=0
ELSE
C THE PRESENT SOLUTION IS OPTIMAL FOR THE ORIGINAL PROBLEM WITH
C RESPECT TO ALL PRIORITIES
C
ICHECK=0
ISIGN=1
ENDIF

C** TO DETERMINE A NONBASIC COLUMN WITH NEGATIVE CRITERION COEFF.
2     K=0
     DO 20 I=1,NOV
20     IPM(I) INDICATES THE SPECIFIC NONBASIC COLUMN
     IPM(I)=0
     DO 30 J=1,NOC
30     IF(ID(I).EQ.IB(J)) GO TO 20
     IF(Z(1,I).LT.0.) THEN
     C** TO COUNT THE NUMBER OF NONBASIC COLUMNS
     K=K+1
     IPM(I)=I
     ENDIF
     20 CONTINUE
     NOVV=NCV-K
     DO 40 I=1,NOVV
     DO 40 J=1,NCC
        ZZ(1,I)=0.
40     TABB(I,J)=0.
     C*** KING COUNTS NUMBER OF VARIABLES
     C***
     IF(KAT.EQ.0) THEN
     KING=NOV
     ELSE
     KING=KING+KAT
     ENDIF
     IDID=0
     IDO=0
     DO 60 I=1,NOV
     IF(I.EQ.IPM(I)) GO TO 150
80     IDID=IDID+1
81     IDO=IDO+1
82     ZZ(1,IDO)=Z(1,IDID)
83     DO 70 J=1,NOC
84     TABB(J,IDO)=TABB(J,IDID)
85     IDD(IDO)=ID(IDID)
86     GO TO 60
87     150
88     IDID=IDID+1
89     60 CONTINUE
     C***
     NOV INDICATES THE NUMBER OF VARIABLES CONSIDERING
     C*** THE FACT THAT SOME OF THEM CAN BE ELEMINATED
     C*** DURING PREVIOUS ITERATIONS.
C***

NDV=IDO
DO 80 I=1,NDV
DO 90 J=1,NOC
90 TAB(J,I)=TABB(J,I)
   ID(I)=IDDI)
80 Z(1,I)=ZZ(1,I)
   CALL PTRG(NDV)
   IF(ICHECK.EQ.1.AND.ISIGN.EQ.0) RETURN
   IF(ICHECK.EQ.0.AND.ISIGN.EQ.1) THEN
DO 50 L,OLY=KINPRO,NPRO
   K=LOLY+1
   IN(K)=0.
   CALL INILS(ICHECK,LX)
   ICALL=ICALL+1
DO 9 I=1,NOC
   SAVE(ICALL,I)=CC(1,I)
9 CONTINUE
   DO 12 I=1,NOC
   DO 13 J=1,KIT
   IF(IB(I).EQ.ID(J)) THEN
   LOCT=IB(I)
   IN(K)=IN(K)+SAVE(K,LOCT)*IV(I)
   LEVATT(K)=IN(K)
   GO TO 12
   ENDIF
13 CONTINUE
12 CONTINUE
50 CONTINUE
   ENDIF
   RETURN
   END

C******************************************************
C* SUBROUTINE INILS
C******************************************************

SUBROUTINE INILS(ICHECK,LX)
DIMENSION TAB(100,100),IV(100),IB(100),ID(100),IN(100)
DIMENSION TAA(100,100),TABB(100,100),C(100,100)
DIMENSION LEVATT(50),Z(100,100),ZZ(1,100),IPM(1000)
DIMENSION TBC(100,100),SAVE(50,100),CC(100,100)
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NDV,IND
COMMON/B4/NPRNT,NOC,NPRO
COMMON/B5/NPRO,IPM,TABB,ZZ
COMMON/B6/IPEMP.T,IPART,IBOUND
COMMON/B6/C
COMMON/B9/NOC1,NOC2
COMMON/B10/KING
COMMON/B12/ITRATN,KINPRO,LEVATT
COMMON/B17/ICALL.CC,SAVE,ISIGN
REAL IN,IV,LEVATT
NOC1=NOC
CALL ADDUP(CC)

NOC2=NOC
NOC3=NOC2-NOC1
NNOC=NOC1+1
DO 5 NBC=1,NOC3
   DO 30 J=1,NOV
      DO 20 I=1,NOC1
         IF(IB(I).EQ.ID(J)) THEN
            L=I
            IF(TAB(NNOC,J).NE.O.) THEN
               IF(TAB(L,J).EQ.O.) GO TO 30
               DD=TAB(NNOC,J)/TAB(L,J)
               DO 10 K=1,NOV
                  TAA(L,K)=-DD*TAB(L,K)
                  MARY=NOV+1
                  TAA(L,MARY)=-DD*IV(L)
               TBC(NNOC,K)=TAB(NNOC,K)+TAA(L,K)
               TAB(NNOC,K)=TBC(NNOC,K)
               CONTINUE
            END IF
         END IF
      END DO
      IV(NNOC)=IV(NNOC)+TAA(L,MARY)
      GO TO 30
   END IF
   CONTINUE
30 CONTINUE
5 CONTINUE

NOC3=NOC2-NOC1
NNOC=NOC1+1
DO 6 NBC=1,NOC3
   IF(IV(NNOC).GE.O) GO TO 40
   KNNN=NNOC-1
   DO 31 J=1,NOV
      IF(TAB(NNOC,J).EQ.-1.) THEN
         K=O
      END IF
31 CONTINUE
40 CONTINUE
IF(TAB(I,J).EQ.0.) K=K+1
CONTINUE
IF(K.EQ.KNNN) THEN
IET=ID(J)
IYES=J
GO TO 34
ENDIF
ENDIF
31 CONTINUE
GO TO 40
34 DO 33 J=1,NOV
TAB(NNOC,J)=-TAB(NNOC,J)
IF(ABS(TAB(NNOC,J)).LE.0.0002) TAB(NNOC,J)=0.
33 CONTINUE
IPD=IB(NNOC)
IV(NNOC)=-IV(NNOC)
IF(ABS(IV(NNOC)).LE.0.0002) IV(NNOC)=0.
IB(NNOC)=IET
PJET=CC(1,IET)
CC(1,IET)=CC(1,IPD)
CC(1,IPD)=PJET
DO 42 KJ=1,NOV
42 IF(ID(KJ).EQ.IPD) INNO=KJ
POTT=C(1,IYES)
C(1,IYES)=C(1,INNO)
C(1,INNO)=POTT
40 NNOC=NNOC+1
6 CONTINUE
KING1=KING+1
DO 70 I=1,NOV
DO 80 J=1,NOC
IF(IB(J).EQ.ID(I)) THEN
Z(1,I)=0.
GO TO 70
END IF
80 CONTINUE
BSUM=0.
ASUM=0.
DO 90 J=1,NOC
K=IB(J)
ASUM=ASUM+CC(1,K)*TAB(J,I)
BSUM=BSUM+CC(1,K)*IV(J)
90 CONTINUE
Z(1,I)=ASUM-C(1,I)
IN(1)=BSUM
70 CONTINUE
CALL PTRG(NOV)
RETURN
END

SUBROUTINE ADDUP(CC)
DIMENSION TAB(100,100),Z(100,100),ID(100),IB(100)
DIMENSION IV(100),IN(100),C(100,100),TAB(100,100)
DIMENSION LEVATT(50),ZZ(1,100),IPM(1000)
DIMENSION CC(100,100)
DIMENSION ZXY(80,100)
REAL IN,IV,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NOV,IND
COMMON/B4/NPRNT,NOC,NOPRO
COMMON/B5/NPRO,IPM,TABB,ZZ
COMMON/B6/IPEMPT,IPART,IBOUND
COMMON/B8/C
COMMON/B10/KING
COMMON/B11/KAT,KIT
COMMON/B12/KINPRO,LEVATT
NOCP=NOC
NOVP=NOV+1
2
WRITE(6,1)
WRITE(10,1)
1 FORMAT(10X,'-->',2X,'ENTER THE NUMBER OF NEW CONST. AND'
+15X,'NUMBER OF NEW VARIABLES')
READ(5,*) KNOC,KNOV
KAT=KNOV
WRITE(6,133) KNOC,KNOV
WRITE(10,133) KNOC,KNOV
133 FORMAT(10X,'KNOC=',I2,5X,'KNOV=',I2)
WRITE(6,114)
WRITE(10,114)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,115)
WRITE(10,115)
ENDIF
IF(KNOV.EQ.0.AND.KNOC.EQ.0) GO TO 1079
C TO CONSIDER A SPECIAL SITUATION
C THIS IS THE CASE WHEN KNOV=0 AND KNOC IS GREATER ZERO
IF(KNOC.NE.0.AND.KNOC.EQ.0) THEN
NOC=NOC+KNOC
GO TO 2101
END IF
NOC=NOC+KNOC
INOV1=NOV+1
NOV=NOV+KNOV
K=KING
DO 10 I=1,KNOV
K=K+1
ID(INOV1)=K
IPM(K)=0
INOV1=INOV1+1
10 CONTINUE
C KIT COUNTES THE TOTAL NUMBER OF VARIABLES
KIT=KING+KNOV
DO 20 I=NDVP,NOV
DO 20 J=1,NOCP
20 TAB(J,I)=0.
C** PURPOSE TO READ THE VALUE OF EACH ELEMENT OF EACH CONST.
KING1=KING+1
2101 DO 2000 K=1,KNOC
DO 2000 I=1,KIT
2000 ZXY(K,I)=C.
C
C WRITE(6,2007)
WRITE(10,2007)
2007 FORMAT(5X,'--->',2X,'ENTER # OF NONZERO ELEMENTS IN THE'
+/10X,'NEW CONSTRAINTS')
C
READ(5,*) NNZRO
WRITE(6,3) NNZRO
WRITE(10,3) NNZRO
3 FORMAT(' # OF NONZERO ELEMENTS = ',5X,I5)
WRITE(6,4)
WRITE(10,4)
4 FORMAT(5X,'--->',2X,'ENTER ROW I , COLUMN J AND ITS VALUE'/10X
+', 'LEAVE ONE SPACE BETWEEN THE ENTRIES')
DO 2008 I=1,NNZRO
5 READ(5,*) L,M,VALUE
WRITE(6,116) L,M,VALUE
WRITE(10,116) L,M,VALUE
WRITE(6,114)
WRITE(10,114)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,115)
WRITE(10,115)
GO TO 5
ENDIF
ISOR=L-NOCP
ZXY(ISOR,M)=VALUE
DO 2003 J=1,NOV
IDONE=ID(J)
DO 2004 N=1,KIT
IF(IDONE.EQ.N) THEN
DO 2001 K=1,KNOC
L=K+NOCP
TAB(L,J)=ZXY(K,N)
2001 CONTINUE
GO TO 2003
ENDIF
2004 CONTINUE
2003 CONTINUE
C** PURPOSE TO READ THE BASIS AND RIGHT HAND SIDE VALUES OF NEW CONSTRAINTES
WRITE(6,6)
WRITE(10,6)
6 FORMAT(5X,'-->',2X,'ENTER ROW I, COLUMN J AND THE RHS VALUES')
DO 33339 I=1,KNOC
7 READ(5,*) L,IBB,VIV
WRITE(6,117)
WRITE(10,117)
117 FORMAT('ROW I =',2X,I3,2X,'BASIS=',2X,13,2X,'RHS=',2X,F8.4)
WRITE(6,114)
WRITE(10,114)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,115)
WRITE(10,115)
GO TO 7
ENDIF
IV(L)=VIV
IB(L)=IBB
33339 CONTINUE
C***
KING1=KING+1
1079 WRITE(6,2009)
WRITE(10,2009)
2009 FORMAT(5X,'-->',2X,'ENTER # OF NONZERO ELEMENTS IN THE')
C*** SUBROUTINE DUALS
C***
C*** SUBROUTINE DUALSX(IPROW,LAB,IBALL)
DIMENSION TAB(100,100),IV(100),IB(100),ID(100)
DIMENSION LEVATT(50),Z(100,100),ZZ(1,100),B(200,101)
DIMENSION TT(100,100),IPM(1000),TABB(100,100),IN(100)
DIMENSION ISVZ(100),PRLV(50,60)
REAL IN,IV,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NOV,IND
COMMON/B4/NPRNT,NOC,NOPRC
COMMON/B5/NPRO,IPM,TABB,ZZ
COMMON/B6/IPEMPT,IPART,ISBOUND
COMMON/B9/NOC1,NOC2
COMMON/B12/ITRATN,KINPRO,LEVATT

C** TO FIND THE PIVOT ROW
LAB=0
IBALL=0
40
AMOST=0.
DO 10 I=1,NOC2
IF(IV(I).GE.AMOST) GO TO 10
I1=I
AMOST=IV(I1)
10 CONTINUE
IF(AMOST.EQ.0.) RETURN
R=1.E+10
IPROW=I1

C** TO FIND THE PIVOT COLUMN
IF(IPART.EQ.1) THEN
J1=0
DO 20 J=1,NOV
IF(TAB(I1,J).GE.O.) GO TO 20
IF(Z(I1,J).GE.O.) GO TO 20
RR=Z(I1,J)/TAB(I1,J)
IF(RR.LT.R) J1=J
IF(RR.LT.R) R=RR
20 CONTINUE
IPVC=J1
IF(IPVC.EQ.0) THEN
IBALL=1
RETURN
ENDIF
ENDIF

C CONSIDER A SITUATION WHEN PREEMPTIVE GOAL PROGRAMMING IS CONCERNED
C
BF(IPEMPT.EQ.1) THEN
IKOT=0
DO 61 KL=1,NOV
IF(TAB(I1,KL).GE.O.) GO TO 61
IKOT=IKOT+1
ISVZ(IKOT)=KL
DO 60 K=1,NDPRO
   IF(Z(K,KL).GE.O) THEN
      PRLV(K,IKOT)=0.
      GO TO 60
   ENDIF
   PRLV(K,IKOT)=ABS(Z(K,KL)/TAB(I1,KL))
60  CONTINUE
61  CONTINUE
   I=1
   L=2
   IPVC=0
90  IF(L.GT.IKOT) GO TO 92
   DO 91 J=1,NOPRO
      IF(PRLV(J,I).EQ.PRLV(J,L)) GO TO 91
      IF(PRLV(J,I).LT.PRLV(J,L)) THEN
         ILL=I
         I=ILL
         L=L+1
         IPVC=ISVZ(I)
         GO TO 90
      ELSE
         ILL=L
         I=ILL
         IPVC=ISVZ(L)
         L=L+1
         GO TO 90
      ENDIF
91  CONTINUE
END IF
92  CALL CALC(IPROW)
CALL PTRG(NOV)
C** CHECK FOR OTHER NEGATIVE RIGHT HAND SIDE VALUES
   J=0
   DO 30 I=1,NOC
      IF(IV(I).GE.O) J=J+1
      IF(J.EQ.NOC) THEN
         DO 51 INO=1,NDPRO
            CALL PIVCDL
            IF(XMAX.EQ.O.OR.IPVC.EQ.O) GO TO 52
            CALL PIVROW(IPROW)
            CALL CALC(IPROW)
            CALL PTRG(NOV)
            IF(INO.EQ.1) GO TO 52
51   CONTINUE
52   LAB=1
GO TO 50
ENDIF
GO TO 40
50 RETURN
END

C******************************************************************************
C* SUBROUTINE FACT1
C******************************************************************************
C**
SUBROUTINE FACT1
DIMENSION IB(100),IREAL(50),IV(100),IN(100)
DIMENSION DUM(100),IPM(1000),TABB(100,100),Z(100,100)
DIMENSION ID(100),ZZ(1,100),LEVATT(50),TAB(100,100)
DIMENSION ZPR0(100,100),INPR0(100),ZPNEW(100,100)
DIMENSION LDECS(100),LPDEV(100),LNDEV(100)
REAL IN,IV,INPRO,LEVATT
INTEGER ZROCUT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NDV,IND
COMMON/B4/NPRNT,NOC,NPRO
COMMON/B5/NPRO,IPM,TABB,ZZ
COMMON/B6/IPEMPT,IPART,IBOUND
COMMON/B7/ICHECK,INTGP,IREAL,NXREAL
COMMON/B10/KING
COMMON/B11/KAT,KIT
COMMON/B12/ITRATN,KINPRO,LEVATT
COMMON/B13/ICOUNT,IFLAG
COMMON/B14/ZPRO,NPRO,KPRIOR,ZPNEW,MVARR,MROWSS
COMMON/B15/IPAT,KRAZY
COMMON/B20/LDECS,LPDEV,LNDEV,LTOT1,LTOT2,LTOT3

C C PURPOSE TO CONTROL THE PROGRAM FOR LIPREGP PROBLEM
C
IFLAG=1
MROWSS=NOC
MVARR=NOV
KPRIOR=NPRO+1
C MOVE THE FIRST PRIORITY LEVEL INTO THE SECOND LEVEL IN
C ORDER TO SAVE A POSITION FOR THE NEW ABSOLUTE PRIORITY
C LEVEL
C
K=1
DO 101 I=1,NPRO
K=K+1
INPRO(K)=IN(I)
DO 102 J=1,NOV
ZPRD(K,J)=0.0
ZPRD(K,J)=Z(I,J)
102 CONTINUE
101 CONTINUE
IDEN=NOV-NXREAL
IDECID=0
IF(IDEN.GE.1) IDECID=1
C TO KEEP THE NUMBER OF EXISTING PRIORITY LEVELS
21 L=K
CALL INTGR(L,IDECID,ZROCUT)
IF(ZROCUT.EQ.1) GO TO 2
GO TO (1,2),ICOUNT
C TO START FROM THE FIRST PRIORITY LEVEL
1 IND=1
DO 5 IN0=1,L
10 CALL PIVCOL
IF(XMAX.EQ.0) GO TO 5
IF(IPVC.EQ.0) GO TO 5
CALL PIVROW(IPROW)
CALL CALC(IPROW)
IF(ITRATN.EQ.200) RETURN
CALL PTRGCNOV)
IF(INO.EQ.1) GO TO 10
IF(IN(INO).EQ.0) GO TO 5
5 CONTINUE
GO TO 21
2 ISUMMY=3
CALL SUMMRY(IB.IN,N0P0,IV,JZJJ,N0C.ISUMMY)
RETURN
END
SUBROUTINE FACT2
DIMENSION I8(100),IREAL(50),IV(100),LEVATT(50),IN(100)
DIMENSION DUM(100),IPM(1000),TABB(100,100) ,Z(100,100)
DIMENSION TAB(100,100),ZPRD(100,100),ID(100),ZZ(1,100)
DIMENSION LDECS(100),LPDEV(100),LNDEV(100)
DIMENSION INPRO(100),ZPNEW(100,100)
REAL IN.IV.INPRO.LEVATT
INTEGER ZROCUT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPCVC,XMAX,NOV,INO
COMMON/B4/NPRNT,NOC,NPRO
COMMON/B5/NPRO,IPM,TABE,ZZ
COMMON/B6/IPEMPT,IPART,IBOUND
COMMON/B7/ICHECK,INTGP,IREAL,NXREAL
COMMON/B10/KING
COMMON/B11/KAT,KIT
COMMON/B12/ITRATN,KINPRO,LEVATT
COMMON/B13/ICOUNT,IFLAG
COMMON/B14/ZPRO,INPRO,KPRIOR,ZPNEW,MVARR,MROWSS
COMMON/B15/IPAT,KRAZY
COMMON/B20/LDECS,LPDEV,LNDEV,LTOT1,LTOT2,LTOT3

C
C PURPOSE TO CONTROL THE PROGRAM FOR LIPARGP PROBLEM
C
IFLAG=1
MROWSS=NOC
KPRIOR=NPRO=1
IOMID=2
CALL ACHECK(ICHECK,IOMID)
MVARR=NOC
K=2
INPRO(K)=IN(1)
DO 19 J=1,100
  ZPRO(K,J)=0.0
19  CONTINUE
DO 22 J=1,NOC
  ZPRO(K,J)=Z(1,J)
22  CONTINUE

C TO DETERMINE THE TYPE OF THE CUTTING PLANE THAT SHOULD BE USED
IDEN=NOC-NXREAL
IDECID=0
IF(IDEN.GE.1) IDECID=1

C TO KEEP THE NUMBER OF EXISTING PRIORITY LEVELS
C
L=K
CALL INTGR(L,IDECID,ZROCUT)
IF(ZROCUT.EQ.1) GO TO 2
GO TO (1,2),ICOUNT
1  IND=1
DO 5 IND=1,L
10  CALL PIVCOL
  IF(XMAX.EQ.0.OR.IPVC.EQ.0) GO TO 5
  CALL PIVROW(IPROW)
  NOPRO=2
  CALL CALC(IPROW)
  IF(ITRATN.EQ.20) RETURN
  CALL PTRG(NOV)

5  CONTINUE
GO TO (1,2),ICOUNT
IF(INO.EQ.1) GO TO 10

CONTINUE

GO TO 21

ISUMMY=3

CALL SUMMRY(IB,IN,NPRO,IV,IZJU,NDC,ISUMMY)

RETURN

END

C********************************************************************
C                  SUBROUTINE INTGR
C********************************************************************

SUBROUTINE INTGR(L,IDECID,ZROCUT)

DIMENSION IB(100),IV(100),IN(100)
DIMENSION ZPRO(100,100),INPRO(100)
DIMENSION ZPNEW(100,100),IREAL(50)
DIMENSION DUM(100),IPM(1000),LEVATT(50),ZZ(1,100)
DIMENSION TABB(100,100),Z(100,100),ID(100),TAB(100,100)
REAL IN,IV,INPRO,LEVATT
INTEGER ZROCUT

COMMON/81/TAB,IV,ID,IB,LIT
COMMON/82/Z,IN,IP,IC,IW
COMMON/83/IPVC,XMAX,NOV,IND
COMMON/84/NPRNT,NDC,NPRO
COMMON/85/NPRO,IPM,IV,JZJU,NOC
COMMON/86/IZJU,TA88,IZJU
COMMON/87/ICHECK,INTGP,IREAL,NXREAL
COMMON/812/ITRATN,KINPRO,LEVATT
COMMON/813/ICOUNT,IFLAG
COMMON/814/ZPRO,INPRO,KPRIOR.ZPNEW,MVARR,MROWSS

C** PURPOSE TO DETERMINE THE MAXIMUM FRACTION OF THE
C** RIGHT HAND SIDE.
C**
IF(NXREAL.NE.0) THEN
CALL GOMORY(J8,KROWS,FMAX)
IF(JB.EQ.1) THEN
WRITE(6,106)
WRITE(10,106)
106 FORMAT(/10X,'THE REQUIRED VARIABLES ARE INTEGERE VALUED')
ICOUNT=2
RETURN
ELSE
 IROW=KROWS
GO TO 107
ENDIF
ENDIF
FMAX=0.0
DO 11 I=1,NDC

DO 10 J=1,MVARR
IF(IPEMPT.EQ.1) THEN
IF(IB(I).NE.J) GO TO 10
GO TO 3
ENDIF
C USING PARTITIONNING GOAL PROGRAMMING PROCEDURE
IF(IPART.EQ.1) THEN
LOCID=ID(J)
IF(IB(I).NE.LOCID) GO TO 10
GO TO 3
ENDIF
3
IRHS=IV(I)
FRACT=IV(I)-IRHS
IF(FRACT.LT.0.98) GO TO 1
IV(I)=IV(I)-FRACT+1.0
FRACT=0.0
IF(FMAX-FRACT) 5,11,11
5
FMAX=FRACT
IROW=I
GO TO 11
10
CONTINUE
11
CONTINUE
107
IF(FMAX.LE.0.1) THEN
ICOUNT=2
IFLAG=2
RETURN
ENDIF
C**
C** TO DEVELOP THE GOMORY CUTTING PLANE CONSTRAINT
C**
IF(IDECID.EQ.0) THEN
DO 25 J=1,NDV
ITT=TAB(IROW,J)
FRACT=TAB(IROW,J)-ITT
IF(ALS(FRACT).LE.0.00001) GO TO 20
IF(FRACT.GE.0.0) GO TO 15
DUM(J)=1.0+FRACT
GO TO 25
15
DUM(J)=FRACT
GO TO 25
20
DUM(J)=0.0
25
CONTINUE
ELSE
C PURPOSE TO DETERMINE A GOMORY CUTTING PLANE CONSTRAINT
C FOR THE MIXED INTEGER VALUES.
C
WRITE(6,1004)
WRITE(10,1004)
1004 FORMAT(//20X,’*** PROGRAM IS USING MIXED INTEGER PROC’)
DO 1000 K=1,NOV
KABA=0
DO 1001 J=1,NXREAL
1001 IF(IREAL(J).EQ.K) KABA=KABA+1
C
C IF(KABA.EQ.1) GO TO 1002
C
C IF(TAB(IROW,K).GE.0) THEN
DUM(K)=TAB(IROW,K)
ELSE
DUM(K)=(FMAX/(FMAX-1.))*TAB(IROW,K)
ENDIF
GO TO 1000
C FOR INTEGER VALUES
1002 ITT=TAB(IROW,K)
FRACT=IT-TAB(IROW,K)=ITT
IF(ABS(FRACT).LE.FMAX) THEN
DUM(K)=FRACT
ELSE
DUM(K)=(FMAX/(1.-FMAX))*(1.-FRACT)
ENDIF
1000 CONTINUE
ENDIF
C CHECK FOR ZERO-CUTTING PLANE. SUCH CUT MAY EXIST WHEN PARTITION
C ING GP PROCEDURE IS USED TO SOLVE A SMALL SIZED PROBLEM.
C
ZROCUT=0
ICOTT=0
DO 260 ILL=1,NOV
IF(DUM(ILL)) 260.27.260
27 ICOTT=ICOTT+1
260 CONTINUE
IF(ICOTT.EQ.NOV) THEN
ZROCUT=1
WRITE(6,28)
WRITE(10,28)
28 FORMAT(//10X,’A ZERO-CUT IS DETECTED’)
IF(IPART.EQ.1) THEN
WRITE(6,29)
WRITE(10,29)
29 FORMAT(//10X,’TRY ANOTHER PROCEDURE’/10X,’USE PREEMPTIVE
IF(IPEMPT.EQ.1) THEN
WRITE(6,30)
WRITE(10,30)
FORMAT(//10X,'TRY ANOTHER PROCEDURE'/10X,'USE PARTITIONING GP PROCEDURE')
ENDIF
RETURN
ENDIF

C** ADD CUTTING PLANE FRACTION TO THE FINAL TABLEAU
C USING PREEMPTIVE GOAL PROGRAMMING PROCEDURE
C
IF(IPEMPT.EQ.1) THEN
NVAR1=NOV+1
NVAR=NOV+2
NCONS=NOC+1
ID(NVAR1)=NVAR1
ID(NVAR)=NVAR
IB(NCONS)=NVAR
DO 70 I=1,NOC
DO 70 J=NVAR1,NVAR
70 TAB(I,J)=0.0
DO 71 I=2,L
DO 71 J=NVAR1,NVAR
71 Z(I,J)=0.0
TAB(NCONS,NVAR1)=-1.0
TAB(NCONS,NVAR)=1.0
DO 90 J=1,NOV
90 TAB(NCONS,J)=DUM(J)
IV(NCONS)=FMAX
GO TO 26
ENDIF
C USING PARTITIONNING GOAL PROGRAMMING PROCEDURE
C
IF(IPART.EQ.1) THEN
LOCT=ID(NOV)
NVAR1=LOCT+1
NVAR2=LOCT+2
NCONS=NOC+1
NVV1=NOV+1
NVV2=NOV+2
ID(NVV1)=NVAR1
ID(NVV2)=NVAR2
IB(NCONS)=NVAR2
DO 81 I=1,NOC
DO 81 J=NVV1,NVV2
81 TAB(I,J)=0.O
DO 83 I=2,L
DO 83 J=NVV1,NVV2
83 Z(I,J)=0.O
TAB(NCONS,NVV1)=-1.
TAB(NCONS,NVV2)=1.
C TO ORGANIZE THE NEW ROW CONSTRAINT FOR THE SIMPLEX TABLEAU
DO 82 J=1,NOV
82 TAB(NCONS,J)=DUM(J)
IV(NCONS)=FMAX
NVAR=NVV2
GO TO 26
ENDIF
C**ADD A NEW PRIORITY TO THE PRIORITY MATRIX
26 CONTINUE
C USING PREEMPTIVE GOAL PROGRAMMING PROCEDURE
IF(IPEMPT.EQ.1) THEN
DO 91 I=1,NVAR
DO 95 J=1,NCONS
IF(IB(J).EQ.I) THEN
ZPRO(1,I)=0.O
GO TO 91
ELSE
ZPRO(1,I)=TAB(NCONS,I)
ENDIF
95 CONTINUE
91 CONTINUE
DO 92 I=2,L
DO 92 J=NVAR1,NVAR
92 ZPRO(I,J)=0.O
ZPRO(1,NVAR1)=-1.
ZPRO(1,NVAR)=0.O
GO TO 94
ENDIF
IF(IPART.EQ.1) THEN
DO 191 I=1,NVV2
DO 192 J=1,NCDNS
IF(IB(J).EQ.ID(I)) THEN
ZPRO(1,I)=0.O
GO TO 191
ELSE
ZPRO(1,I)=TAB(NCONS.I)
ENDIF
192 CONTINUE
CONTINUE
ZPR0(1,NV1)=-1.
ZPR0(1,NV2)=0.0
ZPR0(2,NV1)=0.0
ZPR0(2,NV2)=0.0
ENDIF

**TO DETERMINE THE VALUE OF THE NEW PRIORITY LEVEL**

CONTINUE
ICARE=2
CALL SADD(IFLAG,NCONS,ICARE,LINE)

**TO UPDATE NUMBER OF CONSTRAINTS, VARIABLES AND PRIORITIES.**

NOV=NVAR
NOC=NCONS
NPRO=KPRIOR
NOPRO=NPRO
IF(IFLAG.EQ.2) THEN
DO 1103 J=1,NOV
Z(1,J)=ZPR0(1,J)
IN(1)=INPR0(1)
1103 CONTINUE

SET THE Z VALUES OF THE NEW VARIABLES TO ZERO
IF(IPART.EQ.1) THEN
Z(2,NV1)=0.0
Z(2,NV2)=0.0
ENDIF

**TO USING PREEMPTIVE GP PROCEDURE**

IF(IPEMPT.EQ.1) THEN
Z(2,NVAR1)=0.0
Z(2,NVAR)=0.0
ENDIF

**TO USING PARTITIONNING GP PROCEDURE**

IF(IPART.EQ.1) THEN
NOPRO=2
ENDIF
CALL PTRG(NOV)
RETURN
ENDIF
C USING PREEMPTIVE GP PROCEDURE
C
IF(IPEMPT.EQ.1) THEN
CALL PTRG(NOV)
ENDIF
RETURN
END
C******************************************************************************
C* SUBROUTINE SADD *
C******************************************************************************
SUBROUTINE SADD(IFLAG,NCONS,ICARE,LINEn)
DIMENSION TAB(100,100),IV(100),IB(100),ID(100),IN(100)
DIMENSION Z(100,100),ZPNEW(100,100),ZPRO(100,100),INPROD(100)
DIMENSION TABB(100,100),IPM(1000),LEVATT(50),ZZ(1,100)
REAL IN,IV,INPRO,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAK,NOV,INO
COMMON/B4/NPRNT,NOC,NOPRO
COMMON/B5/NPRD,IPM,TABB,ZZ
COMMON/B6/IPEMPT,IPART,ICARE
COMMON/B12/ITRAAN,KINPRO,LEVATT
COMMON/B14/ZPRO,INPRO,KPRIOR,ZPNEW,MVAR,MROWSS
GO TO (1,2),ICARE
1 IFLAG=1
IF(IPART.EQ.1) THEN
K=2
INPRO(K)=IN(1)
DO 10 J=1,100
ZPRO(K,J)=0.0
10 CONTINUE
GO TO 50
ENDIF
IF(IPEMPT.EQ.1) THEN
K=1
KAKE=NPR0-1
DO 30 I=1,KAKE
INPRO(K)=IN(I)
K=K+1
INPRO(K)=IN(I)
DO 40 J=1,NOV
ZPRO(K,J)=Z(1,J)
30 CONTINUE
GO TO 50
ENDIF
IF(IPART.EQ.1) THEN
K=2
INPRO(K)=IN(1)
DO 10 J=1,100
ZPRO(K,J)=0.0
10 CONTINUE
GO TO 50
ENDIF
IF(IPEMPT.EQ.1) THEN
K=1
KAKE=NPR0-1
DO 30 I=1,KAKE
INPRO(K)=IN(I)
K=K+1
INPRO(K)=IN(I)
DO 40 J=1,NOV
ZPRO(K,J)=Z(1,J)
ZPRO(K,J)=Z(I,J)
40 CONTINUE
30 CONTINUE
ENDIF
50 LINE=K
RETURN
2 BSUM=IV(NCONS)
DO 70 I=1,NOV
DO 80 J=1,NCONS
IF(IB(J).EQ.ID(I)) THEN
ZPNEW(1,I)=0.0
GO TO 70
ENDIF
80 CONTINUE
ZPNEW(1,I)=TAB(NCONS,I)
70 CONTINUE
IF(IFLAG.EQ.1) THEN
DO 100 I = 1, NOV
ZPRO(1,I)=ZPNEW(1,I)
100 CONTINUE
INPRO(1)=BSUM
RETURN
ENDIF
DO 101 I=1,NOV
ZPRO(1,I)=ZPNEW(1,I)+Z(1,I)
101 CONTINUE
INPRO(1)=IN(1)+BSUM
RETURN
END

**************************************************************************

SUBROUTINE GOMORY(JB,KROWS,FMAX)
DIMENSION ID(100),INPRO(100),TAB(100,100),Z(100,100)
DIMENSION IREAL(50),IV(100),IB(100),IN(100),ZPNEW(100,100)
DIMENSION IPM(1000),TABB(100,100),ZZ(1,100),ZPRD(100,100)
DIMENSION LEVATT(50)
REAL IN,IV,INPRO,LEVATT
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NOV,IND
COMMON/B4/NPRNT,NOG,NPROP
COMMON/B5/NPROP,IPM,TABB,ZZ
COMMON/B6/IPMPT,IPART,IBOUND
COMMON/B7/ICHECK,INTGP,IREAL,NXREAL
COMMON/B12/TRATN,KIPRO,LEVATT
COMMON/B13/ICOUNT,IFLAG
COMMON/B14/ZPRO,INPRO,KPRIOR,ZNUEW,MVARR,MROWSS
J=0
K=0
C COUNT THE NUMBER OF INTEGER VARIABLES WHICH ARE AVAILABLE.
DO 17 I=1,NXREAL
DO 18 J=1,NOC
IF(IREAL(I).EQ.IB(J)) THEN
IRHS=IV(J)
FRACT=IV(J)-IRHS
IF(FRACT.LE.0.02.OR.FRACT.GE.0.98) THEN
KD=KD+1
GO TO 17
ENDIF
18 CONTINUE
17 CONTINUE
IF(KD.LT.NXREAL) GO TO 12
JB=1
RETURN
C TO FIND THE MAXIMUM FRACTION FOR THE REQUIRED INTEGER VARIABLES
12 FMAX=0.0
DO 10 I=1,NOC
DO 11 J=1,NXREAL
IF(IB(I).NE.IREAL(J)) GO TO 11
IRHS=IV(I)
FRACT=IV(I)-IRHS
IF(FRACT.LE.99) GO TO 1
IV(I)=IV(I)-FRACT+1.
FRACT=0.0
1 IF(FMAX-FRACT) 5,10,10
5 FMAX=FRACT
KROWS=I
GO TO 10
11 CONTINUE
10 CONTINUE
RETURN
END
***********************
C* Subroutine TSORT *
C*************************
DIMENSION LEVATT(50), IPM(1000), KBASE(50), NONBAS(50), IREAL(50)
REAL IN, IV, ID, IB, LIT
COMMON/B1/TAB, IV, ID, IB, LIT
COMMON/B2/Z, IN, IP, IC, IW
COMMON/B3/IPVC, XMAX, NOV, INO
COMMON/B4/NPRNT, NOC, NOPRO
COMMON/B5/NPRG, IPM, TAB, ZZ
COMMON/B6/IPEMPT, IPART, IBOUND
COMMON/B7/ICHECK, INTGP, IREAL, NXREAL
COMMON/B8/C
COMMON/B10/KING
COMMON/B11/KAT, KIT
COMMON/B12/ITRTN, KINPRO, LEVATT
COMMON/B13/ICOUNT, IFLAG
COMMON/B15/IPAT, KRAZY
C SAVE THE FOLLOWING VECTORS FOR THE REQUIRED INTEGER VARIABLES
DO 10 I = 1, NXREAL
  KBASE(I) = 0
  NONBAS(I) = 0
10 CONTINUE
C THE FOLLOWING SECTION SORT THE BASIC AND NONBASIC VARIABLES
C WHOSE VALUE REQUIRED TO INTEGERED
C TO RECORD THE NONINTEGER BASIC VARIABLES
LOW = 0
DO 30 I = 1, NXREAL
  DO 20 J = 1, NOC
    IF(IREAL(I).EQ.IB(J)) THEN
      LOW = LOW + 1
      KBASE(LOW) = IB(J)
    END IF
 20 CONTINUE
30 CONTINUE
C RECORD THE LIST OF NONBASIC VARIABLES TO BE INTEGERED.
C
LL = 0
DO 40 I = 1, NXREAL
  DO 50 J = 1, LOW
    IF(IREAL(I).EQ.KBASE(J)) THEN
      IZZ = 0
    GO TO 40
      ELSE
      IZZ = 1
    END IF
40 CONTINUE
IF(IZZ.EQ.1) THEN
    LL=LL+1
    NONBAS(LL)=IREAL(I)
ENDIF
40 CONTINUE
RETURN
END

SUBROUTINE BOUND
IMPLICIT NONE

DIMENSION ATAB(100,100),AZE(100,100),AIV(100),AIN(100)
DIMENSION IABASE(100),IDVAR(100),ATT(100,100),AZZ(100,100)
DIMENSION KBB(100),RHV(100),ARHN(100),KDD(100),IN(100)
DIMENSION TAB(100,100),Z(100,100),IB(100),ID(100),IV(100)
DIMENSION TABB(100,100),ZZ(1,100),INPR0(100),XDOM(100)
DIMENSION ZPROD(100,100),XMOD(100),IPM(100),IREAL(50)
DIMENSION F(50),FF(50),ZPNEW(100,100),LEVATT(50)
DIMENSION TUT(100,100),ZUZ(100,100),IUN(100),IUB(100)
DIMENSION IUD(100),IUV(100),XARRY(200),XMOD(100),ISET(100)
DIMENSION LDECS(100),LPDEV(100),LNDEV(100)
REAL IN,IV,IUV,IUN,IPM,LEVATT
INTEGER ZROCUT,SETNOV,SETNOC,SETLL,SUTLL
COMMON/B1/TAB,IV,ID,IB,LIT
COMMON/B2/Z,IN,IP,IC,IW
COMMON/B3/IPVC,XMAX,NOV,INC
COMMON/B4/NPRNT,NOC,NPRO
COMMON/B5/NPRO,IPM,TABB,ZZ
COMMON/B6/IMEPT,IPART,IBOUND
COMMON/B7/ICHECK,INTGP,IREAL,NXREAL
COMMON/B10/KING
COMMON/B12/ITRATN,INPRO,LEVATT
COMMON/B13/ICDUNT,IFLAG
COMMON/B14/ZPRO,INPRO,KPRIOR,ZPNEW,MVARR,MROWSS
COMMON/B15/IPAT,KRAZY
COMMON/B16/ATAB,AZE,AIV,AIBASE,IDVAR,XDOM,XMOD
COMMON/B18/IOUE,SETNOV,SETNOC,SETLL
COMMON/B20/LDECS,LPDEV,LNDEV,LTOT1,LTOT2,LTOT3
NUM8ER=0
IQUE=0
JZJJ=0
NIBB=1
IF(IPART.EQ.1) THEN
  IFLAG=1
MVARR=NOV
MROWSS=NOC
KPRIOR=NPRO+1
MVARR=NOV
K=2
INPRO(K)=IN(1)
DO 19 J=1,100
19  ZPRO(K,J)=0.0
DO 22 J=1,NOV
ZPRO(K,J)=Z(I,J)
22  CONTINUE
GO TO 21
END IF
IF(IPEMPT.EQ.1) THEN
IFLAG=1
MROWSS=NOC
MVARR=NOV
KPRIOR=NPRO+1
K=1
DO 101 I=1,NPRO
101  K=K+1
INPRO(K)=IN(I)
DO 102 J=1,NOV
ZPRO(K,J)=0.
ZPRO(K,J)=Z(I,J)
102  CONTINUE
101  CONTINUE
GO TO 21
END IF
L=K
C TO CONSTRUCT TWO NEW CONSTRAINTES
C
IBR=1
IQUE=IQUE+1
WRITE(6,2222)
WRITE(10,2222)
2222  FORMAT(//10X,'USING IBR=1')
SETNOV=NOV
SETNOC=NOC
SETLL=L
CALL BRNCH(IBR,L,FMAX,JB,ZROCUT,IROW)
IF(ZROCUT.EQ.1) RETURN
IF(JB.EQ.1) THEN
ISUMMY=1
CALL SUMMRY(IB,IN,NPRO,IV,UZIJU,NOC,ISUMMY)
IF(NUMBER.EQ.1) GO TO 1
GO TO 92
ENDIF
GO TO (1,2),ICOUNT
2
IND=1
NIBR=IBR
DO 13 IND=1,L
DO 600 LOT=1,NOC
IF(IV(LOT).LT.O.) GO TO 601
600 CONTINUE
10 CALL PIVCOL
IF(XMAX.EQ.0.OR.IPVC.EQ.0) GO TO 5
CALL PIVROW(IPROW)
IF(IPART.EQ.1) NPRO=2
CALL CALG(IPROW)
IBR=3
CALL BRNCH(IBR,L,FMAX,JB,ZROCUT,IROW)
IF(JB.EQ.1) THEN
ISUMMV=1
CALL SUMNNY(IB,IN,NOPRO,IV,JZJJ,NOC,ISUMMV)
CALL PTRG(NOV)
GO TO 602
ENDIF
5 IF(IN(2).EQ.0) GO TO 602
13 CONTINUE
602 IBR=NIBR
IF(IBR.EQ.2) GO TO 1000
DO 2111 I=1,NOC
DO 222 J=1,NOV
ATT(I,J)=TAB(I,J)
CONTINUE

RHV(I)=IV(I)
KBB(I)=IB(I)

CONTINUE

DO 23 I=1,NOPRO
DO 24 J=1,NOV
AZZ(I,J)=Z(I,J)
KD(I,J)=ID(J)

CONTINUE

ARMN(I)=IN(I)

CONTINUE

NVARL=NOV
KNNOC=NOC
SUTLL=L

SAVE THE VALUE OF PRIORITY OF ONE AND TWO

DO 9 I=1,SUTLL
F(I)=IN(I)

PREPARE TO SOLVE THE SECOND PROBLEM

IBR=2
IQUE=IQUE+1
WRITE(6,22222)
WRITE(10,22222)

FORMAT(/10X,'USING IBR=2')
NOV=SETNOV
NOC=SETNOC
LNEW=SETL
IF(IQUE.EQ.2) LNEW=SETL-1
DO 11 I=1,NOC
DO 20 J=1,SETNOV
TAB(I,J)=ATAB(I,J)

CONTINUE

IV(I)=AIV(I)
IB(I)=IABASE(I)

CONTINUE

SET NOPRO EQUAL TO L

DO 30 I=1,LNEW
DO 40 J=1,SETNOV
Z(I,J)=AZE(I,J)
ID(J)=IDVAR(J)

CONTINUE

IN(I)=AIN(I)

CONTINUE
IF(IQUE.EQ.2) THEN
  NOPRO=L-1
  CALL PTRG(NOV)
  ICARE=1
  CALL SADD(IFLAG,NCONS,ICARE,LINE)
  K=LINE
ELSE
  CALL PTRG(NOV)
ENDIF

C STORE XDOM IN XMOD
DO 50 J=1,SETNOV
  XDOM(J)=0.
  XDOM(J)=XMOD(J)
50 CONTINUE

C FMAX=1.-FMAX
CALL BRNCH(1BR,L,FMAX,JB,ZROCUT,1ROW)
IF(JB.EQ.1.0R.ZROCUT.EQ.1) GO TO 1
GO TO 2
1000 CONTINUE
IF(IPART.EQ.1) NPR0=2
DO 8 I=1,NPRO
  FF(I)=IN(I)
8 CONTINUE

C PREPARE TO SOLVE THE PARTITIONNING PROBLEM
IF(IPART.EQ.1) THEN
  IF(F(1).EQ.O.AND.F(2).GT.O) THEN
    IF(FF(1).EQ.O.AND.FF(2).EQ.O) GO TO 21
    ENDIF
  IF(F(1).EQ.O.AND.F(2).GT.O) THEN
    IF(F(1).EQ.O.AND.F(2).EQ.O) GO TO 57
    ENDIF
  IF(F(1).GT.O.AND.FF(1).EQ.O) GO TO 21
  IF(F(1).EQ.O.AND.FF(1).GT.O) GO TO 57
  NPR0=2
  GO TO 58
ENDIF

C COMPARE THE SOLUTION OF THESE TOW PROBLEMS (FOR PREEMPTIVE)
IF(F(2).GT.O.AND.FF(2).GT.O) GO TO 77
C TERMINATION RULE FOR SUB #1
IF(F(2).GT.O.AND.FF(2).LE.O) GO TO 56
C TERMINATION RULE FOR SUB #2
IF(F(2).LE.O.AND.FF(2).GT.O) GO TO 57
GO TO 58
C CONTINUE BRANCHING FROM SUB #2
56    CALL PTRG(NOV)
      GO TO 21
C CONTINUE BRANCHING FROM SUB #1
57    NDV=NVARL
      NDC=KNNGC
      DO 51 I=1,NDC
      DO 52 J=1,NDV
         TAB(I,J)=ATT(I,J)
      CONTINUE
      IV(I)=RHV(I)
      IB(I)=KBB(I)
51    CONTINUE
C SET NOPRO TO L
      DO 53 I=1,SUTLL
      DO 54 J=1,NOV
         Z(I,J)=AZZ(I,J)
      CONTINUE
      IN(I)=ARHN(I)
53    CONTINUE
      CALL PTRG(NOV)
      IF(ITRATN.GT.100) RETURN
      GO TO 21
58    IAIA=0
      DO 55 KL=1,NPRO
         IF(F(KL).EQ.0.AND.FFlKL).EQ.0) THEN
            IAIA=IAIA+1
            IF(IAIA.EQ.NPRO) GO TO 71
            GO TO 55
         END IF
         IF(F(KL).LT.FF(KL)) THEN
            IF(JB.EQ.1.AND.IQUE.EQ.2) GO TO 68
            GO TO 65
         END IF
      END DO
55    CONTINUE
C SAVE THE TABLEAUE OF SUB #2
65    NIBB=2
      KVV1=NOV
      KCC1=NDC
      DO 66 I=1,KCC1
      DO 67 J=1,KVV1
         TUT(I,J)=TAB(I,J)
      CONTINUE
      IUU(I)=IV(I)
      IUB(I)=IB(I)
66    CONTINUE
      DO 69 I=1,L
DO 70 J=1,NOV
  ZUZ(I,J)=Z(I,J)
  IUD(J)=ID(J)
  CONTINUE
69  INU(I)=IN(I)
C TO USE TABLE OF SUB#1 FOR FURTHER COMPUTATIONS
68  NOV=NVARL
    NDC=KNDOC
    DO 61 I=1,NDC
    DO 62 J=1,NOV
      TAB(I,J)=ATT(I,J)
      IV(I)=RHV(I)
      IB(I)=KBB(I)
    61    CONTINUE
    DO 63 I=1,SUTLL
      DO 64 J=1,NOV
        Z(I,J)=AZZ(I,J)
        ID(J)=KDD(J)
      64    CONTINUE
      IN(I)=ARHN(I)
    63    CONTINUE
    GO TO 76
ELSE
    IF(IQUE.EQ.2) GO TO 71
    GO TO 76
71  NIBB=2
    KVV1=NVARL
    KCC1=KNDOC
    DO 72 I=1,KCC1
      DO 73 J=1,KVV1
        TUT(I,J)=ATT(I,J)
        IUV(I)=RHV(I)
        IUB(I)=KBB(I)
      73    CONTINUE
      DO 74 J=1,L
        ZUZ(I,J)=AZZ(I,J)
        IUD(J)=KDD(J)
      74    CONTINUE
      INU(I)=ARHN(I)
    74 CONTINUE
    GO TO 76
ENDIF
55 CONTINUE
76 CALL PTRG(NOV)
GO TO 21
IF(IQUE.EQ.2) THEN
WRITE(6,91)
WRITE(10,91)
FORMAT(/10X,'This problem has no integer solution')
GO TO 1001
ENDIF

NUMBER=1
IF(NIBB.EQ.2) THEN
NDV=KVV1
NDC=KCC1
DO 78 I=1,NDC
DO 79 J=1,NDV
TAB(I,J)=TUT(I,J)
IV(I)=IUV(I)
IB(I)=IUB(I)
DO 80 I=1,NOPRO
DO 81 J=1,NOV
Z(I,J)=ZUZ(I,J)
ID(J)=IUD(J)
CONTINUE
IN(I)=IUN(I)
80 CONTINUE
CALL PTRG(NOV)
GO TO 21
ENDIF

1 IF(NUMBER.EQ.0) GO TO 92
1001 ISUMM=2
CALL SUMMRY(IB,IN,NOPRO,IV,JZJJ,NOC,ISUMM)
ISUMM=3
CALL SUMMRY(IB,IN,NOPRO,IV,JZJJ,NOC,ISUMM)
RETURN
END

C--------------------------------------------------------------------------------
C* SUBROUTINE BRNCH
C--------------------------------------------------------------------------------

SUBROUTINE BRNCH(IBR,L,FMAX,JB,ZROCUT,IROW)
DIMENSION TAB(100,100),Z(100,100),IB(100),ID(100),IN(100)
DIMENSION IV(100),INPRO(100),ZPRO(100,100),IPM(1000)
DIMENSION ZPNEW(100,100),XDOM(100),XMOD(100),ZZ(1,100)
DIMENSION ATAB(100,100),AIV(100),AIN(100),AZE(100,100)
DIMENSION IABASE(100),IDVAB(100),ATT(100,100),AZZ(100,100)
DIMENSION KBB(100),RHV(100),ARHN(100),TABB(100,100)
DIMENSION LEVATT(50),IREAL(50),KBABSE(50),NDBAS(50)
REAL IN,IV,INPRO,LEVATT
INTEGER ZROCUT,SETNOV,SETNOC,SETLL,SUTL
COMMON/B1/TAB,IV,ID,IB,LI
CALL TSORT(KBASE, NONBAS, LL, LOW)

IFFECT=0
ZROCUT=0
JB=0
KD=0
DO 10 I=1, LOW
DO 9 J=1, NOC
IF(KBASE(I).EQ.IB(J)) THEN
IRHS=IV(J)
FRACT=IV(J)-IRHS
IF(FRACT.LE.0.02.0R.FRACT.GE.0.98) THEN
KD=KD+1
GO TO 10
ENDIF
ENDIF

10 CONTINUE
KD=KD+LL
IF(KD.LT.NXREAL) THEN
GO TO (20, 2, 400).IBM
400 RETURN
ELSE
JB=1
WRITE(6, 40)
40 FORMAT(/15X, 'ALL REQUIRED VARIABLES ARE INTEGER')
RETURN
ENDIF

20 FMAX=0.0

C IF(IFFECT.EQ.1) THEN
DO 300 I=1, NOC
DO 301 J=1, LOW
IF(IB(I).NE.KBASE(J)) GO TO 301
300 CONTINUE
301 CONTINUE
IF (IV(I).EQ.0.) GO TO 301
IRHS=IV(I)
FRACT=IV(I)-IRHS
IF (FRACT.LT.0.5) THEN
FRAKSN=FRACT/IV(I)
ELSE
FRAKSN=(1.-FRACT)/IV(I)
ENDIF
IF (FMAX-FRAKSN) 302, 300, 300
302 FMAX=FRAKSN
IROW=I
GO TO 300
301 CONTINUE
300 CONTINUE
ELSE
DO 30 I=1,NOC
DO 11 J=1,LOW
IF (IB(I).NE.KBASE(J)) GO TO 11
IRHS=IV(I)
FRACT=IV(I)-IRHS
IF (FRACT.LE.0.99) GO TO 111
IV(I)=IV(I)-FRACT+1.
FRACT=0.0
111 IF (FMAX-FRACT) 5, 30, 30
5 FMAX=FRACT
IROW=I
GO TO 30
11 CONTINUE
30 CONTINUE
ENDIF
C
IF (FMAX.LE.0.01) THEN
ICOUNT=1
RETURN
ENDIF
C
C TO DEVELOPE TWO NEW CONSTRAINTS
C
IPROW=IROW
DD 25 J=1, NDV
IF (ID(J).EQ.IB(IPROW)) GO TO 21
XDDM(J)=TAB(IPROW.J)
IF (XDDM(J).EQ.0) THEN
XMOD(J)=XDDM(J)
ELSE
\[ X_{\text{MOD}}(J) = -X_{\text{DOM}}(J) \]

ENDIF

GO TO 25

21  \[ X_{\text{DOM}}(J) = 0.0 \]

\[ X_{\text{MOD}}(J) = 0.0 \]

25  CONTINUE

WRITE(6,110) \((X_{\text{DOM}}(J), J=1, \text{NOV})\)

WRITE(10,110) \((X_{\text{DOM}}(J), J=1, \text{NOV})\)

WRITE(6,110) \((X_{\text{MOD}}(J), J=1, \text{NOV})\)

WRITE(10,110) \((X_{\text{MOD}}(J), J=1, \text{NOV})\)

110  FORMAT(\\2X,10(F8.5,2X))

C CHECK FOR ZERO-CUTTING PLANES

ZROCUT = 0

ICOTT = 0

DO 26 ILL = 1, NOV

IF \((X_{\text{DOM}}(ILL)) \neq 0.0\) 26, 27, 26

27  \[ ICOTT = ICOTT + 1 \]

26  CONTINUE

IF \((ICOTT = \text{NOV})\) THEN

ZROCUT = 1

WRITE(6,28)

WRITE(10,28)

28  FORMAT(\\10X,'A ZERO-CUT IS DETECTED')

IF \((\text{IPART} \neq 1)\) THEN

WRITE(6,29)

WRITE(10,29)

29  FORMAT(\\10X,'TRY ANOTHER PROCEDURE'/\\10X,'USE PREEMPTIVE GP PROCEDURE')

END IF

C PURPOSE TO CONSIDER THE PREEMPTIVE GOAL PROGRAMMING PROCEDURE

IF \((\text{IPEMPT} \neq 1)\) THEN

WRITE(6,24)

WRITE(10,24)

24  FORMAT(\\10X,'TRY ANOTHER PROCEDURE'/\\10X,'USE PARTITIONING GP PROCEDURE')

END IF

RETURN

END IF

C SAVE THE TABLEAU, BASIS, RHS, DECISION VARIABLES, AND MATRIX OF PRIORITY WEIGHTS

DO 70 I = 1, NOC

DO 80 J = 1, NOV

ATAB(I,J) = TAB(I,J)

80  CONTINUE

\[ AIV(I) = IV(I) \]

\[ IABASE(I) = IB(I) \]
CONTINUE
DO 81 I=1,NPRO
DO 82 J=1,NDV
AZE(I,J)=Z(I,J)
IDVAR(J)=ID(J)
82 CONTINUE
CONTINUE
AIN(I)=IN(I)
81 CONTINUE

C ADD THE CUTTING PLANES INTO THE TABLE
C
2 IF(IPEMPT.EQ.1) THEN
NVAR1=NDV+1
NVAR=NDV+2
NCONS=NOC+1
ID(NVAR1)=NVAR1
ID(NVAR)=NVAR
IB(NCONS)=NVAR
DO 90 I=1,NDC
DO 90 J=NVAR1,NVAR
90 TAB(I,J)=O.
DO 92 I=2,L
DO 92 J=NVAR1,NVAR
92 Z(I,J)=O.
TAB(NCONS,NVAR1)=-1.
TAB(NCONS,NVAR)=1.
DO 93 J=1,NOV
93 TAB(NCONS,J)=XDOM(J)
IV(NCONS)=FMAX
ENDIF

C ADD A NEW PRIORITY TO THE PRIORITY MATRIX
C
IF(IPEMPT.EQ.1) THEN
DO 94 I=1,NVAR
DO 95 J=1,NCONS
IF(IB(J).EQ.1) THEN
ZPRO(1,I)=0.0
GO TO 94
ELSE
ZPRO(1,I)=TAB(NCONS,I)
ENDIF
95 CONTINUE
94 CONTINUE
DO 96 I=2,L
DO 96 J=NVAR1,NVAR
ZPR0(I,J)=0.
ZPR0(1,NVAR1)=-1.
ZPR0(1,NVAR)=0.
GO TO 194
ENDIF

C FOR PARTITIONING PROCEDURE
C
IF(IPART.EQ.1) THEN
LOCT=ID(NOV)
NVAR1=LOCT+1
NVAR2=LOCT+2
NCONS=NOC+1
NVV1=NOV+1
NVV2=NOV+2
ID(NVV1)=NVAR1
ID(NVV2)=NVAR2
IB(NCONS)=NVAR2
DO 101 I=1,NOC
DO 101 J=NVV1,NVV2
101 TAB(I,J)=0.
DO 102 J=NVV1,NVV2
102 Z(2,J)=0.
TAB(NCONS,NVV1)=-1.
TAB(NCONS,NVV2)=1.
DO 103 J=1,NOV
103 TAB(NCONS,J)=XDOM(J)
IV(NCONS)=FMAX
NVAR=NVV2
ENDIF
IF(IPART.EQ.1) THEN
DO 104 I=1,NVV2
DO 105 J=1,NCONS
IF(IB(J).EQ.ID(I)) THEN
ZPR0(1,I)=0.
GO TO 104
ELSE
ZPR0(1,I)=TAB(NCONS,I)
ENDIF
105 CONTINUE
104 CONTINUE
ZPR0(1,NVV1)=-1.
ZPR0(1,NVV2)=0.
ZPR0(2,NVV1)=0.
ZPR0(2,NVV2)=0.
ENDIF
C CONTINUE
C UPDATE NUMBER OF CONSTRAINTS, VARIABLES, AND PRIORITIES
C
NOV=NVAR
NOC=NCONS
NPRO=KPRIDR
NOPRO=NPRO
BSUM=IV(NCONS)
INPRO(1)=BSUM
DO 107 J=1,NOV
  Z(1,J)=ZPRO(1,J)
IN(1)=INPRO(1)
107 CONTINUE

C IF(IFLAG.EQ.1) THEN
  IF(IQUE.LE.2) THEN
    DO 108 I=2,L
      DO 109 J=1,NOV
        Z(I,J)=ZPRO(I,J)
109 CONTINUE
      IN(I)=INPRO(I)
    108 CONTINUE
  ENDIF
  ICOUNT=2
  IF(IPART.EQ.1) THEN
    NOPRO=2
    CALL PTRG(NOV)
    RETURN
  ENDIF
  IF(IPEMPT.EQ.1) THEN
    CALL PTRG(NOV)
  ENDIF
  RETURN
END

C---------------------------------------------------------------
C* SUBROUTINE SUMMRY *
C---------------------------------------------------------------
SUBROUTINE SUMMRY(IB,IN,NOPRO,IV,JZJU,NOC,ISUMMY)
C
DIMENSION IB(100),IV(100),SARRY(200),XARRY(200),ISETT(100)
DIMENSION LDECS(100),LPDEV(100),LNDEV(100)
DIMENSION IN(100),BINE(100),KBUT(50)
REAL IV, IN
INTEGER XARRY
COMMON/B20/LDECS,LPDEV,LNDEV,LTOT1,LTOT2,LTOT3

C
GO TO (83,82,999),ISUMMY
C
to store the integer solutions
C
83      JZJJ=JZJJ+1
      IF(JZJJ.EQ.1) THEN
      DD 84 J=1,NOC
      SARRY(J)=IV(J)
      XARRY(J)=IB(J)
      ISETT(JZJJ)=NOC
      DD 60 L=1,NOPRO
      BINE(L)=IN(L)
      KBUT(JZJJ)=NOPRO
      ENDIF
      IF(JZJJ.GE.2) THEN
      MAN1=1
      KBBC=JZJJ-1
      DD 70 I=1,KBBC
      MAN1=MAN1+ISETT(I)
      MAN2=MAN1+NOCT-1
      K=0
      DD 91 J=MAN1,MAN2
      K=K+1
      SARRY(J)=IV(K)
      XARRY(J)=IB(K)
      ISETT(JZJJ)=NOC
      MAN3=1
      LAM=JZJJ-1
      DD 65 L=1,LAM
      MAN3=MAN3+KBUT(L)
      MAN4=MAN3+NOPRO-1
      LK=0
      DD 66 J=MAN3,MAN4
      LK=LK+1
      BINE(J)=IN(LK)
      KBUT(JZJJ)=NOPRO
      ENDIF
      RETURN
C
82      CONTINUE
C
to summarize the integer solutions
C
      IF(JZJJ.EQ.0) THEN
      WRITE(6,90)
WRITE(10,90)
WRITE(6,92)
WRITE(10,92)

92 FORMAT(//20X, 'THIS PROBLEM HAS NO INTEGER SOLUTION')
RETURN
ENDIF
WRITE(6,90)
WRITE(10,90)

90 FORMAT(//20X, 'SUMMARY OF INTEGER SOLUTION')
JET1=1
JET3=1
DO 85 I=1,JZJJ
JET2=ISETT(I)
JET=JET1+JET2-1
WRITE(6,87)
WRITE(10,87)

87 FORMAT(//20X, 'VARIABLES', 11X, 'VALUE')
DO 86 II=JET1,JET
WRITE(6,89) XARRY(II),SARRY(II)
WRITE(10,89) XARRY(II),SARRY(II)

86 CONTINUE
JET1=JET+1
JET4=KBUT(I)
JET5=JET3+JET4-1
WRITE(6,67)
WRITE(10,67)

67 FORMAT(//20X, 'PRIORITY', 11X, 'VALUE')
IOT=O
DO 68 JJ=JET3,JET5
IOT=IOT+1
WRITE(6,69) IOT,BINE(JJ)
WRITE(10,69) IOT,BINE(JJ)

68 CONTINUE
JET3=JET5+1

85 CONTINUE
RETURN

C

999 WRITE(6,109)
WRITE(10,109)
WRITE(6,101)
WRITE(10,101)

101 FORMAT(//10X, 'OPTIMAL SOLUTION FOR ORIGINAL DECIS. VARIABLES')
DO 102 J=1,LTOT1
IBIB=O
DO 103 I = 1, NOC
IF(IB(I).EQ.LDECS(J)) THEN
WRITE(6, 104) J, IV(I)
WRITE(10, 104) J, IV(I)
104 FORMAT(/10X,'X('',12.1X,')=',6X,F16.6)
IBIB=1
GO TO 102
ENDIF
103 CONTINUE
IF(IBIB.EQ.0) THEN
WRITE(6, 105) J
WRITE(10, 105) J
105 FORMAT(/I10X,'X('',12.1X,')=',16X,'0.0000')
ENDIF
102 CONTINUE
WRITE(6, 109)
WRITE(10, 109)
109 FORMAT(/10X,'************************************************************')
WRITE(6, 15550)
WRITE(10, 15550)
15550 FORMAT(/20X,'OVERACHIVEMENTS')
DO 106 J=1, LTDT2
IAIA=0
DO 107 I = 1, NOC
IF(IB(I).EQ.LPDEV(J)) THEN
WRITE(6, 108) J, IV(I)
WRITE(10, 108) J, IV(I)
108 FORMAT(/10X,'D+(',12,1X,')=',6X,F16.6)
IAIA=1
GO TO 106
ENDIF
107 CONTINUE
IF(IAIA.EQ.0) THEN
WRITE(6, 110) J
WRITE(10, 110) J
110 FORMAT(/10X,'D+(',12,1X,')=',16X,'0.0000')
ENDIF
106 CONTINUE
WRITE(6, 109)
WRITE(10, 109)
WRITE(6, 111)
WRITE(10, 111)
111 FORMAT(/20X,'UNDERACHIVEMENT')
DO 112 J = 1, LTDT3
ICIC=0

DO 113 I=1,NOC
  IF(IB(I).EQ.LNDEV(J)) THEN
    WRITE(6,115) J,IV(I)
    WRITE(10,115) J,IV(I)
  115 FORMAT(//1X, 'D-(', I2, 1X, ')=' ,6X,F16.6)
  ICIC=1
  GO TO 112
END IF
113 CONTINUE
  IF(ICIC.EQ.0) THEN
    WRITE(6,114) J
    WRITE(10,114) J
  114 FORMAT(//1X, 'D-(', I2, 1X, ')=' ,16X, '0.0000')
END IF
112 CONTINUE
WRITE(6,109)
WRITE(10,109)
RETURN
END

C**********************************************************************
C SUBROUTINE MEMOH
C**********************************************************************
SUBROUTINE MEMOH(JOYL,ISEN)

DIMENSION IREAL(50)
COMMON/B6/IPEMPT,IPART,IBOUND
COMMON/B7/ICHECK,INTGP,IREAL,NXREAL

GO TO (1,2,4),JOYL

C DISPLAY OF MENU 1

1    WRITE(6,7)
    WRITE(10,7)
  7 FORMAT(15X,'DISPLAY OF MENU 1 ')
    WRITE(6,10)
    WRITE(10,10)
  10 FORMAT(5X,'CONTINUOUS SOLUTION BY PREGP PROCEDURE')
    WRITE(6,11)
    WRITE(10,11)
  11 FORMAT(5X,'***ENTER 1 ***')
    WRITE(6,20)
    WRITE(10,20)
  20 FORMAT(5X,'CONTINUOUS SOLUTION BY PARGP PROCEDURE')
    WRITE(6,21)
    WRITE(10,21)
21 FORMAT(5X,'*** ENTER 2 ***')
IF(JOYL.EQ.1) GO TO 3
C TO DISPLAY OF MENU 3
4 WRITE(6,8)
WRITE(10,8)
8 FORMAT(15X,'DISPLAY OF MENU 3')
WRITE(6,40)
WRITE(10,40)
40 FORMAT(5X, 'INTEGER SOLUTION BY PREGP USING CUTTING PLANE')
WRITE(6,41)
WRITE(10,41)
41 FORMAT(5X,'*** ENTER 3 ***')
WRITE(6,50)
WRITE(10,50)
50 FORMAT(5X,'INTEGER SOLUTION BY PREGP USING B & B')
WRITE(6,51)
WRITE(10,51)
51 FORMAT(5X,'*** ENTER 4 ***')
WRITE(6,60)
WRITE(10,60)
60 FORMAT(5X, 'INTEGER SOLUTION BY PARGP USING CUTTING PLANE')
WRITE(6,61)
WRITE(10,61)
61 FORMAT(5X,'*** ENTER 5 ***')
WRITE(6,68)
WRITE(10,68)
68 FORMAT(5X, 'FIND INTEGER SOLUTION BY PARGP USING B & B')
WRITE(6,69)
WRITE(10,69)
69 FORMAT(5X, '*** ENTER 6 ***')
C WRITE(6,65)
WRITE(10,65)
65 FORMAT(5X, 'TO KEEP THE CONTINUOUS SOLUTION')
WRITE(6,66)
WRITE(10,66)
66 FORMAT(5X, '*** ENTER 7 ***')
3 WRITE(6,9)
WRITE(10,9)
9 FORMAT(15X, '*** CHOOSE THE OPTION ***')
READ (5,*) MOO
IF(MOO.EQ.7) RETURN
IF(MOO.EQ.1) THEN
WRITE(6,70)
WRITE(10,70)
IPART=0
IPEMPT=1
INTGP=C
IBOUND=0
GO TO 999
ENDIF
IF(MOO.EQ.2) THEN
WRITE(6, 71)
WRITE(10, 71)
IPART=1
IPEMPT=0
INTGP=0
IBOUND=0
GO TO 999
ENDIF
IF(MOO.EQ.3) THEN
WRITE(6,90)
WRITE(10,90)
IPEMPT=1
IPART=0
INTGP=1
IBOUND=1
GO TO 999
ENDIF
IF(MOO.EQ.4) THEN
WRITE(6,91)
WRITE(10,91)
IPEMPT=1
IPART=0
INTGP=1
IBOUND=2
GO TO 999
ENDIF
IF(MOO.EQ.5) THEN
WRITE(6,92)
WRITE(10,92)
IPEMPT=1
IPART=1
INTGP=1
IBOUND=1
GO TO 999
ENDIF
IF(MOO.EQ.6) THEN
WRITE(6,93)
WRITE(10,93)
IPART=1

IPEMPT=0
INTGP=1
IBOUND=2
ENDIF
70 FORMAT(//SX,'A CONTINUOUS SOLUTION BY PREGP IS REQUESTED')
71 FORMAT(//SX,'A CONTINUOUS SOLUTION BY PARGP IS REQUESTED')
90 FORMAT(//SX,'LIPREGP AND GOMORY'S CP METHOD IS SELECTED')
91 FORMAT(//SX,'LIPREGP AND BRANCH AND BOUND METHOD IS SELECTED')
92 FORMAT(//SX,'LIPARGP AND GOMORY'S CP METHOD IS SELECTED')
93 FORMAT(//SX,'LIPARGP AND BRANCH AND BOUND METHOD IS SELECTED')
999 IF(INTGP.EQ.1) THEN
   INTUR=2
   CALL INTERS(INTUR)
ENDIF
RETURN
C
C DISPLAY OF MENU 2
C
2 WRITE(6,94)
   WRITE(10,94)
94 FORMAT(15X,*** DISPLAY OF MENU 2 ***)
    WRITE(6,700)
    WRITE(10,700)
700 FORMAT(25X,*** MENU FOR SENSITIVITY ANALYSIS ***)
299 FORMAT(5X,'TO DO NO CHANGES ENTER 5')
    WRITE(6,100)
    WRITE(10,100)
100 FORMAT(5X,'CHANGE THE RHS VALUES')
    WRITE(6,101)
    WRITE(10,101)
101 FORMAT(5X,'** ENTER 1 **')
    WRITE(6,102)
    WRITE(10,102)
102 FORMAT(5X,'TO ADD A NEW DECISION VARIABLE')
    WRITE(6,103)
    WRITE(10,103)
103 FORMAT(5X,'*** ENTER 2 ***')
    WRITE(6,104)
    WRITE(10,104)
104 FORMAT(5X,'TO ADD A NEW OBJECTIVE FUNCTION')
    WRITE(6,105)
    WRITE(10,105)
105 FORMAT(5X,'** ENTER 3 **')
    WRITE(6,106)
    WRITE(10,106)
106 FORMAT(5X,'TO CHANGE THE COEFFICIENT ASSOCIATED WITH THE
SUBROUTINE BINVRS(SENS, ISKIL)
DIMENSION IVZAR(100), STOF(100, 100), ISDD(100), SZV(100, 100)
DIMENSION ID(100), IBOR(100), TAB(100, 100), SENS(100, 100)
DIMENSION IV(100), IB(100), SSIN(100)
COMMON/B1/TAB, IV, ID, IB, LIT
COMMON/B3/IVZAR, XMAX, NOV, INO
COMMON/B4/NPRNT, NDC, NPRO
COMMON/SS1/IBOR, IVZAR, STOF, ISDD, SZV, SSIN
REAL IV
C PURPOSE TO DETERMINE THE MATRIX OF B INVERSE
K=0
DO 10 L=1, NDC
DO 20 J=1, NOV
IF(ID(J).EQ.IBOR(L)) THEN
K=K+1
DO 30 LI=1, NDC
SENS(LI,K)=TAB(LI,J)
GOTO 10
ENDIF
20 CONTINUE
10 CONTINUE
ISKIL=K
DO 40 I=1, ISKIL
WRITE(6,50) (SENS(I,J), J=1, ISKIL)
WRITE(10,50) (SENS(I,J), J=1, ISKIL)
50 FORMAT(5X, 10(F6.2, 2X))
40 CONTINUE
RETURN
END
C***********************************************************************
C* SUBROUTINE SENSTY
C***********************************************************************
C***********************************************************************
SUBROUTINE SENSTY(ISEN, IAM)
DIMENSION STABB(100, 100), SIV(100), ISIB(100), SZZ(100, 100)
DIMENSION ISID(100), SIN(100), LNBVG(100), BMDD(100), SENS(100, 100)
DIMENSION IBOR(100), IVZAR(100), STOF(100, 100), SZV(100, 100)
DIMENSION ISDD(100), SSIN(100), SCC(100, 100), BBC(100), TAA(100, 100)
DIMENSION IPM(1000), ISIN(100), TBC(100, 100)
DIMENSION TAB(100, 100), IV(100), ID(100), IB(100), Z(100, 100)
DIMENSION IN(100), C(100, 100), TABB(100, 100), ZZ(1, 100)
COMMON/B1/TAB, IV, ID, IB, LIT
COMMON/B2/Z, IN, IP, IC, IW
COMMON/B3/IPVC, XMAX, NDV, INO
COMMON/B4/NPRNT, NDC, NDCO
COMMON/B5/NPRD, IPM, TABB, ZZ
COMMON/B8/C
COMMON/B9/NOC1, NDCC
COMMON/SS1/IBOR, IVZAR, STOF, ISDD, SZV, SSIN
REAL IN, IV
C TO SAVE THE OPTIMAL TABLEAU FOR SENSITIVITY

C
52 FORMAT(2X, 'CORRECT', 5X, 'ENTER', 2X, '1: YES', 2X, '2: NO')
53 FORMAT(2X, 'REENTER AGAIN')
GO TO (1, 2, 901), IAM
1 DO 61 I = 1, NDV
DO 62 J = 1, NDV
62 STABB(I, J) = TAB(I, J)
SIV(I) = IV(I)
ISIN(I) = IB(I)
61 CONTINUE
DO 63 I = 1, NDPPR
DO 64 J = 1, NDV
SZZ(I, J) = Z(I, J)
64 ISID(J) = ID(J)
63 SIN(I) = IN(I)
NDC = NDV
NDCO = NDPPR
901 IF(ISEN.EQ.1) THEN
WRITE(6, 108)
WRITE(10, 108)
108 FORMAT(5X, 'TO CHANGE THE RHS VALUES')
WRITE(6, 109)
WRITE(10, 109)
109 FORMAT(5X, 'ENTER THE NUMBER OF CHANGES IN RHS')
READ(5, *) ICHANG
WRITE(6, 51) ICHANG
WRITE(10,51) ICHANG

51  FORMAT(' OF CHANGES = ',2X,I4)

C
WRITE(6,110)
WRITE(10,110)

110  FORMAT(5X,'ENTER THE ROW NUMBER AND ITS VALUE RESPECTIVELY')
DO 120 I=1,ICHANG

55  READ(5,*) IRUD,PROD
WRITE(6,54) IRUD,PROD
WRITE(10,54) IRUD,PROD

54  FORMAT(2X,'ROW= ',2X,I4,' RHS= ',2X,F8.4)
WRITE(6,52)
WRITE(10,52)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 55
END IF

120  IVZAR(IRUD)=PROD
CALL BINVRS(SENS.ISKIL)
DO 121 I=1,ISKIL
SUM=0
DO 122 J=1,ISKIL
SUM=SUM+SENS(I,J)*IVZAR(J)
IV(I)=SUM
122  CONTINUE

C
C EVALUATE THE VALUE OF EACH PRIORITY LEVEL
C
DO 40 I=1,NOPRO
SSM=0
DO 41 KK=1,NOC
LOK=IB(KK)
41  SSM=SSM+C(I,LOK)*IV(KK)
40  IN(I)=SSM
CALL PTRG(NOV)
DO 50 I=1,ISKIL
IF(IV(I).LT.0) THEN
NOC2=NOC
CALL DUALSX(IPROW,LAB,IBALL)
GO TO 999
ENDIF
50  CONTINUE
999  RETURN
END IF
C FOR ADDITION OF NEW DECISION VARIABLES

IF(ISEN.EQ.2) THEN
WRITE(6,320)
WRITE(10,320)

320 FORMAT(5X,'A VARIABLE NEED TO BE ADDED')

322 FORMAT(5X,'ENTER ROW #,VAR # START FROM 1 AND THEN ITS VAL')
WRITE(6,323)
WRITE(10,323)

323 FORMAT(5X,'ENTER THE NUMBER OF NEW VARIABLES')
READ(5,*) NNVR
WRITE(6,56) NNVR
WRITE(10,56) NNVR

56 FORMAT(' # OF NEW VARIABLES=',2X,I4)
NOVB=NOV
DO 324 I=1,NNVR
NOVB1=NOVB+I
ID(NOVB1)=NOVB1
DO 998 IPPL=1,NOPRO
998 C(IPPL,NOVB1)=0.
DO 700 J=1,NOC
700 STOF(J,NOVB1)=0.0
324 CONTINUE
WRITE(6,321)
WRITE(10,321)

321 FORMAT(5X,'ENTER THE NUMBER OF NONZERO ELEMENTS')
READ(5,*) NOCZRO
WRITE(6,57) NOCZRO
WRITE(10,57) NOCZRO

57 FORMAT(2X, # OF NONZERO ELEMENTS =',2X,I4)
WRITE(6,322)
WRITE(10,322)
DO 326 J=1,NOCZRO
59 READ(5,*) IRUW,IRR,VALY1
WRITE(6,58) IRUW,IRR,VALY1
WRITE(10,58) IRUW,IRR,VALY1
58 FORMAT(2X,'ROW#',2X,13.2X,'VARIABLE#',2X,13.2X,'VALUE',2X,F8.4)
WRITE(6,52)
WRITE(10,52)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 59
ENDIF
NOV81=NOV+IRR
STOF(IRUW,NOVB1)=VALY1

326 CONTINUE
CALL BINVRS(SENS.ISKIL)
DO 329 IZB=1,NNVR
   NO=NOV+IZB
   DO 327 LIB=1.ISKIL
       SUM=0
       DO 328 LIC=1.ISKIL
           SUM=SUM+SENS(LIB,LIC)*STOF(LIC.NO)
       BMOD(LIB)=SUM
   CONTINUE
327 CONTINUE
DO 330 I=1.ISKIL
   TAB(I,NO)=BMOD(I)
329 CONTINUE
DO 333 IZB=1,NNVR
   NO=NOV+IZB
   DO 331 I=1,NOPRO
       SUM=0
       DO 332 J=1.NOC
           MP=IB(J)
           SUM=SUM+C(I,MP)*TAB(J,NO)
       Z(I,NO)=SUM-C(I.NO)
   CONTINUE
331 CONTINUE
333 CONTINUE
NOV=NO
CALL PTRG(NOV)
DO 335 INO=1,NOPRO
10 CALL PIVCOL
   IF(XMAX.EQ.0.OR.IPVC.EQ.0) GO TO 335
   CALL PIVROW(IPROW)
   CALL CALC(IPROW)
   CALL PTRG(NOV)
   IF(ING.EQ.1) GO TO 10
   IF(IN(ING).EQ.0) GO TO 335
335 CONTINUE
RETURN
ENDIF

C
CC ADD A NEW OBJECTIVE FUNCTION
C
IF(ISEN.EQ.3) THEN
   WRITE(6,400)
   WRITE(10,400)
400   FORMAT(5X,'YOU ARE IN THE PROCESS OF ADDING A OBJECTIVE FUNCTION')
WRITE(6,401)
WRITE(10,401)

401 FORMAT(5X,'ENTER # OF NEW CONSTRAINTS AND # OF NEW VARIABLES')
READ(5,*) KKNOC,KKNOD
WRITE(6,402) KKNOC,KKNOD
WRITE(10,402) KKNOC,KKNOD

402 FORMAT(5X,'NO. OF NEWCONST=',2X,I3/5X,'NO. OF NEWVAR=',I3)
WRITE(6,52)
WRITE(10,52)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 65
ENDIF
NOC1=NOC
NOVP=NOV+1
NOCP=NOC
NOC=NOC+KKNOC
NOV=NOV+KKNOD
INDV1=NOVP
DO 403 I=1,KKNOC
ID(INDV1)=INDV1
INDV1=INDV1+1

403 CONTINUE
DO 404 I=NOVP,NOV
DO 404 J=1,NOCP
TAB(J,I)=0.0
NOCC=NOC1+1
DO 900 IBOL=NOCC,NOC
DO 900 IBOK=1,NOV
TAB(IBOL,IBOK)=0.
WRITE(6,406)
WRITE(10,406)

406 FORMAT(5X,'ENTER NUMBER OF NONZERO ELEMENTS IN THE NEW + CONSTRAINTS')
READ(5,*) NZRON
WRITE(6,66) NZRON
WRITE(10,66) NZRON

66 FORMAT(2X,'# OF NONZERO ELEMENTS =',2X,I5)
WRITE(6,67)
WRITE(10,67)

67 FORMAT(2X,'ENTER ROW I,COLUMN J AND ITS VALUE')
DO 407 I=1,NZRON
WRITE(6,68) L,M,VALUE
WRITE(10,68) L,M,VALUE
68 FORMAT(2X,'ROW I =',2X,I3,'COLUMN J =',2X,I3,'VALUE=',2X,F8.4)
WRITE(6,52)
WRITE(10,52)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 69
ENDIF
69 READ(5,*) L,M,VALUE
TAB(L,M)=VALUE
407 CONTINUE
WRITE(6,408)
WRITE(10,408)
408 FORMAT(5X,'ENTER BASIS AND RHS VALUE OF EACH CONSTRAINT')
DO 409 I=1,KKNOC
71 READ(5,*) L,IBB,VIV
WRITE(6,70) L,IBB,VIV
WRITE(10,70) L,IBB,VIV
70 FORMAT(2X,'ROW=',2X,I3,'BASIS=',2X,I3,'RHS=',2X,F8.4)
WRITE(6,52)
WRITE(10,52)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 71
ENDIF
IB(L)=IBB
IV(L)=VIV
409 CONTINUE
WRITE(6,410)
WRITE(10,401)
410 FORMAT(5X,'ENTER THE PRIORITY LEVEL OF THIS NEW GOAL AND '+5X,'THE NUMBER OF NONZERO ELEMENTS IN THE NEW PRIORITY LEVEL')
READ(5,*) LPGOAL,NONZO
WRITE(6,72) LPGOAL,NONZO
WRITE(10,72) LPGOAL,NONZO
72 FORMAT(2X,'PRIORITY LEVEL=',2X,I3,2X,'#OF NONZERO ELEMENTS ++',2X,I3)
WRITE(6,412)
WRITE(10,412)
412 FORMAT(/5X,'ENTER THE PRIORITY WEIGHTS')
C
C TO ARRANGE THE MATRIX OF PRIORITY WEIGHTS
C
IF(LPGOAL.EQ.1) THEN
     DO 415 I=1,NOPRO
          K=I+1
          DO 416 J=1,NOV
              SCC(1,J)=0.0
              IF(J.GE.NOVP) THEN
                  SCC(K,J)=0.
              GO TO 416
          END IF
          SCC(K,J)=C(I,J)
  416 CONTINUE
     CONTINUE
  415 CONTINUE
     GO TO 426
     ENDIF
     NOPP=NOPRO+1
     IF(LPGOAL.EQ.NOPP) THEN
          DO 417 I=1,NOPRO
               DO 417 J=1,NOV
                   IF(J.GE.NOVP) THEN
                       SCC(I,J)=0.
                   GO TO 417
               END IF
               SCC(I,J)=C(I,J)
  417 CONTINUE
          DO 418 J=1,NOV
              SCC(NOPP,J)=0.0
          GO TO 426
     ENDIF
     IF(LPGOAL.NE.1.AND.LPGOAL.NE.NOPRO) THEN
          ILP1=LPGOAL-1
          ILP2=LPGOAL+1
          NPPP=NOPRO+1
          DO 420 I=1,NPPP
               IF(I.LE.ILP1) THEN
                    DO 421 J=1,NOV
                        IF(J.GE.NOVP) THEN
                            SCC(I,J)=0.
                        END IF
                        SCC(I,J)=C(I,J)
  421 CONTINUE
                    CONTINUE
               GO TO 420
          ENDIF
          IF(I.EQ.LPGOAL) THEN
               DO 422 J=1,NOV
                    SCC(LPGOAL,J)=0.0
  422 CONTINUE
     ENDIF
CONTINUE
GO TO 420
ENDIF
K=I-1
DO 423 J=1,NOV
IF(J.GE.NOVP) THEN
SCC(I,J)=C.
GO TO 423
ENDIF
SCC(I,J)=C(K,J)
423 CONTINUE
END
426 WRITE(6,73)
WRITE(10,73)
73 FORMAT(2X,'ENTER VAR #', AND PRIRITY WEIGHT OF THIS VARIABLE')
DO 427 I=1,NONZD
75 READ(5,' ') IVAR,IVAL
WRITE(6,74) IVAR,IVAL
WRITE(10,74) IVAR,IVAL
74 FORMAT(2X,'VARIABLE #',2X,I3,2X,'PRIRITY WEIGHT=',2X,I4)
WRITE(6,52)
WRITE(10,52)
READ(5,' ') ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 75
ENDIF
SCC(LPGOAL,IVAR)=IVAL
427 CONTINUE
NOC2=NOC
NOC3=NOC2-NOC1
NNOC=NOC1+1
DO 500 NBC=1,NOC3
DO 501 J=1,NOV
DO 502 I=1,NOC1
IF(IB(Il.EQ.ID(J)) THEN
L=I
IF(TAB(NNOC,J).NE.O.) THEN
IF(TAB(L,J).EQ.O) GO TO 501
DD=TAB(NNOC,J)/TAB(L,J)
DO 503 K=1,NOV
TAA(L,K)=-DD*TAB(L,K)
MARY=NOV+1
TAA(L,MARY)=-DD*IV(L)
502 CONTINUE
501 CONTINUE
500 CONTINUE
428 END
TBC(NNOC,K) = TAB(NNOC,K) + TAA(L,K)
TAB(NNOC,K) = TBC(NNOC,K)

503 CONTINUE
IV(NNOC) = IV(NNOC) + TAA(L, MARY)
GO TO 501
ENDIF
ENDIF

502 CONTINUE
501 CONTINUE
NNOC = NNOC + 1
500 CONTINUE

C
C TO EVALUATE THE VALUE OF PRIORITY LEVELS
NOPRO = NOPRO + 1
DO 600 K = 1, NOPRO
DO 601 I = 1, NOV
DO 602 J = 1, NOC
IF(IB(J) .EQ. ID(I)) THEN
Z(K, I) = 0.0
GO TO 601
ENDIF
602 CONTINUE
BSUM = 0.0
ASUM = 0.0
DO 603 J = 1, NOC
KBK = IB(J)
ASUM = ASUM + SCC(K, KBK) * TAB(J, I)
BSUM = BSUM + SCC(K, KBK) * IV(J)
603 CONTINUE
Z(K, I) = ASUM - SCC(K, I)
IN(K) = BSUM
601 CONTINUE
600 CONTINUE
NOPRO = NOPRO
CALL PTRG(NOV)
C
C TO FIND THE OPTIMAL SOLUTION
C
NOPRO = NOPRO
DO 902 I = 1, NOC
IF(IV(I) .LT. 0.) THEN
NOC2 = NOC
CALL DUALSX(IPROW, LAB, IBALL)
RETURN
ENDIF
902 CONTINUE
DO 604 IND=1,NOPRO

605 CALL PIVCDL
   IF(XMAX.EQ.0.OR.IPVC.EQ.0) GO TO 604
   CALL PIVROW(IPROW)
   CALL CALC(IPROW)
   CALL PTRG(NOV)
   IF(IND.EQ.1) GO TO 605
   IF(IN(IND).EQ.0) GO TO 604

604 CONTINUE
   RETURN
END IF

C TO CHANGE A(I,J) THE COEFFICIENTS OF THE JTH VAR. IN THE ITH ROW
C NOTE A(I,J) ARE ASSOCIATED WITH NONBAIC VARIABLES ONLY
C
   IF(ISEN.EQ.4) THEN
     WRITE(6,300)
     WRITE(10,300)
   300 FORMAT(5X, 'IT IS ONLY POSSIBLE TO CHANGE THE COEFICIENT OF
         + THE NONBASIC VARIABLES')
   C FIND THE LIST OF NONBASIC VARIABLES
   K=0
   DO 301 I=1,NOV
     LBC=0
     DO 302 J=1,NOC
       IF(ID(I).NE.IB(J)) THEN
         LBC=LBC+1
       END IF
     302 CONTINUE
     IF(LBC.EQ.NOC) THEN
       K=K+1
       LNBVG(K)=ID(I)
     END IF
   301 CONTINUE
   IF(LBC.EQ.NOC) THEN
     K=K+1
     LNBVG(K)=ID(I)
   END IF
   DO 304 LK=1,K
     WRITE(6,303) LNBVG(LK)
     WRITE(10,303) LNBVG(LK)
   303 FORMAT(/5X,12)
   304 CONTINUE
   WRITE(6,305)
   WRITE(10,305)
   305 FORMAT(/5X,'ENTER THE ROW #, VARIABLE #, AND THEN ITS VALUE')
   READ(*,*) IZC,IZB,POT
   WRITE(6,76) IZC,IZB,POT
WRITE(10,76) IZC,IZB,POT
76 FORMAT(2X,'ROW=',2X,I3,2X,'VAR=',2X,I3,2X,'VALUE=',2X,F8.4)
WRITE(6,52)
WRITE(10,52)
REA0(5,•) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,53)
WRITE(10,53)
GO TO 77
END IF
STDF(IZC,IZB)=PDT
CALL BINVRS(SENS,ISKIL)
DO 306 LIB=1,ISKIL
SUM=0
DO 307 LIC=1,ISKIL
307 SUM=SUM+SENS(LIB,LIC)•STDF(LIC,IZB)
BMOD(LIB) =SUM
306 CONTINUE
DO 309 I=1,ISKIL
309 TAB(I,IZB)=BMOD(I)
C TO FIND THE VALUE OF Z(!,u)
DO 3007 I=1,NOPRO
SUM=0
DO 308 u=1,NOC
MPciB(u)
SUM=SUM+C(I,MP)•TAB(u,IZB)
308 CONTINUE
Z(I,IZB)=SUM-C(I,IZB)
3007 CONTINUE
C CALL PTRG(NOV)
DO 3009 IN0=1,NOPRO
310 CALL PIVCOL(IF(MAX.EQ.0.OR.IPVC.EQ.0)) GO TO 310
CONTINUE
IF(IN0.EQ.1) GO TO 310
IF(IN(IN0).EQ.0) GO TO 3009
3009 CONTINUE
C END IF
RETURN
WRITE(10,3)
3 FORMAT(/5X,'DO YOU WISH TO CONTINUE WITH THE RESULTS ' +/5X,'ASSOCIATED WITH THE SENSITIVITY ANALYSIS')
WRITE(6,4)
WRITE(10,4)
4 FORMAT(/5X,'** ENTER 1 FOR YES */5X,** ENTER 2 FOR NO **')
C READ(5,*) NNDDYY
IF(NNDDYY.EQ.1) RETURN
IF(NNDDYY.EQ.2) THEN
NOC=NCO
NDV=NVC
NPRO=NPQ
DO 5 I=1,NOC
DO 6 J=1,NDV
6 TAB(I,J)=STABB(I,J)
IV(I)=SIV(I)
IB(I)=ISIN(I)
5 CONTINUE
C DO 7 I=1,NPRO
DO 8 J=1,NDV
8 Z(I,J)=SZZ(I,J)
ID(I)=ISID(I)
7 IN(I)=SIN(I)
ENDIF
RETURN
END
APPENDIX B

INTERACTIVE COMPUTER PROGRAM FOR THE
STOCHASTIC VEHICLE ROUTING PROBLEM
C*---------------------------------------------------------------------
C* FORTRAN COMPUTER PROGRAM FOR THE
C* STOCHASTIC VEHICLE ROUTING PROBLEM (SVRP)
C*---------------------------------------------------------------------
C* AUTHOR : YAHYA ZARE-MEHRIJERDI
C* ADVISOR : DR. M. P. TERRELL
C* DATE : NOVEMBER 1986
C* COMPUTER : IBM 3081D
C*---------------------------------------------------------------------
C* SCHOOL OF INDUSTRIAL ENGINEERING
C* AND MANAGEMENT
C* STILLWATER, OK. 74078
C*---------------------------------------------------------------------
C* THIS PROGRAM ALLOWS THE USERS TO SOLVE THE FOLLOWING TYPES OF
C* THE SVRP:
C* 1. DETERMINISTIC VEHICLE ROUTING PROBLEM (DVRP)
C* 2. STOCHASTIC VEHICLE ROUTING PROBLEM HAVING ONLY PROBABILISTIC
C* CUSTOMER DEMANDS (SVRP)
C* 3. STOCHASTIC VRP WITH PROBABILISTIC CUSTOMER DEMAND AND
C* TRAVEL AND UNLOAD TIMES OF THE "F" TYPE PROBLEM AND
C* 4. STOCHASTIC VRP WITH PROBABILISTIC CUSTOMER DEMAND AND
C* TRAVEL AND UNLOAD TIMES OF THE "F" TYPE PROBLEM
C*---------------------------------------------------------------------
C* THE FOLLOWING SUBROUTINGS ARE USED IN THIS PROGRAM:
C* MEMOH = THIS SUBROUTING PROVIDES THE AVAILABLE MENUES
C* FOR THE USERS
C* DETERM = IT IS USED FOR CONTROLLING THE PROCESS OF PROBLEM
C* SOLVIG OF THE SVRP
C* TSORT, HEAPSN, SWAPN, PUSHD, AND SAVMAT = THESE SUBROUTINES
C* PERFORM TOGETHER TO PROVIDE THE SORTED SAVINGS FOR THE DVRP AND SVRP
C* INPT = THIS SUBROUTINE PROVIDES THE INPUT DATA FOR THE DVRP
C* RTCONT = IT IS USED TO CONSTRUCT THE VEHICLE ROUTES
C* INTR = CHECKS THE EXISTENCE OF ANY INTERIOR STATION IN CONSTRUCTED
C* VEHICLE ROUTES FOR ROUTING A NEW PAIR OF SELECTED STATIONS
C* COMBND = TO ADD A NEW STATION INTO AN AVAILABLE ROUTE
C* COMBRT = IT IS USED FOR PURPOSE OF COMBINING TWO ROUTES TOGETHER
C* FEASBL = CHECKS THE FEASIBILITY OF THE ROUTES FOR DVRP
C* CTD = EVALUATE THE COST, TIME OR DISTANCE OF THE CONSTRUCTED VEHICLE ROUTES FOR DVRP
C* CHKK = EVALUATES THE TOTAL DEMAND OF EACH ROUTE FOR DVRP
C* WWRT = PRINTS THE FINAL INFORMATIONS FOR EACH CONSTRUCTED ROUTE
C* SWCH = IT PERFORMS THE TASK OF SWITCHING THE PLACE OF AVAILABLE ROUTES AFTER TWO ROUTES HAVE BEEN COMBINED TOGETHER
C* PROB = CONTROL THE PROGRAM FOR THE SVRP OF THE "E" TYPE PROBLEM
C* STINPT = TO READ THE INPUT DATA FOR THE SVRP
C* PRCHK = CHECKS THE ROUTE FEASIBILITY FOR THE "E" AND "F" TYPE PROBLEMS
C* STFSBL = CHECK THE FEASIBILITY OF ADDING A NODE INTO A ROUTE OR COMBINING TWO ROUTES TOGETHER
C* CONTR = EVALUATE THE VARIANCE DEMAND, TRAVELLING, AND UNLOADING TIMES BEFORE ADDITION OF ANY NODE INTO A ROUTE
C* SOFT = DETERMINE MEAN AND VARIANCE OF TRAVEL TIME OF EACH ROUTE
C* RUSH = DETERMINE MEAN AND VARIANCE OF DEMAND AND UNLOAD TIME OF ROUTE
C* STSAVE = DETERMINE THE SORTED SAVINGS FOR THE SVRP OF THE "F" TYPE PROBLEM
C* FSBL = CHECK THE FEASIBILITY FOR SVRP WITH ONLY PROBABILISTIC CUSTOMER Demands
C* FCHECK = IS USED FOR THE SVRP WITH THE PROBABILISTIC DEMAND
C* FAST = DETERMINE TOTAL DEMAND OF A ROUTE BEFORE ADDITION OF A NEW NODE
C* STCONT = EVALUATE THE VARIANCE OF DEMAND
C* STARS = IS USED FOR SOLVING THE "E" TYPE PROBLEM
C* STCTD = EVALUATE THE TOTAL ELAPSE TIME FOR SVRP OF THE "F" TYPE PROBLEM
C* THE FOLLOWING IS THE LIST OF VARIABLES USED IN THIS PROGRAM
C*
C* NPT = NUMBER OF DEMAND POINTS INCLUDING CENTRAL DEPOT
C* TCAP = VEHICLE CAPACITY, ALL VEHICLES ARE ASSUMED TO BE HOMOGENEOUS
C* X(I) = IS THE X COORDINATE OF STATION I
C* Y(I) = IS THE Y COORDINATE OF STATION I
C* DIST(I,J) = DISTANCE BETWEEN STATIONS I AND J
C* DDT = STANDS FOR THE DETERMINISTIC VRP
C* SST = STANDS FOR THE STOCHASTIC VRP
C* MSVA = ARRAY OF SAVING AFTER SORTING
C* NSAVE = ARRAY OF SAVING BEFORE SORTING
C* NB(I) = A POINTER WHICH SHOWS THE FIRST ELEMENT OF EACH VEHICLE ROUTE
C* NF(I) = A POINTER WHICH SHOWS THE LAST ELEMENT OF EACH VEHICLE ROUTE
C* NR(I) = A POINTER WHICH SHOWS THE NUMBER OF STATIONS ON EACH VEHICLE ROUTE
C* TDMAND(I) = TOTAL DEMAND OF VEHICLE ROUTE I
P = NUMBER OF CONSTRUCTED VEHICLE ROUTE
NTRY = IS THE NUMBER OF SORTED SAVINGS WHICH ARE GREATER THAN ZERO
ROUTE(I,J)=INDICATES THE JTH ELEMENT OF ROUTE I
ALPHA = IS THE NORMAL DEVIATE OF THE ROUTE FAILURE PROBABILITY FOR TRAVEL TIME
BATA = IS THE NORMAL DEVIATE OF THE ROUTE FAILURE PROBABILITY FOR UNLOAD TIME
ATAH = IS THE NORMAL DEVIATE OF THE ROUTE FAILURE PROBABILITY FOR THE DEMAND
UTIME = UPPER BOUND OF UNLOAD TIME FOR EACH VEHICLE ROUTE
TTTIME = UPPER BOUND OF TRAVEL TIME FOR EACH VEHICLE ROUTE
IEE = INDICATES THE "E" TYPE PROBLEM
IFF = INDICATES THE "F" TYPE PROBLEM
DELTA = A CONSTANT VALUE USED IN THE ALGORITHM 2 OF "F" TYPE PROBLEM
IALGOL = INDICATES THE TYPE OF ALGORITHM: 1 OR 2.
BKAMA = INDICATES THE VALUE OF GAMA FOR ALGORITHM 1 OF "F" TYPE PROBLEM
KPRO = INDICATES THAT THE PROBLEM IS SVRP WITH ONLY PROBABILITY

DMAND(I) = MEAN DEMAND OF DEMAND POINT I
VDMAND(I) = VARIANCE OF DEMAND POINT I
MEAN(I,J) = MEAN TRAVEL TIME BETWEEN STATIONS I AND J
VARS(I,J) = VARIANCE OF TRAVEL TIME BETWEEN STATIONS I AND J
MINE(I) = MEAN UNLOAD TIME OF STATION I
VIRS(I) = VARIANCE OF UNLOAD TIME OF STATION I
IPROBL = NUMBER OF PROBLEMS TO BE SOLVED
NCHANG = NUMBER OF LOCATIONS WHICH THEIR COORDINATES ARE NEEDED TO BE CHANGED
IDMN = MEAN DEMAND OF THE SPECIAL LOCATION THAT NEED TO BE CHANGED
IVDMN = VARIANCE DEMAND OF THE SPECIAL LOCATION THAT NEED TO BE CHANGED
MCHANG = NUMBER OF NECESSARY CHANGES IN THE TRAVEL TIME
THAT KEEP THE ITH SUBSCRIPT OF THE SORTED
TJHAT(I) = AN ARRAY THAT KEEP THE JTH SUBSCRIPT OF THE SORTED
SAVING S(I,J)
DD = TOTAL COST, TIME, OR DISTANCE
TLOAD(I) = TOTAL UNLOAD TIME OF ROUTE I
TRAVL(I) = TOTAL TRAVEL TIME OF ROUTE I
WAR(I,J) = THE SAVING IN VARIANCE FOR RANDOM VARIABLE TRAVEL TIME USING ALGORITHMS(I) AND (II) OF "F" TYPE PROBLEM
MAR(I,J) = THE SAVING IN MEAN FOR RANDOM VARIABLE TRAVEL TIME USING ALGORITHM (I) AND (II) OF "F" TYPE PROBLEM
C* TTOTAL = TOTAL TRAVEL TIME FOR "F" TYPE PROBLEM
C* IDDT = 0 USING EUCLIDIAN DISTANCE
C* = 1 USING STRAIGHT LINE DISTANCE
C* GAMA = IS THE NORMAL DEVIATE OF THE ROUTE FAILURE PROBABILITY
C* FOR SVRP HAVING ONLY PROBABILISTIC CUSTOMER DEMANDS
C* IDSTB = 1 FOR DISTRIBUTIONS SUCH AS POISSON,BINOMIAL,GAMMA,
C* EXPONENTIAL,NEGATIVE BINOMIAL,AND CHI-SQUARE
C* 0 FOR OTHER DISTRIBUTIONS
C
C*************************************************************************
C* MAIN PROGRAM
C*************************************************************************
DIMAENSION NSA(5000),TI(5000),TJ(5000)
INTEGER TIHAT,TJHAT,TI,TJ,NSA,DIST
INTEGER FLI,FLU,FLIJI,FIJ,ROUTE,Demand,R,P,XX
INTEGER TCAP,X,Y,T,PP,TT,DDT,SST
COMMON/A1/(X(300),Y(300))
COMMON/A2/(NPT,NW,TCAP,MNP,NTRY)
COMMON/A3/(MSVA(5000),NSAVE(5000),XX(5000))
COMMON/A4/(NB(100),NF(100),NR(100),P)
COMMON/A5/(Demand(300),TDemand(100))
COMMON/A6/(LI,LJ,LI1,LJ1,LJ2,LRI,LRI2,LRIJ)
COMMON/A7/(IVB,IBV,WB)
COMMON/A8/(DIST(300,300))
COMMON/A9/(TIHAT(5000),TJHAT(5000),ROUTE(100,100))
COMMON/A10/(ALPHA,BATA,ATAH,UTIME,TTTIME)
COMMON/A12/(DDT,SST,IZAR)
COMMON/A15/(IEE,IFF,DELTA.IALGOL,BKAMA)
COMMON/A16/(KPRO,GAMA)
C* READ THE NUMBER OF PROBLEMS TO BE SOLVED
 35  WRITE(6,30)
      WRITE(10,30)
 30  FORMAT(/5X,'NUMBER OF PROBLEMS YOU WISH TO SOLVE')
      READ(S.*) IPROBL
      WRITE(6,31) IPROBL
      WRITE(10,31) IPROBL
 31  FORMAT(/5X,'NUMBER OF PROBLEMS=’,2X,I2)
      WRITE(6,3)
      WRITE(10,3)
      READ(S.*) ICORR
      IF(ICORR.EQ.2) THEN
          WRITE(6,4)
          WRITE(10,4)
      GO TO 35
      ENDF
DO 99 I=1,IPROBL
IF(IPROBL.GE.1) THEN
WRITE(6,33)
WRITE(10,33)
33 FORMAT(//5X,'SELECT ONE OF THE FOLLOWINGS')
ELSE
WRITE(6,34)
WRITE(10,34)
34 FORMAT(//5X,'ERROR MESSAGE',2X,'REENTER AGAIN')
GO TO 35
ENDIF
C
IAAA=0
IZAR=1
DELGAM=0
C
C TO DISPLAY MENU 1
C
KALL=1
CALL MEMOH(MOD,KALL,MOH)
C
C PURPOSE TO SOLVE THE DETERMINISTIC VRP
1000 IF(ODT.EQ.1) THEN
CALL DETERM
GO TO 999
ENDIF
C PURPOSE TO SOLVE THE SVRP HAVING ONLY PROBABILISTIC DEMAND
C
IF(KPRO.EQ.1) THEN
CALL STATS
GO TO 999
ENDIF
C PURPOSE TO SOLVE THE SVRP
501 IF(SST.EQ.1) THEN
C PURPOSE TO SOLVE THE "E" TYPE PROBLEM OF SVRP
C
IF(IEE.EQ.1) THEN
CALL PROB
GO TO 999
ENDIF
C PURPOSE TO SOLVE THE "F" TYPE PROBLEM OF SVRP
C
IF(IFF.EQ.1) THEN
8 FORMAT(//15X,'--->',2X,'ENTER A VALUE FOR DELTA')
9 FORMAT(//15X,'--->','SUGGESTED VALUES ARE'/19X,'.5,1.,1.5,'+2.2.5,3,3.5,4')
7 WRITE(6,32)
WRITE(10,32)
32  FORMAT(//10X,'ENTER YOUR CHOICE OF ALGORITHMS "F" TYPE PROB.')
    WRITE(6,36)
    WRITE(10,36)
36  FORMAT(//10X,'ENTER 1 ----> ALGORITHM I'/16X,'2 ---->ALGORITHM +II')
C
    READ(5,*) IALGOL
    WRITE(6,5) IALGOL
    WRITE(10,5) IALGOL
5   FORMAT(//5X,'THE SELECTED ALGORITHM IS',2X,I2)
    WRITE(6,3)
    WRITE(10,3)
    READ(5,*) ICORR
    IF(ICORR.EQ.2) THEN
        WRITE(6,4)
        WRITE(10,4)
        GO TO 7
    END IF
C TO OBSERVE FOR CHANGES OF ALGORITHM
    IFAA=IFAA+1
    IF(IFAA.GE.2) THEN
        WRITE(6,502)
        WRITE(10,502)
502  FORMAT(5X,'DO YOU WISH TO CHANGE THE VALUE OF DELTA OR GAMA')
        WRITE(6,503)
        WRITE(10,503)
503  FORMAT(5X,'ENTER',2X,'1: YES',2X,'2: NO')
C TO SEE IF THE USER WANTS TO CHANGE THE VALUE OF DELTA OR GAMA
    READ(5,*) DELGAM
    IF(DELGAM.EQ.1) IZAR=2
    ENDIF
    IF(IFAA.EQ.1) THEN
        IFALGL=IALGOL
        ELSE
            IF(IFALGL.EQ.IALGOL) GO TO 888
            IZAR=2
            IFALGL=IALGOL
        ENDIF
C TO USE THE ALGORITHM(I) OF "F" TYPE PROBLEM
888  IF(IALGOL.EQ.2) THEN
12   WRITE(6,8)
    WRITE(10,8)
    WRITE(6,9)
    WRITE(10,9)
    READ(5,*) DELTA
    WRITE(6,6) DELTA
WRITE(10,6) DELTA

6 FORMAT(//5X,'DELTA=',F8.3)
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 12
ENDIF
ENDIF
C TO USE THE ALGORITHM(1) OF "F" TYPE PROBLEM
IF(IALGOL.EQ.1) THEN
14 WRITE(6,37)
WRITE(10,37)
37 FORMAT(//10X,'ENTER A VALUE FOR GAMA')
WRITE(6,38)
WRITE(10,38)
38 FORMAT(//10X,'0 < GAMA < = 1 ')
READ(B,*) BKAMA
WRITE(6,21) BKAMA
WRITE(10,21) BKAMA
21 FORMAT(//5X,'BKAMA=',2X,F8.3)
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 14
ENDIF
CALL STATS
GO TO 999
ENDIF
IF(DELTA.EQ.0) THEN
11 FORMAT(//15X,'--->',2X,'ZERO IS NOT ACCEPTABLE. TRY AGAIN')
GO TO 12
ENDIF
CALL STATS
ELSE
WRITE(6,20)
WRITE(10,20)
20 FORMAT(BX,'PLEASE CHECK YOUR FIRST DATA CARD')
ENDIF
ENDIF

C TO DO ANY NECESSARY CHANGES FOR THIS PROBLEM BEFORE MOVING TO ANOTHER
C PROBLEM.
C TO DISPLAY MENU 2
999 KALL=2
   CALL MEMOH(MDD,KALL,MOH)
   IF(MOH.EQ.12) GO TO 99
   IF(MOH.EQ.10) THEN
      SST=1
      IEE=1
      IFF=0
      IZAR=4
      CALL PROB
   ENDIF
   GO TO 999
   IF(MOH.EQ.11) GO TO 501
   GO TO 1000
99 CONTINUE
3 FORMAT(/5X,'CORRECT',2X,'ENTER',2X,'1:YES',2X,'2:NO')
4 FORMAT(/5X,'REENTER AGAIN')
STOP
END

C*****************************************************************************
C*****************************************************************************
C* SUBROUTINE MEMOH
C*****************************************************************************
C*****************************************************************************
C
SUBROUTINE MEMOH(MDD,KALL,MOH)
DIMENSION MEAN(300,300),VIRS(300),MINE(300),VARS(300,300)
DIMENSION VDMAND(300)
COMMON/A1/X(300),Y(300)
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A3/MSVA(5000),NSAVE(5000),XX(5000)
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A9/TIHAT(5000),TJHAT(5000).ROUTE(100,100)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A12/DDT,SST,IZAR
COMMON/A15/IEE,IFF,DELTA,IALGOL,BKAMA
COMMON/A16/KPRO,GAMA
INTEGER DDT,ROUTE,P,SST,TCAP,X,Y,DMAND
INTEGER VDMAND,VARS,VIRS,UTIME,TTTIME
3 FORMAT(/5X,'CORRECT',2X,'ENTER',2X,'1:YES',2X,'2:NO')
4 FORMAT(/5X,'REENTER AGAIN')
GO TO (1,2),KALL
C DISPLAY OF MENU 1
C
1 WRITE(6,8)
   WRITE(10,8)
8 FORMAT(15X,'*** DISPLAY OF MENU 1 *** ')
   WRITE(6,10)
   WRITE(10,10)
10 FORMAT(/5X,'TO SOLVE THE DETERMINISTIC VRP ')
   WRITE(6,11)
   WRITE(10,11)
11 FORMAT(5X,' ** ENTER 1 ** ')
   WRITE(6,20)
   WRITE(10,20)
20 FORMAT(5X,'TO SOLVE A SVRP WITH PROBABILISTIC DEMAND ')
   WRITE(6,21)
   WRITE(10,21)
21 FORMAT(5X,' ** ENTER 2 ** ')
   WRITE(6,30)
   WRITE(10,30)
30 FORMAT(5X,'TO SOLVE SVRP OF "E" TYPE PROBLEM ')
   WRITE(6,31)
   WRITE(10,31)
31 FORMAT(5X,' ** ENTER 3 ** ')
   WRITE(6,40)
   WRITE(10,40)
40 FORMAT(5X,'TO SOLVE SVRP OF "F" TYPE PROBLEM ')
   WRITE(6,41)
   WRITE(10,41)
41 FORMAT(5X,' ** ENTER 4 ** ')
C
9 FORMAT(/5X,'*** CHOOSE THE OPTION *** ')
IF(IZAR.EQ.2) THEN
9 WRITE(6,621)
   WRITE(10,621)
621 FORMAT(/5X,'ENTER ONLY 3 OR 4 ')
ENDIF
READ(5,*) MOO
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
9 WRITE(6,4)
   WRITE(10,4)
GO TO 1
DDT=0
SST=C
IEE=0
IFF=0
KPRO=0

IF(MOO.EQ.1) THEN
WRITE(6,50)
WRITE(10,50)
DDT=1
GO TO 999
ENDIF

IF(MOO.EQ.2) THEN
WRITE(6,60)
WRITE(10,60)
SST=1
KPRD=1
GO TO 999
ENDIF

IF(MOO.EQ.3) THEN
WRITE(6,70)
WRITE(10,70)
SST=1
IEE=1
GO TO 999
ENDIF

IF(MOO.EQ.4) THEN
WRITE(6,80)
WRITE(10,80)
SST=1
IFF=1
GO TO 999
ENDIF

50 FORMAT(//5X,'A SOLUTION TO DVRP IS REQUIRED')
60 FORMAT(//5X,'A SOLUTION TO SVRP WITH PROBABILISTIC DEMAND'/
+5X,'IS REQUIRED')
70 FORMAT(//5X,'YOUR PROBLEM IS SVRP OF "E" TYPE PROBLEM ')
80 FORMAT(//5X,'YOUR PROBLEM IS SVRP OF "F" TYPE PROBLEM ')
999 RETURN
2 CONTINUE
C
ICHANG=1
C DISPLAY OF MENU 2
519 WRITE(6,7)
WRITE(10,7)
7 FORMAT(15X, '*** DISPLAY OF MENU 2***')
WRITE(6,90)
WRITE(10,90)
90 FORMAT(5X, '** DO YOU WISH TO DO ANY CHANGES **')
WRITE(6,100)
WRITE(10,100)
100 FORMAT(5X, 'TO CHANGE THE CAPACITY OF TRUCK')
WRITE(6,101)
WRITE(10,101)
101 FORMAT(5X, '** ENTER 1 **')
WRITE(6,102)
WRITE(10,102)
102 FORMAT(5X, 'TO CHANGE THE "UTIME" OR "TTTIME"')
WRITE(6,103)
WRITE(10,103)
103 FORMAT(5X, '** ENTER 2 **')
WRITE(6,104)
WRITE(10,104)
104 FORMAT(5X, 'TO CHANGE "ALPHA","BATA" AND "ATAH"')
WRITE(6,105)
WRITE(10,105)
105 FORMAT(5X, '** ENTER 3 **')
WRITE(6,106)
WRITE(10,106)
106 FORMAT(5X, 'TO CHANGE THE COORDINATE OF LOCATIONS')
WRITE(6,107)
WRITE(10,107)
107 FORMAT(5X, '** ENTER 4 **')
WRITE(6,108)
WRITE(10,108)
108 FORMAT(5X, 'TO CHANGE THE CUSTOMER DEMAND')
WRITE(6,109)
WRITE(10,109)
109 FORMAT(5X, '** ENTER 5 **')
WRITE(6,110)
WRITE(10,110)
110 FORMAT(5X, 'TO CHANGE THE UNLOAD TIME')
WRITE(6,111)
WRITE(10,111)
111 FORMAT(5X, '** ENTER 6 **')
WRITE(6,112)
WRITE(10,112)
112 FORMAT(5X, 'TO CHANGE THE TRAVEL TIME')
WRITE(6,113)
WRITE(10,113)
113 FORMAT(5X, '** ENTER 7 **')
WRITE(6,114)
WRITE(10,114)
114 FORMAT(5X,'*** TO DO NO CHANGES ENTER 8 ***')
WRITE(6,622)
WRITE(10,622)
622 FORMAT(5X,'TO CHANGE ALGORITHM I INTO II OR VISE VERSA')
WRITE(6,623)
WRITE(10,623)
623 FORMAT(5X,'*** ENTER 9 ***')
IF(ICHANG.EQ.1) THEN
WRITE(6,91)
WRITE(10,91)
READ(5,*) NCHA
ENDIF
IF(NCHA.GT.1) THEN
WRITE(6,501)
WRITE(10,501)
ENDIF
WRITE(6,25)
WRITE(10,25)
WRITE(6,25)
WRITE(10,25)
25 FORMAT(15X,'*** CHOOSE THE OPTION ***')
518 READ(5,*) MOH
WRITE(6,624) MOH
WRITE(10,624) MOH
624 FORMAT(/5X,'YOUR OPTION =',2X,I3)
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 518
ENDIF
IF(MOH.EQ.9) THEN
IZAR=2
GO TO 130
ENDIF
IF(MOH.EQ.8) THEN
WRITE(6,618)
WRITE(10,618)
618 FORMAT(/5X,'DO YOU WISH TO SOLVE THIS PROBLEM BY ANOTHER'/
+5X,'METHOD. POSSIBLE SELECTIONS ARE "F"-->"E", AND "E"-->"F"')
WRITE(6,619)
WRITE(10,619)
619 FORMAT(/5X,'ENTER',2X,'1:F--->E',2X,'2:E--->F',2X,'3:NO')
READ(5,*) KYNOO
C PURPOSE TO CHANGE THE "F" TYPE PROBLEM INTO "E" TYPE
   IF(KYNOO.EQ.1) THEN
     MOH=10
     IZAR=4
   ENDIF

C PURPOSE TO CHANGE THE "E" TYPE PROBLEM INTO "F" TYPE
   IF(KYNOO.EQ.2) THEN
     SST=1
     IFF=1
     IEE=0
     MOH=11
     IZAR=2
   ENDIF
   IF(KYNOO.EQ.3) THEN
     MOH=12
     RETURN
   ENDIF
   GO TO 130
END IF

91 FORMAT(5X,'ENTER THE NUMBER OF CHANGES')
501 FORMAT(5X,'CHANG IN COORDINATION OR TRAVAL. TIME COMES LAST')
   IF(MOH.EQ.1) THEN
     WRITE(6,120)
     WRITE(10,120)
   120 FORMAT(5X,'ENTER THE NEW CAPACITY OF TRUCK')
502 READ(5,*) TCAP
   WRITE(6,115) TCAP
   WRITE(10,115) TCAP
115 FORMAT(//5X,'THE NEW CAPACITY OF TRUCK=',2X,I4)
   WRITE(6,3)
   WRITE(10,3)
   READ(5,*) ICORR
   IF(ICORR.EQ.2) THEN
     WRITE(6,4)
     WRITE(10,4)
     GO TO 502
   ENDIF
   IZAR=3
   DO 200 I=1,NTRY
     NSAVE(I)=MSVA(I)
     GO TO 130
   ENDIF
   IF(MOH.EQ.2) THEN
503 WRITE(6,121)
      WRITE(10,121)
121 FORMAT(5X,'ENTER NEW VALUES FOR UTIME AND TTTIME')
READ(5,*) UTIME ,TTTIME
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 503
ENDIF
WRITE(6,117) UTIME ,TTTIME
WRITE(10,117) UTIME ,TTTIME
117 FORMAT(/5X,'UTIME=' ,2X,I5,2X,'TTTIME=' ,2X,I5)
IZAR=3
DO 201 I=1,NTRY
201 NSAVE(I)=MSVA(I)
GO TO 130
ENDIF
IF(MOH.EQ.3) THEN
504 WRITE(6,122)
WRITE(10,122)
122 FORMAT(5X,'ENTER VALUES FOR ALPHA , BATA AND ATAH')
READ(5,*) ALPHA,BATA,ATAH
WRITE(6,118)
WRITE(10,118)
118 FORMAT(/5X,'THE NEW VALUES FOR ALPHA , BATA AND ATAH ARE')
WRITE(6,119) ALPHA,BATA,ATAH
WRITE(10,119) ALPHA,BATA,ATAH
119 FORMAT(//5X,'ALPHA=' ,2X,F6.3,2X,'BATA=' ,2X,F6.3,2X,'ATAH=' ,2X,F6.3)
IZAR=3
WRITE(6,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 504
ENDIF
GO TO 130
ENDIF
IF(MOH.EQ.4) THEN
505 WRITE(6,123)
WRITE(10,123)
123 FORMAT(5X,'ENTER THE # OF LOCATIONS NEED TO BE CHANGED')
READ(5,*) NLOCAT
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) GO TO 505
DO 140 I=1,NLOCAT
506 WRITE(6,141)
WRITE(10,141)
141 FORMAT(5X, 'ENTER LOCATION. THEN ITS COORDINATES X AND Y')
READ(5,*) LOCAT,JXJ,JYJ
WRITE(6,507) LOCAT,JXJ,JYJ
WRITE(10,507) LOCAT,JXJ,JYJ
507 FORMAT(5X, 'LOCATION=' ,2X, I3,1X, 'X=' ,2X, I3,2X, 'Y=' ,2X, I3)
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 506
END IF
X(LOCAT)=JXJ
Y(LOCAT)=JYJ
CONTINUE
WRITE(6,202)
WRITE(10,202)
202 FORMAT(25X, '** PLEASE WAIT **')
IZAR=2
GO TO 130
ENDIF
IF(MOH.EQ.5) THEN
508 WRITE(6,124)
WRITE(10,124)
124 FORMAT(5X, 'ENTER THE # OF CUSTOMER DEMAND POINTS TO BE CHANGED')
READ(5,*) NCUSTM
WRITE(6,42) NCUSTM
WRITE(10,42) NCUSTM
42 FORMAT(//5X, 'NUMBER OF CHANGES = ' ,2X, I5)
WRITE(6,3)
WRITE(10,3)
READ(5,*) ICORR
IF(ICORR.EQ.2) GO TO 508
DO 142 I=1,NCUSTM
510 WRITE(6,143)
WRITE(10,143)
143 FORMAT(5X, 'ENTER CUSTOMER # , MEAN AND THEN VARIANCE OF DEMAND')
READ(5,*) NCSTM,IDMN,IVDMN
WRITE(6,509) NCSTM,IDMN,IVDMN
WRITE(10,509) NCSTM,IDMN,IVDMN
WRITE(6, 3)
WRITE(10, 3)
READ(5, *) ICORR
IF (ICORR .EQ. 2) THEN
WRITE(6, 4)
WRITE(10, 4)
GO TO 510
ENDIF
DMAND(NCSTM) = IDMN
VDMAND(NCSTM) = IVDMN
CONTINUE
IZAR = 3
DO 203 I = 1, NTRY
NSAVE(I) = MSVA(I)
GO TO 130
ENDIF
IF (MDH .EQ. 6) THEN
WRITE(6, 125)
WRITE(10, 125)
FORMAT(5X, 'ENTER # OF CUSTOMERS WITH NEW UNLOAD TIME VALUES')
READ(5, *) NCUSTM
WRITE(6, 517) NCUSTM
WRITE(10, 517) NCUSTM
WRITE(6, 3)
WRITE(10, 3)
READ(5, *) ICORR
IF (ICORR .EQ. 2) GO TO 511
DO 144 I = 1, NCUSTM
WRITE(6, 145)
WRITE(10, 145)
FORMAT(5X, 'ENTER CUSTOMER #, MEAN AND VAR OF UNLOAD TIME')
READ(5, *) NCSTM, IDMN, IVDMN
WRITE(6, 512) NCSTM, IDMN, IVDMN
WRITE(10, 512) NCSTM, IDMN, IVDMN
WRITE(6, 3)
WRITE(10, 3)
READ(5, *) ICORR
IF (ICORR .EQ. 2) THEN
WRITE(6, 4)
WRITE(10, 4)
GO TO 513
ENDIF
VINE(NCSTM) = IDMN
VIRS(NCSTM)=IVDMN
CONTINUE
IZAR=3
DO 204 I=1,NTRY
NSAVE(I)=MSVA(I)
GO TO 130
END IF
IF(MOH.EQ.7) THEN
WRITE(6,126)
WRITE(10,126)
126 FORMAT(5X, 'ENTER THE # OF CHANGES OF TRAVEL TIME')
READ(5,* ) NCHANG
WRITE(6,514) NCHANG
WRITE(10,514) NCHANG
514 FORMAT(5X, '#OF CHANGES = ',2X,I3)
DO 147 I=1,NCHANG
516 WRITE(6,146)
WRITE(10,146)
146 FORMAT(5X, 'ENTER I,J,MEAN AND VARIANCE OF TRAVEL TIME')
READ(5,* ) K,L,KB,KZ
WRITE(6,515) K,L,KB,KZ
WRITE(10,515) K,L,KB,KZ
+2X,I3)
WRITE(6,3)
WRITE(10,3)
READ(5,* ) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,4)
WRITE(10,4)
GO TO 516
ENDIF
MEAN(K,L)=KB
VARS(K,L)=KZ
MEAN(L,K)=MEAN(K,L)
VARS(L,K)=VARS(K,L)
147 CONTINUE
IZAR=2
GO TO 130
ENDIF
130 ICHANG=0
NCHA=NCHA-1
IF(NCHA.GE.1) GO TO 519
DO 616 I=1,NW
ROUTE(I,1)=I
ROUTE(I,2)=1
MNP1=MNP-1
DO 617 J=3,MNP1
   ROUTE(I,J)=0
   NB(I)=0
   NF(I)=0
   NR(I)=2
RETURN
END

C******************************************************************************
C*                      SUBROUTINE DETERM                           *
C******************************************************************************

C
           SUBROUTINE DETERM

C***
           DIMENSION NSA(5000),TI(5000),TJ(5000)
           INTEGER TIHAT,TJHAT,NI,TJ,NSA,DIST
           INTEGER FLI,FLJ,FLIJ,FIU,ROUTE,Demand,R,P,XX
           INTEGER TCAP,X,Y,T,PP,TT
           INTEGER DDT,SST
           COMMON/A1/X(300),Y(300)
           COMMON/A2/NPT,NW,TCAP,MNP,NTRY
           COMMON/A3/MSVA(5000),NSAVE(5000),XX(5000)
           COMMON/A4/NB(100),NF(100),NR(100),P
           COMMON/A5/Demand(300),TDemand(100)
           COMMON/A6/LI,LJ,LI1,LJ1,LV1,LV2,LRI,LRI
           COMMON/A7/IBV,IWB
           COMMON/A8/DIST(300,300)
           COMMON/A9/TIHAT(5000),TJHAT(5000),ROUTE(100,100)
           COMMON/A12/DOT,SST,IZAR

C**** READ IN IDDT AS 0 OR 1.
C****  IDDT= 0 IF USING THE EUCLIDIAN DISTANCE
C****     = 1 IF USING THE STRAIGHT LINE DISTANCE
C****
4    WRITE(6,1)
    WRITE(10,1)
1    FORMAT(2X,'-->',5X,'ENTER 0 FOR EUCLIDIAN DISTANCE'//
       + 2X,'-->',5X,'ENTER 1 FOR LINEAR DISTANCE')
    READ(5,*) IDDT
    CALL INPT(IDDT)
5    CALL SAVMAT
    TT=NTRY
    CALL TSORT(NSAVE,TIHAT,TJHAT,NTRY)

C*** SET THE TOTAL DEMAND OF EACH ROUTE TO ZERO
DD SO I=1,NTRY
MSVA(I)=NSAVE(I)

DO 7 P=1,NW
T=1

7 TDemand(P)=0

P=1
R=3
IX=5
PP=P

ITI=TIHAT(T)
JTJ=TJHAT(T)

TDemand(P)=TDemand(P)+Demand(ITI)+Demand(JTJ)

IF(TDemand(P).LE.TCAP) THEN

ROUTE(P,R)=TIHAT(T)
NB(P)=ROUTE(P,R)

R=R+1

ROUTE(P,R)=TJHAT(T)
NF(P)=ROUTE(P,R)

NR(P)=R

ENDIF

K=T+1

IF(TDemand(P).GT.TCAP) THEN

NSAVE(K-1)=0
TDemand(P)=0
T=K
GO TO 11

ENDIF

C*** CONSTRUCT THE ROUTE

DO 10 T=K,TT
NSAVE(T-1)=0

IYOUTH=1

CALL INTR(IN,PP,T,IYOUTH)

IF(IN.EQ.1) GO TO 10

PP=P

CALL RTCONT(PP,T)

10 CONTINUE

IYOUTH=2

CALL INTR(IN,PP,T,IYOUTH)

CALL WWRT(PP)

RETURN

END

SUBROUTINE TSORT

SUBROUTINE TSORT (NSAVE, TIHAT, TJHAT, NTRY)

DIMENSION TIHAT(5000), TJHAT(5000), TI(5000), TJ(5000), NSAVE(5000)

DIMENSION NSA(5000)
INTEGER TIHAT, TJHAT, TI, TJ, NSA, NSAVE

C**
TO SORT IN DECREASING ORDER
CALL HEAPSN(NSAVE, TIHAT, TJHAT, NTRY)
DO 50 J=1, NTRY
   KK=NTRY+1-J
   NSA(KK)=NSAVE(J)
   TI(KK)=TIHAT(J)
   TJ(KK)=TJHAT(J)
50 CONTINUE
DO 60 I=1, NTRY
   NSAVE(I)=NSA(I)
   TIHAT(I)=TI(I)
   TJHAT(I)=TJ(I)
60 CONTINUE
WRITE(6, 10)
10 FORMAT(10X, 'SORTED SAVINGS')
RETURN
END

SUBROUTINE INPT(IDDT)
*
DIMENSION VDMAND(300), MEAN(300, 300), VARS(300, 300)
DIMENSION MINE(300), VIRS(300)
INTEGER TCAP, X, DDT, SST, Y, VDMAND, DMAND, TDMAND
COMMON/A1/X(300), Y(300)
COMMON/A2/NPT, NW, TCAP, MNP, NTRY
COMMON/A5/DMAND(300), TDMAND(100)
COMMON/A8/DIST(300, 300)
COMMON/A12/DDT, SST, IZAR
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
COMMON/A16/KPRO, GAMA
78 WRITE(6, 1)
    WRITE(10, 1)
1 FORMAT(5X, '---', 2X, 'ENTER THE NUMBER OF STOP POINTS'/
+15X, 'INCLUDING THE TERMINAL AND TRUCK CAPACITY RESPECTIVELY')
READ(5, *) NPT, TCAP
WRITE(6, 2) NPT, TCAP
WRITE(10, 2) NPT, TCAP
2 FORMAT(5X, 'NUMBER OF DEMAND POINTS=', I3, '/
+5X, 'CAPACITY OF TRUCK=', I5)
WRITE(6, 76)
WRITE(10, 76)
READ(5, *) ICORR
IF(ICORR.EQ.2) THEN
   WRITE(6, 77)
WRITE(10,77)
GO TO 78
ENDIF
IF(IDDT.EQ.1) GO TO 13
WRITE(6,3)
WRITE(10,3)
FORMAT(5X,'--->'5X.'ENTER THE EUCLIDIAN DISTANCE'/
+18X.'FOR ALL STOP POINTS AND TERMINALS')
WRITE(6,4)
WRITE(10,4)
FORMAT(5X,'--->'5X.'ENTER EUCLIDIAN DIST. FOR TERMINAL FIRST')
DO 10 I=1,NPT
79 READ(S.*) X(I),Y(I)
WRITE(6,16) X(I),Y(I)
WRITE(10,16) X(I),Y(I)
WRITE(6,76)
READ(S.*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,77)
WRITE(10,77)
GO TO 79
ENDIF
10 CONTINUE
C***
C** WRITE THE EUCLIDIAN DISTANCE

WRITE(6,11)
WRITE(10,11)
FORMAT(10X.'EUCLIDIAN DISTANCE',//16X.'X',8X,'Y')
DO 12 I=1,NPT
WRITE(6,16) X(I),Y(I)
WRITE(10,16) X(I),Y(I)
16 FORMAT(14X,I4,8X,I4)
12 CONTINUE
C****
C** EVALUATE THE DISTANCE BETWEEN POINTS I AND J

DO 20 I=1,NPT
DO 20 J=1,NPT
IF(I.EQ.J) DIST(I,J)=0
IF(I.GE.J) GO TO 20
WW=FLOAT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
DIST(I,J)=SQRT(WW)
DIST(J,I)=DIST(I,J)
20 CONTINUE
13 IF(IDDT.EQ.0) GO TO 95
WRITE(6, 19)
WRITE(10, 19)

19 FORMAT(5X, 'ENTER THE LINEAR DISTANCE BETWEEN THE POINTS')
DO 35 I = 1, NPT
READ(5, *) (DIST(I, J), J = I, NPT)
35 CONTINUE
DO 36 I = 2, NPT
K = I - 1
DO 37 J = 1, K
DIST(I, J) = DIST(J, I)
37 CONTINUE
36 CONTINUE
95 IF (DDT.EQ.1) THEN
WRITE(6, 7)
WRITE(10, 7)
7 FORMAT(5X, '--->', 5X, 'ENTER THE CUSTOMER DEMANDS')
DO 8 I = 2, NPT
81 READ(5, *) DMAND(I)
WRITE(6, 26) I, DMAND(I)
WRITE(10, 26) I, DMAND(I)
8 CONTINUE
ENDIF
8 CONTINUE
C*** WRITE THE DEMANDS
C***
WRITE(6, 24)
WRITE(10, 24)
24 FORMAT(2X, 'DEMAND POINT', 8X, 'DEMAND')
DO 75 I = 2, NPT
WRITE(6, 26) I, DMAND(I)
WRITE(10, 26) I, DMAND(I)
75 CONTINUE
GO TO 111
ENDIF
77 IF (ICORR.EQ.2) THEN
WRITE(6, 142)
WRITE(10, 142)
142 FORMAT(//10X, 'ENTER THE VALUE OF GAMA')
GO TO 111
ENDIF
READ(5,*) GAMA
WRITE(6,141) GAMA
WRITE(10,141) GAMA
FORMAT(//20X,'GAMA=',5X,F10.4)
WRITE(6,76)
WRITE(10,76)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,77)
WRITE(10,77)
GO TO 82
ENDIF
WRITE(6,112)
WRITE(10,112)
FORMAT(//10X,'---',2X,'ENTER MEAN AND VARIANCE OF DEMAND')
DO 113 I=2,NPT
83 READ(5,*) DMAND(I),VDMAND(I)
WRITE(6,116) I,DMAND(I),VDMAND(I)
WRITE(10,116) I,DMAND(I),VDMAND(I)
WRITE(6,76)
WRITE(10,76)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,77)
WRITE(10,77)
GO TO 83
ENDIF
113 CONTINUE
C** WRITE THE DEMAND
WRITE(6,114)
WRITE(10,114)
FORMAT(//20X,'DEMAND POINT',8X,'MEAN DEMAND',8X,'VAR DEMAND')
DO 115 I=2,NPT
WRITE(6,116) I,DMAND(I),VDMAND(I)
WRITE(10,116) I,DMAND(I),VDMAND(I)
116 FORMAT(//11X,I2,12X,I5,12X,I5)
CONTINUE
END IF
111 WRITE(6,38)
WRITE(10,38)
FORMAT(10X,'DISTANCE')
ITDD=0
DO 41 I=2,NPT
ITDD=ITDD+DMAND(I)
41 CONTINUE
NW=(ITDD/TCAP)+20
NNW=NW-20
MNP=(NPT/NNW)+1
MNP=MNP+7

76 FORMAT(/2X,'CORRECT',2X,'ENTER',1X,'1: YES',2X,'2: NO')
77 FORMAT(/2X,'REENTER AGAIN')
RETURN
END

C******************************************************************************
C** SUBROUTINE HEAPSN
C******************************************************************************
C
SUBROUTINE HEAPSN(XX,POS,PPOSS,N)
INTEGER XX(5000),POS(5000),PPOSS(5000)
N2=N/2
DO 10 J=1,N2
i=N2+1-J
10 CALL PUSHDN(XX,POS,PPOSS,I,N)
N1=N-1
DO 20 JJ=1,N1
i=N1+1-JJ
CALL SWAPN(XX(I),XX(I+1),POS(I),POS(I+1),PPOSS(I),PPOSS(I+1))
20 CALL PUSHDN(XX,POS,PPOSS,1,i)
RETURN
END

C******************************************************************************
C** SUBROUTINE SWAPN
C******************************************************************************
C
SUBROUTINE SWAPN(I,J,P,Q,R,S)
INTEGER P,Q,R,S
K=I
I=J
J=K
Z=P
P=Q
Q=Z
T=R
R=S
S=T
RETURN
END

C******************************************************************************
C** SUBROUTINE PUSHDN
C******************************************************************************
C
SUBROUTINE PUSHDN(XX,POS,PPOSS,I,N)
INTEGER XX(5000),POS(5000),PPOSS(5000)
LOGICAL FIN
FIN=.FALSE.
K=XX(I)
Z=POS(I)
T=PPOSS(I)
J=I/2

10 CONTINUE
IF(J.LE.N.AND..NOT.FIN) THEN
IVV=J+1
IF(IVV.LE.N) THEN
IF(J.LT.N.AND.XX(J).LT.XX(J+1)) J=J+1
ENDIF
IF(K.GE.XX(J)) THEN
FIN=.TRUE.
ELSE
XX(J/2)=XX(J)
POS(J/2)=POS(J)
PPOSS(J/2)=PPOSS(J)
J=J/2
ENDIF
XX(J/2)=K
POS(J/2)=Z
PPOSS(J/2)=T
GO TO 10
ENDIF
RETURN
END

C***************************************************************************
SUBROUTINE SAVMAT
***************************************************************************
SUBROUTINE SAVMAT
** THIS SUBROUTINE CONSTRUCT THE SAVING MATRIX AND THE INITIAL SOLUTION TO THE PROBLEM
DIMENSION ISAVE(300,300)
INTEGER TCAP,X,P,Tihat,Tjhat,ROUTE
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A3/MSVA(5000),NSAVE(5000),XX(5000)
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A8/Tihat(5000),Tjhat(5000),ROUTE(100,100)
COMMON/A9/DIST(300,300)
NTRY=0
DO 60 I=2,NPT
DO 60 J=I,NPT
ISAVE(I,J)=-99999
IF(DIST(I,J).EQ.0) GO TO 70

DO 60
ISAVE(I,J) = DIST(I,1) + DIST(1,J) - DIST(I,J)
70
ISAVE(I,J) = ISAVE(I,J)
60
CONTINUE
DO 80 I=2,NPT
ISAVE(I,1) = -99999
ISAVE(I,1) = ISAVE(I,1)
80
CONTINUE
C** CONSTRUCT THE INITIAL SOLUTION OR INITIAL ROUTE B
C** THE FOLLOWING MATRIX. PUT THE ARRAY ISAVE INTO THE
C** NEW ARRAY NSAVE WHICH IS ONE DIMENSIONAL.
L=0
IPT=NPT-1
DO 100 I=2,IPT
K=I+1
DO 100 J=K,NPT
IF(ISAVE(I,J).LE.0) GO TO 100
L=L+1
NSAVE(L) = ISAVE(I,J)
TIHAT(L) = I
TJHAT(L) = J
100
CONTINUE
NTRY=L
DO 170 I=1,NW
ROUTE(I,1) = I
ROUTE(I,2) = 1
ROUTE(I,MNP) = 1
MNP1 = MNP-1
DO 180 J=3,MNP1
ROUTE(I,J) = 0
180
DO 170 I=1,NW
WRITE(6,23) (ROUTE(I,J),J=1,MNP)
WRITE(10,23) (ROUTE(I,J),J=1,MNP)
23
CONTINUE
RETURN
END
C***************************************************************
C SUBROUTINE RTCONT
C***************************************************************
C SUBROUTINE RTCONT(PP, T)
DIMENSION VARS(300,300), MINE(300), MEAN(300,300)
DIMENSION VIRS(300), VDMAND(300)
INTEGER P, T, R, PP, ROUTE, TIHAT, TJHAT, DMAND, TDMAND
INTEGER DDT, SST, VARS, VIRS, VDMAND
COMMON/A4/NB(100),NF(100), NR(100), P
COMMON/A5/DMAND(300), TDMAND(100)
COMMON/A6/LI, LJ, LI1, LI2, LJ1, LJ2, LRI, LRJ
COMMON/A9/TIHAT(5000), TJHAT(5000), ROUTE(100, 100)
COMMON/A12/DDT, SST, IZAR
COMMON/A14/MEAN, MINE, VIRS, VDMAND
COMMON/A15/IEE, IFF, DELTA, IALGOL, BKAMA
COMMON/A16/KPRO, GAMA
COMMON/A17/KDMAND, KTULOD, KTTRVL

C** LRI= INDICATES ROUTE LRI
C** LRJ=INDICATES ROUTE LRJ
C** LI1 =INDICATES THAT TIHAT(T) IS EQUAL TO THE NB(KPP)
C** LI2 = INDICATES THAT TIHAT(T) IS EQUAL TO THE NF(KPP)
C** LJ1=INDICATES THAT TJHAT(T) IS EQUAL TO THE NB(KPP)
C** LJ2= INDICATES THAT TJHAT(T) IS EQUAL TO THE NF(KPP)

LRI=0
LRJ=0
LI1=0
LI2=0
LJ1=0
LJ2=0
LI=0
LJ=0
KPP=PP
DO 10 KPP=1, PP
IF(NB(KPP).EQ.TIHAT(T).OR.NF(KPP).EQ.TIHAT(T)) THEN
LI=1
LRI=KPP
IF(NB(KPP).EQ.TIHAT(T)) LI1=1
IF(NF(KPP).EQ.TIHAT(T)) LI2=1
IF(NB(KPP).EQ.TIHAT(T).OR.NF(KPP).EQ.TIHAT(T)) RETURN
ENDIF
IF(NB(KPP).EQ.TIHAT(T).OR.NF(KPP).EQ.TIHAT(T)) THEN
LJ=1
LRJ=KPP
IF(NB(KPP).EQ.TIHAT(T)) LJ1=1
IF(NF(KPP).EQ.TIHAT(T)) LJ2=1
IF(NB(KPP).EQ.TIHAT(T).OR.NF(KPP).EQ.TIHAT(T)) RETURN
ENDIF
10 CONTINUE
IF(LI.EQ.0.AND.LJ.EQ.0) THEN
P=PP+1
R=3
ROUTE(P,R)=TIHAT(T)
NB(P)=ROUTE(P,R)
R=R+1
ROUTE(P,R)=TJHAT(T)
NF(P)=ROUTE(P,R)
PP=P
NR(P)=R
RETURN
END IF
IF(LI.EQ.1.AND.LJ.EQ.1)
THEN
CALL COMBR(T,IVB,IWB,IXBB,IYBB,PP,T)
IF(IVB.EQ.0.AND.IWB.EQ.0.AND.IXBB.EQ.0.AND.IYBB.EQ.0)
RETURN
CALL SWTCH(IVB,IWB,IXBB,IYBB,PP)
RETURN
END IF
C* TO COMBINE TWO ROUTES TOGETHER
IF(LI.EQ.1.AND.LJ.EQ.0)
THEN
CALL CMDBND(T)
RETURN
END IF
C* TO ADD A NODE INTO AN EXISTING ROUTE
IF(LI.EQ.0.AND.LJ.EQ.1)
THEN
CALL CMDBND(T)
END IF
RETURN
END
C* SUBROUTINE INTR
C* SUBROUTINE INTR(IN,PP,T,IYOUTH)
INTEGER PP,TCAP,TIHAT,TJHAT,ROUTE,T
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A9/TIHAT(5000),TJHAT(5000),ROUTE(100,100)
GO TO (1,2),IYOUTH
1 IN=0
DO 20 I=1,PP
NRR=NR(I)-1
IF(NRR.LT.4) GO TO 20
DO 10 J=4,NRR
IF(ROUTE(I,J).EQ.TIHAT(T).OR.ROUTE(I,J).EQ.TJHAT(T)) IN=1
10 CONTINUE
20 CONTINUE
RETURN
ICANCL=0
DO 30 I=2,NPT
   DO 40 J=1,PP
      MNOPP=NR(J)
      DO 50 K=3,MNOPP
         IF(I.EQ.ROUTE(J,K)) GO TO 30
      50 CONTINUE
   40 CONTINUE
   ICANCL=I
   P=PP+1
   R=3
   ROUTE(P,R)=ICANCL
   NB(P)=ROUTE(P,R)
   NF(P)=ROUTE(P,R)
   NR(P)=R
   PP=P
30 CONTINUE
RETURN
END

SUBROUTINE COMBND
SUBROUTINE COMBND(T)
INTEGER TCAP,P,DMAND,THAT,THAT,ROUTE,FLI,FLJ
INTEGER FLIJ,FIJ,T,PP,DDT,SST
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A6/LI,LJ,LI1,LI2,LJ1,LJ2,LRI,LRJ
COMMON/A9/THAT(5000),THAT(5000),ROUTE(100,100)
COMMON/A12/DDT,SST,IZAR
COMMON/A15/II,IF,II,DEL,IALGDL,BKAMA
COMMON/A16/KPRO,GAMA
COMMON/A17/KDMAND,KTULOD,KTTRVL
IF(LI.EQ.1) THEN
   IF(LI.EQ.1) THEN
      IF(KPRO.EQ.1) THEN
         IX=1
         CALL FEASBL(IX,FLI,FLJ,FLI,J,FIJ,T,IZX)
         GO TO 5
      ENDIF
      IF(KPRO.EQ.1) THEN
         IX=1
         CALL FSBL(IX,FLI,FLJ,FLI,J,FIJ,T,IZX,KDMAND)
GO TO 5
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=1
CALL STFSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND,KTULOD,KTTRVL)
GO TO 5
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
IX=1
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 5
ENDIF
5 IF(FLI.EQ.1) THEN
C** ADD A NODE IN FRONT OF ROUTE LRI
LOC=4
NOV1=NR(LRI)+1
NVV=NOV1+1
NVZ=NVV-LOC
DO 10 K=1,NVZ
I=NVV-K
J=I-1
10 ROUTE(LRI,I)=ROUTE(LRI,J)
ROUT (LRI,3)=TJHAT(T)
NF(LRI)=ROUTE(LRI,NOV1)
NB(LRI)=TJHAT(T)
NR(LRI)=NOV1
ENDIF
RETURN
ENDIF
IF(LI2.EQ.1) THEN
IF(DDT.EQ.1) THEN
IX=1
CALL FEASBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX)
GO TO 15
ENDIF
IF(KPRO.EQ.1) THEN
IX=1
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 15
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=1
C * PURPOSE TO CHECK THE FEASIBILITY OF VEHICLE ROUTES
CALL STFSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND,KTULOD,KTTRVL)
GO TO 15
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
IX=1
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 15
ENDIF

15 IF(FLI.EQ.1) THEN
C** ADD A NODE AT THE END OF ROUTE OF LRI
LOC=NR(LRI)+1
ROUTE(LRI,LOC)=TJHAT(T)
NR(LRI)=LOC
NF(LRI)=TJHAT(T)
ENDIF
ENDIF
RETURN

ENDIF
IF(LJ.EQ.1) THEN
IF(LJ1.EQ.1) THEN
IF(DDT.EQ.1) THEN
IX=2
CALL FEASBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX)
GO TO 20
ENDIF
IF(KPRO.EQ.1) THEN
IX=2
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 20
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=2
CALL STFSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND,KTULOD,KTTRVL)
GO TO 20
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
IX=2
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 20
ENDIF

20 IF(FLJ.EQ.1) THEN
C** ADD A NODE IN THE FRONT OF ROUTE LRJ
LOC=4
NOV2=NR(LRJ)+1
NWW=NOV2+1
NWZ=NWW-LOC
DO 30 K=1,NWZ
I=NWW-K
J=I-1

30 CONTINUE
ROUTE(LRJ,I)=ROUTE(LRJ,J)
ROUTE(LRJ,3)=TIHAT(T)
NF(LRJ)=ROUTE(LRJ,N02)
NR(LRJ)=N02
NB(LRJ)=TIHAT(T)
ENDIF
RETURN
ENDIF
IF(LJ2.EQ.1) THEN
IF(DDT.EQ.1) THEN
IX=2
CALL FEASBL(IX,FLI,FLJ,FIJ,T,IX)
GO TO 25
ENDIF
IF(KPRO.EQ.1) THEN
IX=2
CALL FSBL(IX,FLI,FLJ,FIJ,T,IX,KDMAND)
GO TO 25
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=2
CALL STFSBL(IX,FLI,FLJ,FIJ,T,IX,KDMAND,KTULOD,KTTRVL)
GO TO 25
ENDIF
IF(SSi.EQ.1.AND.IFF.EQ.1) THEN
IX=2
CALL FSBL(IX,FLI,FLJ,FIJ,T,IX,KDMAND)
GO TO 25
ENDIF
25
IF(FLJ.EQ.1) THEN
C** ADD A NODE AT THE END OF ROUTE LRj
LOC=NR(LRJ)+1
ROUTE(LRJ,LOC)=TIHAT(T)
NR(LRJ)=LOC
C** NODE NB DOES NOT CHANGE
NF(LRJ)=TIHAT(T)
ENDIF
ENDIF
ENDIF
RETURN
END

SUBROUTINE COMBRT
DIMENSION TULOAD(100),TTRAVL(100)
INTEGER TIHAT,TJHAT,ROUTE,P,FLI,FLJ,FLIJ,FIJ
INTEGER DMAND,TDMAND,PP,T,DDT,SST
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A6/LI,LJ,LI1,LI2,LJ1,LJ2,LRI,LRJ
COMMON/A9/TIHAT(5000),TJHAT(5000),ROUTE(100,100)
COMMON/A12/DDT,SST,IZAR
COMMON/A15/IEE,IFF,DELTA,IALGOL,BKAMA
COMMON/A16/KPRO,GAMA
COMMON/A17/KDMAND,KTULOD,KTTRVL
IVB=0
IWB=0
IXBB=0
IYBB=0
IF(LI2.EQ.1.AND.LJ2.EQ.1) THEN
IF(DDT.EQ.1) THEN
IX=3
CALL FEASBL (IX,FLI,FLJ,FLIJ,FIJ,T,IZX)
GO TO 41
ENDIF
IF(KPRO.EQ.1) THEN
IX=3
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 41
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=3
CALL STFSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND,KTULOD,KTTRVL)
GO TO 41
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
IX=3
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 41
ENDIF
CC** FOR CHECK
C**
41 IF(FLIJ.EQ.1) THEN
IXBB=1
C** PUT ROUTE LRJ IN ROUTE LRI
LC=NR(LRI)+1
LD=NR(LRJ)-2
LB=LC+LD-1
U=NR(LRJ)
NF(LRI)=NB(LRJ)
DO 10 I=LC,LB
ROUTE(LRI,I)=ROUTE(LRJ,J)
ROUTE(LRJ,J)=0
J=J-1
10 CONTINUE

C** NB(LRI) DOES NOT CHANGE.
NR(LRI)=LB
NR(LRJ)=2
NB(LRJ)=0
NF(LRJ)=0
ELSE
RETURN
ENDIF

IF(DDT.EQ.1) THEN
TDMAND(LRI)=TDMAND(LRI)+TDMAND(LRJ)
TDMAND(LRJ)=0
RETURN
ENDIF

IF(KPRO.EQ.1) THEN
TDMAND(LRI)=KDMAND
TDMAND(LRJ)=0
RETURN
ENDIF

IF(SST.EQ.1.AND.IEE.EQ.1) THEN
TDMAND(LRI)=KDMAND
TULOAD(LRI)=KTULOD
TULOAD(LRJ)=O
TTRAVL(LRI)=KTTRVL
TTRAVL(LRJ)=O
RETURN
ENDIF

IF(SST.EQ.1.AND.IFF.EQ.1) THEN
TDMAND(LRI)=KDMAND
ENDIF
RETURN
ENDIF

IF(LI2.EQ.1.AND.LU1.EQ.1) THEN
IF(DDT EQ.1) THEN
IX=3
CALL FEASBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX)
GO TO 42
ENDIF
IF(KPRO.EQ.1) THEN
IX=3
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 42
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=3
CALL STFSBL(IX,FLI,FLJ,FIJ,T,IZX,KDMAND,KTULOD,KTTRVL)
GO TO 42
ENDIF

C**
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
IX=3
CALL FSBL(IX,FLI,FLJ,FIJ,T,IZX,KDMAND)
GO TO 42
ENDIF

42 IF(FLIJ.EQ.1) THEN
IVB=1
LC=NR(LRI)+1
LD=NR(LRJ)-2
LB=LD+LC-1
J=3
DO 15 I=LC,LB
ROUTE(LRI,I)=ROUTE(LRJ,J)
ROUTE(LRJ,J)=O
J=J+1
15 CONTINUE
NF(LRI)=NF(LRJ)
NR(LRI)=LB
NR(LRJ)=2
NB(LRJ)=O
NF(LRJ)=O
ELSE
RETURN
ENDIF
IF(DDT.EQ.1) THEN
TDMAND(LRI)=TDMAND(LRI)+TDMAND(LRJ)
TDMAND(LRJ)=O
RETURN
ENDIF
IF(KPRO.EQ.1) THEN
TDMAND(LRI)=KDMAND
TDMAND(LRJ)=O
RETURN
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
TDMAND(LRI)=KDMAND
TDMAND(LRJ)=O
TULOAD(LRI)=KTULOD
TULOAD(LRJ)=0
TTRAVL(LRI)=KTTRVL
TTRAVL(LRJ)=0
RETURN
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
TDMAND(LRI)=KDMAND
TDMAND(LRJ)=C
ENDIF
RETURN
ENDIF

C
IF(LI1.EQ.1.AND.LJ2.EQ.1) THEN
IF(ODT.EQ.1) THEN
IX=4
CALL FEASBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX)
GO TO 43
ENDIF
IF(KPRO.EQ.1) THEN
IX=4
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KOMANO)
GO TO 43
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
IX=4
CALL STFSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND,KTULOD,KTTRVL)
GO TO 43
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
IX=4
CALL FSBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
GO TO 43
ENDIF
43
IF(FIJ.EQ.1) THEN
IWB=1
LC=NR(LRJ)+1
LD=NR(LRI)-2
LB=LD+LC-1
J=3
DO 30 I=LC,LD
ROUTE(LRJ,I)=ROUTE(LRI,J)
ROUTE(LRI,J)=0
J=J+1
30 CONTINUE
NF(LRJ)=NF(LRI)
C** NB(LRJ) DOES NOT CHANGE.


NR(LRJ)=LB
NR(LRI)=2
NB(LRI)=O
NF(LRI)=O
ELSE
RETURN
ENDIF
IF(DDT.EQ.1) THEN
TDMAND(LRJ)=TDMAND(LRJ)+TDMAND(LRI)
TDMAND(LRI)=O
RETURN
ENDIF
IF(KPRO.EQ.1) THEN
TDMAND(LRJ)=KDMAND
TDMAND(LRI)=O
RETURN
ENDIF
IF(SST.EQ.1.AND.IEE.EQ.1) THEN
TDMAND(LRJ)=KDMAND
TDMAND(LRI)=O
TULOAD(LRJ)=KTULOD
TULOAD(LRI)=O
TTRAVL(LRJ)=KTTRVL
TTRAVL(LRI)=O
RETURN
ENDIF
IF(SST.EQ.1.AND.IFF.EQ.1) THEN
ENDIF
ENDIF
ENDIF
RETURN
END

SUBROUTINE FEASBL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX)
INTEGER TCAP,DMAND,TIHAT,TJHAT,ROUTE,FLI,FLJ,FLIJ,FIJ
INTEGER TDMAND,P,PP
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/AS/DMAND(300),TDMAND(100)
COMMON/AS/LJ,LI,LI1,LI2,LU1,LU2,LRI,LRJ
COMMON/AS/TIHAT(5000),TJHAT(5000),ROUTE(100,100)
GO TO (1,2,3,4,5),IX
IYZ = 1
ISET = 1
CALL CHCKK(LRI, IYZ, ISET)
JTJ = TJHAT(T)
KDMAND = TDMAND(LRI) + DMAND(JTJ)
IF (KDMAND .LE. TCAP) THEN
  FLI = 1
ELSE
  FLI = 0
ENDIF
RETURN

IYZ = 1
ISET = 1
CALL CHCKK(LRJ, IYZ, ISET)
ITI = TIHAT(T)
KDMAND = TDMAND(LRJ) + DMAND(ITI)
IF (KDMAND .LE. TCAP) THEN
  FLJ = 1
ELSE
  FLJ = 0
ENDIF
RETURN

IYZ = 1
ISET = 1
CALL CHCKK(LRI, IYZ, ISET)
IYZ = 1
CALL CHCKK(LRJ, IYZ, ISET)
KDMAND = TDMAND(LRI) + TDMAND(LRJ)
IF (KDMAND .LE. TCAP) THEN
  FLIJ = 1
ELSE
  FLIJ = 0
ENDIF
RETURN

IYZ = 1
ISET = 1
CALL CHCKK(LRI, IYZ, ISET)
ISET = 1
IYZ = 1
CALL CHCKK(LRJ, IYZ, ISET)
KDMAND = TDMAND(LRI) + TDMAND(LRJ)
IF (KDMAND .LE. TCAP) THEN
  FLJ = 1
ELSE
  FLJ = 0
ENDIF
RETURN

ITI = TIHAT(T)
JTJ = TJHAT(T)
TDMAND(P) = TDMAND(P) + DMAND(ITI) + DMAND(JTJ)
IZX = 0
IF(TDMAND(P).LE.TCAP) IZX = 1
RETURN
END

C*************************************************************************
C SUBROUTINE CTD
C*************************************************************************
SUBROUTINE CTD(PP, DD, IDD)
INTEGER PP, P, ROUTE, TIHAT, TJHAT
REAL IDD
DIMENSION IDD(100)
COMMON/A4/NB(100), NF(100), NR(100), P
COMMON/A8/DIST(300, 300)
COMMON/A9/TIHAT(5000), TJHAT(5000), ROUTE(100, 100)
DD = 0
DO 10 I = 1, PP
   IDD(I) = 0
   NN = NR(I)
   DO 20 J = 2, NN
      K = ROUTE(I, J)
      L = ROUTE(I, J + 1)
      LL = J + 1
      IF(LL.GT.NN) THEN
         IDD(I) = IDD(I) + DIST(K, 1)
      ELSE
         IDD(I) = IDD(I) + DIST(K, L)
      ENDIF
   20 CONTINUE
   DD = DD + IDD(I)
10 CONTINUE
RETURN
END

C*************************************************************************
C SUBROUTINE CHCKK
C*************************************************************************
SUBROUTINE CHCKK(PP, IYZ, ISET)
INTEGER DMAND, TDMAND, PP, ROUTE, TIHAT, TJHAT, P
REAL IDD
DIMENSION IDD(100)
COMMON/A4/NB(100), NF(100), NR(100), P
COMMON/A5/DMAND(300), TDMAND(100)
COMMON/A9/TIHAT(5000),TJHAT(5000),ROUTE(100, 100)
GO TO (1, 2), IYZ
1
NN=NR(PP)
TDMAND(PP)=O
DO 20 J=3,NN
K=ROUTE(PP,J)
20
TDMAND(PP)=TDMAND(PP)+DMAND(K)
RETURN
C**
2
DO 40 I=ISET,PP
TDMAND(I)=O
NN=NR(I)
DO 30 J=3,NN
K=ROUTE(I,J)
IF(K.EQ.0) GO TO 40
30
TDMAND(I)=TDMAND(I)+DMAND(K)
40 CONTINUE
RETURN
END
********************************************************
C* SUBROUTINE WWRT
********************************************************
C**
SUBROUTINE WWRT(PP)
DIMENSION MOLE(100),IDD(100),TULOAD(100),TTRAVL(100)
INTEGER TULOAD,TTRAVL,DDT,SST,TULDD,TTDD
INTEGER P,ROUTE,TDMAND,DMAND,TIHAT,TJHAT,PP
REAL IDD
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A9/TIHAT(5000),TJHAT(5000),ROUTE(100, 100)
COMMON/A12/DDT,SST,IZAR
COMMON/A13/TULOAO,TTRAVL
COMMON/A15/IIE,IFF,DELTA,IALGOL,BKAMA
COMMON/A16/KPRO,GAMA
WRITE (6,10)
WRITE(10,10)
10 FORMAT(5X,'THE ROUTES ARE THE FOLLOWINGS')
MNK=MNP-1
DO 20 I=1,PP
WRITE(6,30) I,(ROUTE(I,J),J=2,MNK),ROUTE(I,MNP)
WRITE(10,30) I,(ROUTE(I,J),J=2,MNK),ROUTE(I,MNP)
20 CONTINUE
IF(DDT.EQ.1.OR.IIE.EQ.1.OR.KPRO.EQ.1) THEN

30 FORMAT(/5X,'ROUTE ',2X,I3,2X,'-->',2X,30(1X,I2))
CALL CTD(PP, DD, IDD)
ENDIF
IF (IFF.EQ.1) THEN
CALL STCTD(PP, DD)
WRITE(6,41) DD
WRITE(10,41) DD
FORMAT(/30X,'TOTAL ELAPSE TIME OF WHOLE SYSTEM IS',2X,F10.4)
GO TO 42
ENDIF
KKDD=DD
WRITE(6,40) KKDD
WRITE(10,40) KKDD
FORMAT(10X,'TOTAL DISTANCE IS',1X,IB)
DO 50 I=1,PP
MOLI(I)=IDD(I)
WRITE(6,60) I,MOLI(I)
WRITE(10,60) I,MOLI(I)
60 FORMAT(//5X,'DISTANCE ROUTE',2X,I3,'IS',2X,I6)
50 CONTINUE
42 IYZ=2
ISET=1
IF (DDT.EQ.1) THEN
CALL CHCKK(PP, IYZ, ISET)
ENDIF
IF (KPRO.EQ.1) THEN
CALL FCHECK(PP, IYZ, ISET)
GO TO 99
ENDIF
IF (SST.EQ.1) THEN
CALL PRCHK(PP, IYZ, ISET)
TULDD=0
TTDD=0
WRITE(6,51)
WRITE(10,51)
51 FORMAT(/30X,'TOTAL UNLOAD TIME OF EACH ROUTE')
DO 52 I=1,PP
TULDD=TULDD+TULOAD(I)
WRITE(6,53) I,TULOAD(I)
WRITE(10,53) I,TULOAD(I)
53 FORMAT(/30X,'UNLOAD TIME',2X,I3,2X,'IS',2X,I6)
52 CONTINUE
WRITE(6,101) TULDD
WRITE(10,101) TULDD
101 FORMAT(/30X,'TOTAL UNLOAD TIME OF WHOLE SYSTEM=',2X,I6)
WRITE(6,104)
WRITE(10,104)
54 FORMAT(//30X,'TOTAL TRAVEL TIME OF EACH ROUTE')
DO 55 I=1,PP
    TTDD=TTDD+TTRAVL(I)
    WRITE(6,56) I,TTRAVL(I)
    WRITE(10,56) I,TTRAVL(I)
55 CONTINUE
    WRITE(6,102) TTDD
    WRITE(10,102) TTDD
102 FORMAT(//30X,'TOTAL TRAVEL TIME OF WHOLE SYSTEM=',2X,I6)
END IF
99 WRITE(6,70)
    WRITE(10,70)
70 FORMAT(//20X,'TOTAL DEMAND OF EACH ROUTE')
DO 80 I=1,PP
    WRITE(6,90) I,TDMAND(I)
    WRITE(10,90) I,TDMAND(I)
80 CONTINUE
    K=PP
    WRITE(6,100) K
    WRITE(10,100) K
100 FORMAT(//30X,'NUMBER OF THE REQUIRED VEHICLES=',2X,I2)
RETURN
END

C***************************************************************************
C* SUBROUTINE SWTCH
C***************************************************************************
C**
SUBROUTINE SWTCH(IVS,IWS,IYBB,IYBB,PP)
DIMENSION KROOT(100, 100),KNR(100),KNF(100),KNB(100)
INTEGER P,T,R,PP,ROUTE,THAT,THAT,DMAND,TDMAND
INTEGER DDT,SST
COMMON/A4/NB(100),NF(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A6/LI,LJ,LJ1,LJ2,LRI,LRI,LRJ
COMMON/A9/THAT(5000),THAT(5000),ROUTE(100, 100)
COMMON/A12/DDT,SST,IZAR
COMMON/A16/KPRO,GAMA
DO 5 I=1,PP
    DO 15 J=1,20
        KROOT(I,J)=0
        KNR(I)=2
        KNB(I)=0
C** VALUE 20 IN THE ABOVE DD IS STANDING FOR MNP
15 KROOT(I,J)=0
C**
C***************************************************************************
5 KNF(I)=0
MEET=0
DO 10 I=1,PP
IF(IXBB.EQ.1.AND.I.EQ.LRJ) GO TO 10
IF(IYBB.EQ.1.AND.I.EQ.LRJ) GO TO 10
IF(IWB.EQ.1.AND.I.EQ.LRJ) GO TO 10
IF(IWB.EQ.1.AND.I.EQ.LRI) GO TO 10
NRI= NR(I)
MEET=MEET-1
DO 20 J=3,NRI
20 KROOT(MEET,J)=ROUTE(I,J)
KNR(MEET)=NR(I)
KNF(MEET)=NF(I)
KNB(MEET)=NB(I)
10 CONTINUE
DO 50 I=1,PP
NRI=NR(I)
DO 60 J=3,NRI
60 ROUTE(I,J)=O
NR(I)=2
NF(I)=O
NB(I)=O
50 CONTINUE
PP=MEET
P=MEET
DO 30 I=1,MEET
KNRI=KNR(I)
DO 40 J=3,KNRI
40 ROUTE(I,J)=KROOT(I,J)
NR(I)=KNR(I)
NB(I)=KNB(I)
NF(I)=KNF(I)
30 CONTINUE
ISET=1
IYZ=2
IF(DDT.EQ.1) THEN
CALLCHKK(PP,IYZ,ISET)
RETURN
ENDIF
CALLFCHECK(PP,IYZ,ISET)
RETURN
ENDIF
IF(SST.EQ.1) THEN
CALLPRCHK(PP,IYZ,ISET)
ENDIF
RETURN
SUBROUTINE PROB

C***********************************************************
C* SUBROUTINE PROB *
C***********************************************************

INTEGER TIHAT, TJHAT, TI, TJ, NSA, TCAP, X, Y, T, PP, TT
INTEGER ROUTE, TDMAND
INTEGER TULOAD, TTRAVL, VDMAND, DMAND
INTEGER VARS, VIRS, P, UTIME, TTIME, DDT, SST
REAL LOAD, KMAND, LTRAV
DIMENSION NSA(5000), TI(5000), TJ(5000)
DIMENSION TULOAD(100), TTRAVL(100), MEAN(300, 300)
DIMENSION VARS(300, 300), MINE(300), VIRS(300), VDMAND(300)
COMMON/A1/X(300), Y(300)
COMMON/A2/NPT, NW, TCAP, MNP, NTRY
COMMON/A3/MSVA(5000), NSAVE(5000), XX(5000)
COMMON/A4/NB(100), NF(100), NR(100), P
COMMON/A5/DMAND(300), TDMAND(100)
COMMON/A6/LI, LU, LI1, LI2, LJ1, LJ2, LRI, LRJ
COMMON/A7/IBV, IW
COMMON/A8/DIST(300, 300)
COMMON/A9/TIHAT(5000), TJHAT(5000), ROUTE(100, 100)
COMMON/A10/ALPHA, BATA, ATAH, UTIME, TTIME
COMMON/A12/DDT, SST, IZAR
COMMON/A13/TULOAD, TTRAVL
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
COMMON/A15/IEE, IFF, DELTA, IALGOL, BKAMA
GO TO (4, 5, 8, 4), IZAR

4 CALL STINPT
5 CALL SAVMAT
TT=NTRY
CALL TSORT(NSAVE, TIHAT, TJHAT, NTRY)

C** SET TOTAL DEMAND OF EACH ROUTE TO ZERO
C** SET TOTAL TRAVELING TIME OF EACH ROUTE TO ZERO
C** SET TOTAL UNLOADING TIME OF EACH ROUTE TO ZERO

DO 50 I=1, NTRY
50 MSVA(I)=NSAVE(I)

DO 8 P=1, NW
   TDMAND(P)=0
   TTRAVL(P)=0
   TULOAD(P)=0
8 CONTINUE

T=1
P=1
R=3
IX=5
PP=P
ITI=TIHAT(T)
JTJ=TJHAT(T)

C** TO DETERMINE THE DEMAND OF DEMAND POINTS ITI AND JTJ
C**
KMAND=VDMAND(ITI)+VDMAND(JTJ)
KMAND=SQRT(KMAND)
TDMAND(P)=TDMAND(P)+(ATAH • KMAND)+DMAND(ITI)+DMAND(JTJ)

C** TO DETERMINE THE TOTAL UNLOADING TIME OF DEMAND POINTS ITJ, JTJ
LOAD=VIRS(ITI)+VIRS(JTJ)
LOAD=SQRT(LOAD)
TULOAD(P)=TULOAD(P)+(BATA*LOAD)+MINE(ITI)+MINE(JTJ)

C** TO DETERMINE THE TOTAL TRAVELING TIME FOR DEMAND POINTS ITI
C** AND JTJ
C**
LTRAV=VARS(1,ITI)+VARS(JTJ,1)+VARS(ITI,JTJ)
LTRAV=SQRT(LTRAV)
TTRAVL(P)=TTRAVL(P)+MEAN(1,ITI)+MEAN(JTJ,1)+MEAN(ITI,JTJ)
+(ALPHA=LTRAV)
IF(TDMAND(P).LE.TCAP.AND.TULOAD(P).LE.UTIME.AND.TTRAVL(P).LE.TTTIME) THEN
ROUTE(P,R)=TIHAT(T)
NB(P)=ROUTE(P,R)
R=R+1
ROUTE(P,R)=TJHAT(T)
NF(P)=ROUTE(P,R)
NR(P)=R
ENDIF
K=T+1
IF(TDMAND(P).GT.TCAP.OR.TULOAD(P).GT.UTIME.OR.TTRAVL(P).GT.TTTIME) THEN
NSAVE(K-1)=0
TDMAND(P)=0
TTRAVL(P)=0
TULOAD(P)=C
T=K
GO TO 11
ENDIF

C** TO CONSTRUCT A ROUTE
C**
DO 10 T=K,TT
NSAVE(T-1)=0
IYOUTH=1
CALL INTR(IN,PP,T,IYOUTH)
IF(IN.EQ.1) GO TO 10
PP=P
MNK=MNP-1
CALL RTCONT(PP,T)
CONTINUE
IYOUTH=2
CALL INTR(IN,PP,T,IYOUTH)
CALL WWRT(PP)
RETURN
END

SUBROUTINE STINPT
INTEGER DMAND,VDMAND,TCAP,VARS,VIRS,UTIME,TTTIME
INTEGER X,Y,TDMAND
DIMENSION MEAN(300,300) ,VARS(300,300),MINE(300)
DIMENSION VIRS(300),VDMAND(300)
COMMON/A1/X(300),Y(300)
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A3/MSVA(5000),NSAVE(5000),XX(5000)
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A8/DIST(300,300)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A12/DDT,SST,IZAR
COMMON/A14/MEAN,VARS,MINE,VIRS,VDMAND
COMMON/A15/IEE,IFF,DELTA,IALGOL,BKAMA

IDSTB = 1 STANDS FOR DISTRIBUTIONS SUCH AS POISSON, BINOMIAL, EXPONENTIAL, GAMMA, NEGATIVE BINOMIAL, AND CHI-SQUARE
IDSTB=0 STANDS FOR OTHER DISTRIBUTIONS.

IF(IZAR.EQ.4) GO TO 10
WRITE(6,80)
WRITE(10,80)
80 FORMAT(5X,'ENTER THE TYPE OF DISTRIBUTION FUNCTIONS'/' + 'ENTER 1 FOR EXPON, BINOMIAL, CHI-SQUARE, POISSON, NEG-BINO, GAMMA')
WRITE(6,81)
WRITE(10,81)
81 FORMAT(5X,'O OTHERWISE')
204 READ(5,*) IDSTB
WRITE(6,200)
WRITE(10,200)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 204
ENDIF
205
WRITE(6,1)
WRITE(10,1)
FORMAT(SX, '--->', 'ENTER THE NUMBER OF STOP POINTS INCLUDING'/' + 'THE TERMINAL AND TRUCK CAPACITY RESPECTIVELY')
C**
READ(5,*) NPT,TCAP
WRITE(6,101) NPT,TCAP
WRITE(10,101) NPT,TCAP
101
FORMAT(//20X,I5,5X,I5)
WRITE(6,200)
WRITE(10,200)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 205
ENDIF
206
WRITE(6,3)
WRITE(10,3)
3
FORMAT(5X,'--->', 'ENTER THE TOTAL UNLOAD TIME AND TOTAL'/ + 'TRAVELING TIME FOR EACH ROUTE')
READ(5,*) UTIME,TTTIME
WRITE(6,220) UTIME,TTTIME
WRITE(10,220) UTIME,TTTIME
220
FORMAT(2X,'UTIME=','1X,I3,2X,'TTTIME=','1X,I3)
WRITE(6,200)
WRITE(10,200)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 206
ENDIF
C**
C** ALPHA = PROBABILITY OF ROUTE FAILING FOR VIOLATING THE TOTAL
C** TRAVELING TIME.
C** BETA = PROBABILITY OF ROUTE FAILING FOR VIOLATING THE TOTAL
C** UNLOADING TIME
C** ETAH = PROBABILITY OF ROUTE FAILING FOR VIOLATING THE CAPACITY
C**
OF TRUCK.

C**
207 WRITE(6,5)
WRITE(10,5)

5 FORMAT(5X,'ENTER VALUES OF ALPHA,BATA,ATAH RESPECTIVELY')

C
READ(5,*') ALPHA,BATA,ATAH
WRITE(6,11237) ALPHA,BATA,ATAH
WRITE(10,11237) ALPHA,BATA,ATAH

11237 FORMAT(/20X,F10.4,2X,F10.4,2X,F10.4)
WRITE(6,200)
WRITE(10,200)
READ(5,*') ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 207
ENDIF
GO TO 15

10 WRITE(6,2)
WRITE(10,2)

2 FORMAT(/10X,' --->',2X,'ENTER 0 FOR EUCLIDIAN DISTANCE'/
+14X,'1 FOR LINEAR DISTANCE')
READ(5,*') IEUC
IF(IEUC.EQ.1) THEN

209 WRITE(6,12)
WRITE(10,12)

12 FORMAT(5X,' --->','ENTER THE LINEAR DISTANCE OR COST MATRIX')
DO 11 I=1,NPT
READ(5,*') (DIST(I,J),J=1,NPT)
WRITE(6,42) (DIST(I,J),J=1,NPT)
WRITE(10,42) (DIST(I,J),J=1,NPT)
WRITE(6,200)
WRITE(10,200)
READ(5,*') ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 209
ENDIF
CONTINUE

11 CONTINUE
DO 14 I=2,NPT
K=I-1
DO 13 J=1,K
DIST(I,J)=DIST(J,I)
13 CONTINUE
CONTINUE
ELSE
WRITE(6,4)
WRITE(10,4)
WRITE(6,4)
WRITE(10,4)
FORMAT(/10X, '--->', 2X, 'ENTER THE EUCLIDIAN DISTANCE'/
+10X, 'WITH THE COORDINATE OF TERMINAL POINT FIRST')
DO 6 I=1,NPT
READ(5,*) X(I),Y(I)
WRITE(6,9) X(I),Y(I)
WRITE(10,9) X(I),Y(I)
WRITE(6,200)
WRITE(10,200)
READ(5,*) ICORR
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 210
ENDIF
6 CONTINUE
C**
C** WRITE THE EUCLIDIAN DISTANCE
C**
WRITE(6,7)
WRITE(10,7)
FORMAT(10X, 'EUCLIDIAN DISTANCE'/16X,'X',10X,'Y')
DO 8 I=1,NPT
WRITE(6,9) X(I),Y(I)
WRITE(10,9) X(I),Y(I)
9 FORMAT(14X,I4,8X,I4)
8 CONTINUE
C**
C** EVALUATE THE DISTANCE BETWEEN POINTS I AND J
DO 91 I=1,NPT
DO 91 J=1,NPT
IF(I.EQ.J) DIST(I,J)=0
IF(I.GE.J) GO TO 91
SDB=FLOAT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
DIST(I,J)=SQRT(SDB)
DIST(J,I)=DIST(I,J)
91 CONTINUE
ENDIF
IF(IZAR.EQ.4) RETURN
IF(IEE.EQ.1) GO TO 223
CONTINUE
WRITE(6,16)
WRITE(10,16)
16 FORMAT(5X, 'ENTER THE MEAN TRAVEL TIME BETWEEN I AND J')
DO 17 I=1,NPT
11
2.1 READ(9,*) (MEAN(I,J),J=I,NPT)
WRITE(10,45) (MEAN(I,J),J=1,NPT)
WRITE(6,200)
WRITE(10,200)
C
READ(5,*) ICORR
ICORR=1
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 211
ENDIF
CONTINUE
WRITE(6,102)
WRITE(10,102)
102 FORMAT(5X, 'ENTER THE VARIANCE OF TRAVEL TIME BETWEEN I, J')
DO 52 I=1,NPT
212 READ(9,*) (VARS(I,J),J=I,NPT)
WRITE(10,48) (VARS(I,J),J=1,NPT)
WRITE(6,200)
WRITE(10,200)
IF(ICORR.EQ.2) THEN
WRITE(6,201)
WRITE(10,201)
GO TO 212
ENDIF
52 CONTINUE
DO 30 I=2,NPT
K=I-1
DO 31 J=1,K
MEAN(I,J)=MEAN(J,I)
VARS(I,J)=VARS(J,I)
31 CONTINUE
30 CONTINUE
WRITE(6,18)
WRITE(10,18)
18 FORMAT(5X, '--->', 'ENTER THE MEAN AND VARIANCE OF UNLOAD TIME FOR EACH DEMAND POINT I')
DO 19 I=2,NPT
213 READ(9,*) MINE(I),VIRS(I)
ICORR=1
WRITE(6,221) MINE(I),VIRS(I)
WRITE(10,221) MINE(I),VIRS(I)
221 FORMAT(/2X, 'MEAN=',2X.I4, 'VAR=',2X.I4)
WRITE(6,200)
WRITE(10,200)  
IF(ICORR.EQ.2) THEN  
WRITE(6,201)  
WRITE(10,201)  
GO TO 213  
ENDIF  
19 CONTINUE  
WRITE(6,20)  
WRITE(10,20)  
20 FORMAT(5X,'--->','ENTER THE MEAN AND VARIANCE OF THE '+'+9X,'DEMAND POINT I')  
DO 21 I=2,NPT  
214 READ(9,'(*)') DMAND(I),VDMAND(I)  
WRITE(6,222) DMAND(I),VDMAND(I)  
WRITE(10,222) DMAND(I),VDMAND(I)  
222 FORMAT(2X,'DMAND=',1X,I4,2X,'VDMAND=',2X,I4)  
WRITE(6,200)  
WRITE(10,200)  
ICORR=1  
IF(ICORR.EQ.2) THEN  
WRITE(6,201)  
WRITE(10,201)  
GO TO 214  
ENDIF  
21 CONTINUE  
IF(IEE.EQ.1) GO TO 10  
223 WRITE(6,22)  
WRITE(10,22)  
IF(IEE.EQ.1) THEN  
22 FORMAT}//15X,'DISTANCE MATRIX')  
DO 25 I=1,NPT  
WRITE(6,42) (DIST(I,J),J=1,NPT)  
WRITE(10,42) (DIST(I,J),J=1,NPT)  
42 FORMAT(1X,15F5.1)  
25 CONTINUE  
ENDIF  
WRITE(6,43)  
WRITE(10,43)  
43 FORMAT}//15X,'MEAN TRAVEL TIME')  
45 FORMAT(1X,20I4)  
WRITE(6,46)  
WRITE(10,46)  
46 FORMAT}//15X,'VARIANCE TRAVEL TIME')  
48 FORMAT(1X,20I4)  
WRITE(6,26)  
WRITE(10,26)
DO 27 I=2,NPT
WRITE(6,28) I,DMAND(I),VDMAND(I),MINE(I),VIRS(I)
WRITE(10,28) I,DMAND(I),VDMAND(I),MINE(I),VIRS(I)
28 FORMAT(10X, I3, 2X, I5, 19X, I5, 19X, I5, 19X, I5)
27 CONTINUE
IF(IDSTB.EQ.1) GO TO 50
ITDD=0
IVDD=0
DO 41 I=2,NPT
ITDD=ITDD+DMAND(I)
IVDD=IVDD+VDMAND(I)
41 CONTINUE
BB=FLOAT(IVDD)
IVDD=SQRT(BB)
IVDD=ATAH*IVDD
ITOTAL=IVDD+ITDD
NW=(ITOTAL/TCAP)+10
NNW=NW-10
MNP=(NPT/NNW)+1
MNP=MNP+5
GO TO 99
50 WRITE(6,777)
WRITE(10,777)
777 FORMAT(5X, 'ENTER THE VALUE OF SAI')
READ(*,*) ISAI
BK=(ATAH**4)*ISAI**2+4*TCAP*(ATAH**2)*ISAI
BKK=SQRT(BK)
TCAPBR=2*TCAP-(ATAH**2)*ISAI-BKK)**.5
WRITE(6,61) TCAPBR
WRITE(10,61) TCAPBR
61 FORMAT(/15X, 'ARTIFICIAL CAPAVITY OF TRUCK=', F8.3)
C**
C** TOTAL MEAN DEMAND ON EACH ROUTE MUST BE LESS THAN TCAP=TCAPBR
C**
ITDD=0
DO 51 I=2,NPT
ITDD=ITDD+DMAND(I)
51 CONTINUE
NW=(ITDD/TCAP)+10
NNW=NW-10
MNP=(NPT/NNW)+1
MNP=MNP+5
99 CONTINUE
200 FORMAT(/2X, 'CORRECT', 2X, '1:YES', 2X, '2:NO')
201 FORMAT(/2X, 'REENTER AGAIN')
RETURN
END

SUBROUTINE PRCHCK
C*******************************************************************************
SUBROUTINE PRCHCK(PP, IYZ, ISET)
INTEGER DMAND, TDMAND, PP, ROUTE, TIHAT, TJHAT, P
INTEGER VDMAND, MEAN, VARS, MINE, VIRS
REAL KVD, KVRS, KWVARS
INTEGER TULOAD, TTRAVL, UTIME, TTIME
DIMENSION IDD(100), VDMAND(300), MEAN(300, 300)
DIMENSION VARS(300, 300), MINE(300), VIRS(300)
DIMENSION TULOAD(100), TTRAVL(100)
COMMON/A4/NB(100), NF(100), NR(100), P
COMMON/A5/DMAND(300), TDMAND(100)
COMMON/A6/LI, LJ, LI1, LI2, LJ1, LJ2, LRI, LRJ
COMMON/A9/TIHAT(5000), TJHAT(5000), ROUTE(100, 100)
COMMON/A10/ALPHA, BATA, ATAH, UTIME, TTIME
COMMON/A11/VD, IMEAN, VVRS, KMP, WVARS, KMON
COMMON/A13/TULOAD, TTRAVL
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
GO TO (1, 2), IYZ
1
NN=NR(PP)
TDMAND(PP)=0
TTRAVL(PP)=0
TULOAD(PP)=0
VD=0
DO 20 J=3, NN
K=ROUTE(PP, J)
20 VD=VD+VDMAND(K)
KVD=SQRT(VD)
KVD=ATAH*KVD
DO 30 J=3, NN
K=ROUTE(PP, J)
30 TDMAND(PP)=TDMAND(PP)+DMAND(K)
IMEAN=TDMAND(PP)
C** TOTAL DEMAND OF ROUTE PP, Considering MEAN AND VARIANCE
TDMAND(PP)=TDMAND(PP)+KVD
C** TO CALCULATE TOTAL STANDARD DEVIATION OF UNLOAD TIME
VVRS=0
DO 40 J=3, NN
K=ROUTE(PP, J)
40 VVRS=VVRS+VIRS(K)
KVRS=SQRT(VVRS)
KVRS=ATAH*KVRS
DO 50 J=3,NN
   K=ROUTE(PP,J)
   TULOAD(PP)=TULOAD(PP)+MINE(K)
   KMP=TULOAD(PP)
C** TOTAL UNLOAD TIME CONSIDERING MEAN AND VARIANCE
   TULOAD(PP)=TULOAD(PP)+KVRS
C*** TO CALCULATE TOTAL TRAVEL TIME OF ROUTE PP.
C**
   VWARS=0
   DO 60 J=3,NN
      K=J-1
      KA=ROUTE(PP,K)
      KB=ROUTE(PP,J)
      VWARS=WVARS+VARS(KA,KB)
   KG=ROUTE(PP,NN)
   VWARS=WVARS+VARS(KG,1)
   KWVARS=SQRT(WVARS)
   KWVARS=ALPHA*KWVARS
   KMEN=0
   DO 70 J=3,NN
      K=J-1
      LA=ROUTE(PP,K)
      LB=ROUTE(PP,J)
      KMEN=KMEN+MEAN(LA,LB)
      LG=ROUTE(PP,NN)
      KMEN = KMEN+MEAN(LG,1)
      TTRAVL(PP)=TTRAVL(PP)+KMEN+KWVARS
   RETURN
C*****
C***
   2 DO 80 I=ISET,PP
      TDMAND(I)=0
      NN=NR(I)
      VD=0
      DO 90 J=3,NN
         K=ROUTE(I,J)
         VD=VD+VDMAND(K)
         KVD=SQRT(VD)
         KVD=ATAN*KVD
         DO 100 J=3,NN
            K=ROUTE(I,J)
      TDMAND(I)=TDMAND(I)+DMAND(K)
      TDMAND(I)=TDMAND(I)+KVD
     80 CONTINUE
C** TO FIND THE TOTAL UNLOAD TIME OF EACH CONSTRUCTED ROUTE
DO 110 I=ISET,PP
     TULOAD(I)=0
     NN=NR(I)
     VVRS=0
     DO 120 J=3,NN
         K=ROUTE(I,J)
     120
         VVRS=VVRS+VIRS(K)
         KVVRS=SQRT(VVRS)
         KVVRS=BATA*KVVRS
         DO 130 J=3,NN
             K=ROUTE(I,J)
         130
             TULOAD(I)=TULOAD(I)+MINE(K)
     TULOAD(I)=TULOAD(I)+KVVRS
     110 CONTINUE
C** TO FIND THE TOTAL TRAVEL TIME FOR EACH CONSTRUCTED ROUTE
C**
     DO 140 I=ISET,PP
         TTRAVL(I)=0
         NN=NR(I)
C** TO CALCULATE TOTAL VARIANCE OF TRAVEL TIME FOR ROUTE I
     WVARS=0
     DO 150 J=3,NN
         K=J-1
         KA=ROUTE(I,K)
         KB=ROUTE(I,J)
     150
         WVARS=WVARS+VARS(KA,KB)
         KG=ROUTE(I,NN)
         WVARS=WVARS+VARS(KG,1)
         KWVARS=SQRT(WVARS)
         KWVARS=ALPHA*KWVARS
C** TO CALCULATE TOTAL MEAN TRAVEL TIME OF ROUTE I
     KMEN=0
     DO 160 J=3,NN
         K=J-1
         LA=ROUTE(I,K)
         LB=ROUTE(I,J)
     160
         KMEN=KMEN+MEAN(LA,LB)
         LG=ROUTE(I,NN)
         KMEN=KMEN+MEAN(LG,1)
C**
C** TO FIND TOTAL MEAN TRAVEL TIME OF ROUTE I
     KMON=KMEN+MEAN(LG,1)
     TTRAVL(I)=TTRAVL(I)+KMON+KWVARS
140 CONTINUE
RETURN
END
C******************************************************************************
SUBROUTINE STFSBL

DIMENSION MEAN(300,300), VARS(300,300), MINE(300)
DIMENSION TULOAO(100), TTRAVL(100)
DIMENSION VIRS(300), VDMAND(300)
INTEGER TCAP, DMAND, TIHAT, TJHAT, ROUTE, FLI, FLJ, FLIJ, FIJ
INTEGER TDMAND, P, T, PP, UTIME, TTTIME
INTEGER VARS, VIRS, VDMAND, TULOAO, TTRAVL
REAL KVD, KVRS, LOAD, KMAND, LTRAV, KWVARS
COMMON/A2/NPT, NW, TCAP, MNP, NTRY
COMMON/A4/NE(100), NF(100), NR(100), P
COMMON/A5/DMAN0(300), TDMAND(100)
COMMON/A6/LI, LJ, LI1, LJ1, LJ2, LRI, LRJ
COMMON/A10/TIHAT(5000), TJHAT(5000), ROUTE(100, 100)
COMMON/A11/VD, IMEM, VVRS, KMP, WVARS, KMON
COMMON/A13/TULOAO, TTRAVL
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
ITI = TIHAT(T)
JTJ = TJHAT(T)
GO TO (1, 2, 3, 4)

1
IYZ = 1
ISET = 1
CALL PRCHCK(LRI, IYZ, ISET)
IRAS = ITI
IRAK = JTJ
KRAFT = LRI
CALL CONTRL(IRAS, IRAK, KRAFT, KMAND, KTULOD, KTTRVL)
IF(KDMAND.LE.TCAP.AND.KTULOD.LE.UTIME.AND.KTTRVL.LE.TTTIME)
+ THEN
FLI = 1
ELSE
FLI = 0
ENDIF
RETURN

2
IYZ = 1
ISET = 1
CALL PRCHCK(LRJ, IYZ, ISET)
IRAK = ITI
IRAS = JTJ
KRAFT = LRJ
CALL CONTRL(IRAS, IRAK, KRAFT, KMAND, KTULOD, KTTRVL)
IF(KDMAND.LE.TCAP.AND.KTULOD.LE.UTIME.AND.KTTRVL.LE.TTTIME)
+ THEN
FLJ=1
ELSE
FLJ=0
ENDIF
RETURN
3
IYZ=1
ISET=1
CALL PROCHK(LRI,IYZ,ISET)
M1=IMEAN
V1=VD
N1=KMP
W1=VVRS
MYN1=KMON
VD1=WVARS
CALL PROCHK(LRJ,IYZ,ISET)
M2=IMEAN
V2=VD
N2=KMP
W2=VVRS
MYN2=KMON
VD2=WVARS
CALL SOFT(KTTRVL,VD1,VD2,MYN1,MYN2,ITI,JTI)
CALL RUSH(M1,M2,N1,N2,V1,V2,W1,W2,KDMAND,KTULOD)
IF(KDMAND.LE.TCAP.AND.KTULOD.LE.UTIME.AND.KTTRVL.LE.TTIME)
+ THEN
FLIJ=1
ELSE
FLIJ=0
ENDIF
RETURN
4
IYZ=1
ISET=1
CALL PROCHK(LRI,IYZ,ISET)
M1=IMEAN
V1=VD
N1=KMP
W1=VVRS
MYN1=KMON
VD1=WVARS
CALL PROCHK(LRJ,IYZ,ISET)
M2=IMEAN
V2=VD
N2=KMP
W2=VVRS
MYN2=KMON
VD2=WVARS
CALL SOFT(KTTRVL,VD1,VD2,MYN1,MYN2,ITI,ITJ)
CALL RUSH(M1,M2,N1,N2,V1,V2,W1,W2,KDMAND,KTULOD)
IF(KDMAND.LE.TCAP.AND.KTULOD.LE.UTIME.AND.KTTRVL.LE.TTIME) THEN
  FIJ=1
ELSE
  FIJ=0
ENDIF
RETURN
END

SUBROUTINE CONTRL
INTEGRAL DMAND,VDMAND,TCAP,VARS,VIRS,UTIME,TTTIME
REAL KVD,KVVRS,KVVRSU
INTEGER TULOAD,TTRAVL,TDMAND
DIMENSION MEAN(300,300),VARS(300,300),MINE(300)
DIMENSION VIRS(300),VDMAND(300)
DIMENSION TULOAD(100),TTRAVL(100)
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A11/VD,IMEAN,VVRS,KMP,WVARS,KMON
COMMON/A13/TULOAD,TTRAVL
COMMON/A14/MEAN.VARS,MINE.VIRS,VDMAND

C** TO EVALUATE THE VARIANCE OF DEMAND
VDNEW=VD+VDMAND(IRAK)

C** TO EVALUATE THE VARIANCE OF UNLOADING TIME
VVRSNU=VVRS+VIRS(IRAK)

C** TO EVALUATE THE VARIANCE OF TRAVEL TIME
WVARSU=WVARS+VARS(1,IRAK)+VARS(IRAS,IRAK)-VARS(IRAS,1)

KVD=SQRT(VDNEW)
KVD=ATAH*KVD
KDMAND=IMEAN+DMAND(IRAK)+KVD

KVVRS=SQRT(VVRSNU)
KVVRS=BATA*KVVRS
KTULOD=KMP+MINE(IRAK)+KVVRS

C**
KWVARS = SQRT(WVARS)
KWVARS = ALPHA * KWVARS
KTTRVL = KMON + MEAN(1,IRAK) + MEAN(IRAS,IRAK) + KWVARS - MEAN(IRAS,1)
RETURN
END

SUBROUTINE SOFT(KTTRVL, VD1, VD2, MYN1, MYN2, ITI, JTJ)
INTEGER VARS, VIRS, VDMAND
REAL KWVARS
DIMENSION MEAN(300, 300), VARS(300, 300)
DIMENSION MINE(300), VIRS(300), VDMAND(300)
COMMON/A10/ALPHA, BATA, ATAH, UTIME, TTIME
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
MEN = MEAN(1, ITI) + MEAN(1, JTJ)
MENS = MEN - MEAN(ITI, JTJ)
MEANTL = MYN1 + MYN2 - MENS
V03 = VARS(1, ITI) + VARS(1, JTJ)
SVD3 = VD3 - VARS(ITI, JTJ)
VD = VD1 + VD2 - SVD3
KWVARS = SQRT(VD)
KWVARS = ALPHA * KWVARS
KTTRVL = MEANTL + KWVARS
RETURN
END

SUBROUTINE RUSH(M1, M2, N1, N2, V1, V2, W1, W2, KDMAND, KTULOD)
COMMON/A10/ALPHA, BATA, ATAH, UTIME, TTIME
REAL IV3, IW3
M3 = M1 + M2
V3 = V2 + V1
IV3 = SQRT(V3)
IV3 = ATAH * IV3
KDMAND = M3 + IV3
N3 = N1 + N2
W3 = W1 + W2
IW3 = SQRT(W3)
IW3 = BATA * IW3
KTULOD = N3 + IW3
RETURN
END

SUBROUTINE STSAVE
C**************************************************************
C SUBROUTINE SOFT
C**************************************************************
C SUBROUTINE RUSH
C**************************************************************
C SUBROUTINE STSAVE
C* THIS SUBROUTINE CONSTRUCT THE SAVING MATRIX FOR PROBLEM F WHEN TIME IS CONSIDERED.

DIMENSION VARS(300,300), MINE(300), VIRS(300), VDMAND(300)
DIMENSION MEAN(300,300), ISAVE(300,300), WAR(100,100)
DIMENSION MAR(100,100)
INTEGER TCAP, X, P, TIHAT, TJHAT, ROUTE, VARS, VIRS, VDMAND
COMMON/A2/NPT, NW, TCAP, MNP, NTRY
COMMON/A3/MSVA(5000), NSAVE(5000), XX(5000)
COMMON/A4/NB(100), NF(100), NR(100), P
COMMON/AB/DIST(300,300)
COMMON/A9/TVHAT(5000), TJHAT(5000), ROUTE(100,100)
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
COMMON/A15/IEE, IFF, DELTA, IALGOL, BKAMA
IF (IALGOL.EQ.2) THEN
  ISIGMA=0
  DO 10 I=2,NPT
  DO 10 J=2,NPT
  ISIGMA=ISIGMA+VARS(I,J)
10 CONTINUE
  K=NPT-1
  KK=K*K
  IBAR=ISIGMA/(KK*DELTA)
ENDIF
  DO 15 I=2,NPT
  DO 15 J=2,NPT
  WAR(I,J)=0
  MAR(I,J)=0
15 CONTINUE
  DO 20 I=2,NPT
  DO 30 J=I,NPT
    C PURPOSE TO DETERMINE THE LIST OF SAVINGS FOR BOTH ALGORITHMS OF "F" TYPE PROBLEM
    ISAVE(I,J)=-99999
    IF (MEAN(I,J).EQ.0) GO TO 40
    MAR(I,J)=MEAN(I,1)+MEAN(1,J)-MEAN(I,J)
  C PURPOSE TO DETERMINE THE SAVINGS FOR ALGORITHM(I)
  IF (IALGOL.EQ.1) THEN
    MAR(I,J)=BKAMA*MAR(I,J)
  ENDIF
C 40 IF (VARS(I,J).EQ.0) GO TO 50
  WAR(I,J)=VARS(I,1)+VARS(1,J)-VARS(I,J)
C 50 IF (IALGOL.EQ.1) THEN
  WAR(I,J)=(1-BKAMA)*SORT(WAR(I,J))
END IF

50  MAR(J,I)=MAR(I,J)
   WAR(J,I)=WAR(I,J)
   IF(WAR(I,J).EQ.0) GO TO 30
   IF(IALGOL.EQ.2) THEN
      ISAVE(I,J)=MAR(I,J)+IBAR/SQRT(WAR(I,J))
   ENDIF
   IF(IALGOL.EQ.1) THEN
      ISAVE(I,J)=MAR(I,J)+WAR(I,J)
   ENDIF
   ISAVE(J,I)=ISAVE(I,J)
30  CONTINUE
20  CONTINUE
   IF(IALGOL.EQ.1.AND.SKAMA.EQ.1) THEN
      ISAVE(I,J)=MAR(I,J)
      ISAVE(J,I)=ISAVE(I,J)
   ENDIF
   DO 60 I=2,NPT
      ISAVE(I,1)=-99999
   CONTINUE
C**
C**"PUT ARRAY ISAVE INTO A ONE DIMENSIONAL ARRAY
L=O
   IPT=NPT-1
   DO 100 I=2,IPT
      K=I+1
      DO 100 J=K,NPT
      IF(ISAVE(I,J).LE.0) GO TO 100
      L=L+1
      NSAVE(L)=ISAVE(I,J)
      TIHAT(L)=I
      TJHAT(L)=J
100  CONTINUE
      NTRY=L
   DO 170 I=1,NW
      ROUTE(I,1)=I
      ROUTE(I,MNP)=1
      ROUTE(I,2)=1
      MNP1=MNP-1
      DO 180 J=3,MNP1
      ROUTE(I,J)=O
      NB(I)=O
NF(I)=0
NR(I)=2
CONTINUE
DO 13 I=1,NW
WRITE(6,23) (ROUTE(I,J),J=1,MNP)
23 FORMAT(5X,30(I2.2X))
CONTINUE
RETURN
END

SUBROUTINE FSSL(IX,FLI,FLJ,FLIJ,FIJ,T,IZX,KDMAND)
DIMENSION MEAN(300,300),VARS(300,300)
DIMENSION MINE(300),TULOAD(100),TTRAVL(100),VIRS(300)
DIMENSION VDMAND(300)
INTEGER TCAP,DMAND,TIHAT,TJHAT,ROUTE,FLI,FLJ,FLIJ,FIJ
INTEGER TDMAND,P,T,PP,XX,UTIME,TTTIME
INTEGER VARS,VIRS,VDMAND,TULOAD,TTRAVL
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A3/MSVA(5000),NSAVE(5000),XX(5000)
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A6/LI,LJ,LI1,LI2,LJ1,LJ2,LRI,LRJ
COMMON/A7/IBV,IWB
COMMON/A8/TIHAT(5000),TJHAT(5000),ROUTE(100,100)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A11/VD,IMEAN,VVRS,KMP,WVARS,KMON
COMMON/A13/TULOAD,TTRAVL
COMMON/A14/MEAN,VARS,MINE,VIRS,VDMAND
COMMON/A15/IEE,IFF,DELTA,IALGDL,BKAMA
COMMON/A16/KPRO,GAMA
C PURPOSE OF THIS SUBROUTINE IS TO DETERMINE THE FEASIBILITY
C FOR THE SVRP WHEN CUSTOMER DEMANDS ARE ONLY PROBABILISTIC
IZX=1
ISET=1
ITI=TIHAT(T)
JTJ=TJHAT(T)
GO TO (1,2,3,4),IX
1 CALL FCHECK(LRI,IZX,ISET)
IRAK=ITI
IRAS=JTJ
KRAFT=LRI
CALL STCONT(IRAS,IRAK,KRAFT,KDMAND)
IF(KDMAND.LE.TCAP) THEN
FLI=1
ELSE
FLJ=O
ENDIF
RETURN

2 CALL FCHECK(LRJ,IYZ,ISET)
IRAS=ITI
IRAK=JTJ
KRAFT=LRJ
CALL STCONT(IRAS,IRAK,KRAFT,KDMAND)
IF(KDMAND.LE.TCAP) THEN
FLJ=1
ELSE
FLJ=O
ENDIF
RETURN

3 CALL FCHECK(LRJ,IYZ,ISET)
M1=IMEAN
V1=VD
CALL FCHECK(LRJ,IYZ,ISET)
M2=IMEAN
V2=VD
CALL SFAST(M1,M2,V1,V2,KDMAND)
IF(KDMAND.LE.TCAP) THEN
FLJ=1
ELSE
FLJ=O
ENDIF
RETURN

C***
C***
4 CALL FCHECK(LRJ,IYZ,ISET)
M1=IMEAN
V1=VD
CALL FCHECK(LRJ,IYZ,ISET)
M2=IMEAN
V2=VD
CALL SFAST(M1,M2,V1,V2,KDMAND)
IF(KDMAND.LE.TCAP) THEN
FLJ=1
ELSE
FLJ=O
ENDIF
RETURN
END

C**********************************************************************
C* SUBROUTINE FCHECK
*
SUBROUTINE FCHECK(PP, IYZ, ISET)
DIMENSION IDD(100), VDMAND(300), MEAN(300, 300)
DIMENSION VARS(300, 300), MINE(300), VIRS(300)
DIMENSION TULDAD(100), TTRAVL(100)
INTEGER DMAND, TDMAND, PP, ROUTE, TIHAT, TJHAT, P
INTEGER VDMAND, MEAN, VARS, MINE, VIRS
INTEGER TULDAD, TTRAVL, UTIME, TTTIME
REAL KVD
COMMON/A4/NB(100), NF(100), NR(100), P
COMMON/A5/ DMAND(300), TDMAND(100)
COMMON/A6/LI, LJ, LI1, LJ1, LJ2, LRI, LRJ
COMMON/A9/TIHAT(5000), TJHAT(5000), ROUTE(100, 100)
COMMON/A10/ALPHA, BATA, ATAH, UTIME, TTTIME
COMMON/A11/VD, IMEAN, VVRS, KMP, WVARS, KMOM
COMMON/A13/TULDAD, TTRAVL
COMMON/A14/MEAN, VARS, MINE, VIRS, VDMAND
COMMON/A15/ IEE, IFF, DELTA, IALGOL, BKAMA
COMMON/A16/KPRO, GAMA
IF(KPRO.EQ. 1) ATAH=GAMA

C** TOTAL DEMAND OF ROUTE PP CONSIDERING MEAN AND VARIANCE
C** OF ALL DEMAND POINTS LOCATED ON THIS ROUTE.
C
GO TO (1, 2), IYZ
1
NN=NR(PP)
TDMAND(PP)=0
VD=0
DO 10 J=3, NN
  K=ROUTE(PP, J)
  VD=VD+VDMAND(K)
10
KVD=SQRT(VD)
  KVD=ATAH*KVD
DO 20 J=3, NN
  K=ROUTE(PP, J)
  TDMAND(PP)=TDMAND(PP)+DMAND(K)
  IMEAN=TDMAND(PP)
20
RETURN
DO 80 I=ISET, PP
  TDMAND(I)=0
  NN=NR(I)
  VD=0
DO 90 J=3, NN
  K=ROUTE(I, J)
90
  VD=VD+VDMAND(K)
KVD=SQRT(VD)
RETURN
$KVD = \text{ATAH} \times KVD$

DO 100 J=3,NN
  K=ROUTECI(J)
100  TDMAND(I)=TDMAND(I)+DMAND(K)

TDMAND(I)=TDMAND(I)+KVD

CONTINUE
RETURN
END

C***************************************************************
C SUBROUTINE SFAST
C***************************************************************
SUBROUTINE SFAST(M1,M2,V1,V2,KDMAND1
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
C***
REAL IV3
COMMON/A16/KPRG,GAMA
IF(KPRG.EQ.1) ATAH=GAMA
M3=M1+M2
V3=V1+V2
IV3=SQRT(V3)
IV3=ATAH*IV3
KDMAND=M3+IV3
RETURN
END

C***************************************************************
C SUBROUTINE STCONT
C***************************************************************
SUBROUTINE STCONT(IRAS,IRAK,KRAFT,KDMAND)
DIMENSION MEAN(300,300),VARS(300,300),MINE(300)
DIMENSION VIRS(300),VDMAND(300),TLOAD(100),TTRAVL(100)
INTEGER DMAND,VDMAND,TCAP,VARS,VIRS,UTIME,TTTIME
INTEGER TLOAD,TTRAVL,TDMAND
REAL KVD
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A11/VD,IMEAN,VVRS,KMP,WWARS,VMON
COMMON/A13/TLOAD,TTRAVL
COMMON/A14/MEAN,VARS,MINE,VIRS,VDMAND
COMMON/A15/IEE,IFF,DELTA,IALGOL,BKAMA
COMMON/A16/KPRG,GAMA
IF(KPRG.EQ.1) ATAH=GAMA
C**
C** TO EVALUATE THE VARIANCE OF DEMAND
VDNEW=VD+VDMAND(IRAS)
KVD=SQRT(VDNEW)
KVD=ATAH*KVD
KDMAND=IMEAN+DMAND(IRAS)+KVD
RETURN
END

C***********************************************************************
C*SUBROUTINE STATS
C***********************************************************************

SUBROUTINE STATS

DIMENSION MEAN(300,300),VARS(300,300),MINE(300)
DIMENSION TULOAD(100),TTRAVL(100),VIRS(300)
DIMENSION VDMAND(300)
INTEGER TCAP,DMAND,TIHAT,TJHAT,ROUTE,FLI,FLJ,FIJ
INTEGER TDMAND,P,T,PP,TT,XX,UTIME,TTTIME
INTEGER VARS,VIRS,VDMAND,TULOAD,TTRAVL,DDT,SST
REAL KMAND
COMMON/A2/NPT,NW,TCAP,MNP,NTRY
COMMON/A3/MSVA(5000).NSAVE(5000).XX(5000)
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A6/LI,LU,L1,L2,LJ1,LJ2,LRI,LRJ
COMMON/A7/IBV,IBW
COMMON/A8/TIHAT(5000),TJHAT(5000),ROUTE(100,100)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A11/VD,IMEAN,VVRS,KMP,WVARS,KMON
COMMON/A13/TULOAD,TTRAVL
COMMON/A14/MEAN,VARS,MINE,VIRS,VDMAND
COMMON/A15/IIE,IFF,DELTA,IALGDL,BKAMA
COMMON/A12/DDT,SST,IZAR
COMMON/A16/KPRO,GAMA
COMMON/A17/KDMAND,KTULOD,KTTRVL

IF(KPRO.EQ.1) THEN
GO TO (20,21,22),IZAR
20 WRITE(6,4)
4 FORMAT(//10X,'---•',2X,'ENTER 0 FOR EUCLIDIAN DISTANCE AND +1 FOR LINEAR DISTANCE')
READ(5,*) IDDT
CALL INPT(IDDT)
21 CALL SAVMAT
GO TO 9
ENDIF

GO TO (30,31,22),IZAR
30 CALL STINPT
31 CALL STSAVE
9 TT=NTRY
CALL TSORT(NSAVE,TIHAT,TJHAT,NTRY)
C** SET THE TOTAL DEMAND OF EACH ROUTE TO ZERO
50  MSVA(I)=NSAVE(I)
22  DO 7 P=1,NW
  TDMAND(P)=0
7  CONTINUE
T=1
11  P=1
  R=3
  PP=P
  ITI=TIHAT(T)
  JTJ=TJHAT(T)
C** TO DETERMINE THE DEMAND POINTS OF ITI AND JTJ.
C**
  IF(KPRO.EQ.1) ATAH=GAMA
  KMAND=VDMAND(ITI)+VOMAND(JTJ)
  KMAND=SQRT(KMAND)
  TDMAND(P)=TDMAND(P)+(ATYPE*KMAND)+DMAND(ITI)+DMAND(JTJ)
  IF(TDMAND(P).LE.TCAP) THEN
    ROUTE(P,R)=TIHAT(T)
    NB(P)=ROUTE(P,R)
    R=R+1
    ROUTE(P,R)=TJHAT(T)
    NF(P)=ROUTE(P,R)
    NR(P)=R
    ENDIF
  K=T+1
  IF(TDMAND(P).GT.TCAP) THEN
    NSAVE(K-1)=0
    TDMAND(P)=0
    T=K
    GO TO 11
  ENDIF
C** TO CONSTRUCT A ROUTE
10  CONTINUE
   IYOUTH=2
   CALL INTR(IN,PP,T,IYOUTH)
   IF(IN.EQ.1) GO TO 10
   PP=P
   MNK=MNP-1
   CALL RTCONT(PP,T)
   CONTINUE
   IYOUTH=1
   CALL INTR(IN,PP,T,IYOUTH)
CALL WWRT(PP)
RETURN
END

C**************************************************
SUBROUTINE STCTD
C**************************************************

SUBROUTINE STCTD(PP,TTOTAL )
DIMENSION IDD(100),VDMAND(300),MEAN(300,300)
DIMENSION TTRAVL(100),VARS(300,300),MINE(300)
DIMENSION VIRS(300),TULOAD(100)
INTEGER TULOAD,TTRAVL,UTIME,TTTIME,VDMAND,MEAN,VARS
INTEGER MINE,VIRS,DMAND,PP,ROUTE,P
INTEGER TIHAT,TJHAT
COMMON/A4/NB(100),NF(100),NR(100),P
COMMON/A5/DMAND(300),TDMAND(100)
COMMON/A6/LI,LJ,LI1,LI2,LU1,LU2,LR1,LRJ
COMMON/A9/THAT(5000),TJHAT(5000),ROUTE(100,100)
COMMON/A10/ALPHA,BATA,ATAH,UTIME,TTTIME
COMMON/A11/VD,IMEAN,VVRS,KMP,WVARS,KMON
COMMON/A13/TULO,TTULO,TTRAVL
COMMON/A14/MEAN,VARS,MINE,VIRS,VDMAND

TTOTAL=0.
IYZ=2
ISET=1
CALL PRCHCK(PP,IYZ,ISET)
DO 10 I=1,PP
   TTOTAL=TTOTAL+TULOAD(I)+TTRAVL(I)
10 CONTINUE
RETURN
END

C
APPENDIX C

DATA FOR TEST PROBLEMS 1, 2, AND 3
TABLE XXIV

50 NODE PROBLEM

TEST PROBLEM #1

<table>
<thead>
<tr>
<th>i</th>
<th>x_i</th>
<th>y_i</th>
<th>\mu_i</th>
<th>\sigma_i^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>52</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>49</td>
<td>30</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>52</td>
<td>64</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>26</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>30</td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>47</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>63</td>
<td>19</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td>62</td>
<td>23</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>52</td>
<td>33</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>51</td>
<td>21</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>42</td>
<td>41</td>
<td>19</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>31</td>
<td>32</td>
<td>29</td>
<td>53</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>25</td>
<td>23</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>42</td>
<td>21</td>
<td>49</td>
</tr>
<tr>
<td>15</td>
<td>36</td>
<td>16</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
<td>41</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>17</td>
<td>27</td>
<td>23</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>33</td>
<td>41</td>
<td>67</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>13</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>57</td>
<td>58</td>
<td>28</td>
<td>87</td>
</tr>
<tr>
<td>21</td>
<td>62</td>
<td>42</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>42</td>
<td>57</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>57</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>52</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>38</td>
<td>28</td>
<td>16</td>
</tr>
</tbody>
</table>

Central Depot is at \( x_o = 30, \ y_o = 40 \)

Vehicle Capacity is \( Q = 160 \)
### TABLE XXV

#### 75 NODE PROBLEM

**TEST PROBLEM #2**

<table>
<thead>
<tr>
<th>i</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$\mu_i$</th>
<th>$\sigma_i^2$</th>
<th>i</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$\mu_i$</th>
<th>$\sigma_i^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>22</td>
<td>18</td>
<td>7</td>
<td>44</td>
<td>21</td>
<td>48</td>
<td>17</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>26</td>
<td>26</td>
<td>27</td>
<td>45</td>
<td>50</td>
<td>30</td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>45</td>
<td>11</td>
<td>2</td>
<td>46</td>
<td>51</td>
<td>42</td>
<td>27</td>
<td>46</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>35</td>
<td>30</td>
<td>18</td>
<td>47</td>
<td>50</td>
<td>15</td>
<td>19</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>55</td>
<td>20</td>
<td>21</td>
<td>28</td>
<td>48</td>
<td>48</td>
<td>21</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>33</td>
<td>34</td>
<td>19</td>
<td>23</td>
<td>49</td>
<td>12</td>
<td>38</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td>50</td>
<td>15</td>
<td>5</td>
<td>50</td>
<td>15</td>
<td>56</td>
<td>22</td>
<td>54</td>
</tr>
<tr>
<td>8</td>
<td>55</td>
<td>45</td>
<td>16</td>
<td>28</td>
<td>51</td>
<td>29</td>
<td>39</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>26</td>
<td>59</td>
<td>29</td>
<td>17</td>
<td>52</td>
<td>54</td>
<td>38</td>
<td>19</td>
<td>40</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>66</td>
<td>26</td>
<td>75</td>
<td>53</td>
<td>55</td>
<td>57</td>
<td>22</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>55</td>
<td>65</td>
<td>37</td>
<td>38</td>
<td>54</td>
<td>67</td>
<td>41</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>35</td>
<td>51</td>
<td>16</td>
<td>16</td>
<td>55</td>
<td>10</td>
<td>70</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>62</td>
<td>35</td>
<td>12</td>
<td>4</td>
<td>56</td>
<td>6</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>14</td>
<td>62</td>
<td>57</td>
<td>31</td>
<td>107</td>
<td>57</td>
<td>65</td>
<td>27</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>62</td>
<td>24</td>
<td>8</td>
<td>2</td>
<td>58</td>
<td>40</td>
<td>60</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>16</td>
<td>21</td>
<td>36</td>
<td>19</td>
<td>10</td>
<td>59</td>
<td>70</td>
<td>64</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>33</td>
<td>44</td>
<td>20</td>
<td>8</td>
<td>60</td>
<td>64</td>
<td>4</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>56</td>
<td>13</td>
<td>11</td>
<td>61</td>
<td>36</td>
<td>6</td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>19</td>
<td>62</td>
<td>48</td>
<td>15</td>
<td>6</td>
<td>62</td>
<td>30</td>
<td>20</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>20</td>
<td>66</td>
<td>14</td>
<td>22</td>
<td>30</td>
<td>63</td>
<td>20</td>
<td>30</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>44</td>
<td>13</td>
<td>28</td>
<td>87</td>
<td>64</td>
<td>15</td>
<td>5</td>
<td>28</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>26</td>
<td>13</td>
<td>12</td>
<td>9</td>
<td>65</td>
<td>50</td>
<td>70</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>11</td>
<td>28</td>
<td>6</td>
<td>2</td>
<td>66</td>
<td>57</td>
<td>72</td>
<td>37</td>
<td>152</td>
</tr>
<tr>
<td>24</td>
<td>7</td>
<td>43</td>
<td>27</td>
<td>81</td>
<td>67</td>
<td>45</td>
<td>42</td>
<td>30</td>
<td>36</td>
</tr>
<tr>
<td>25</td>
<td>17</td>
<td>64</td>
<td>14</td>
<td>22</td>
<td>68</td>
<td>38</td>
<td>33</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>41</td>
<td>46</td>
<td>18</td>
<td>36</td>
<td>69</td>
<td>50</td>
<td>4</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>27</td>
<td>55</td>
<td>34</td>
<td>17</td>
<td>6</td>
<td>70</td>
<td>66</td>
<td>8</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>28</td>
<td>35</td>
<td>16</td>
<td>29</td>
<td>23</td>
<td>71</td>
<td>59</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>52</td>
<td>26</td>
<td>13</td>
<td>7</td>
<td>72</td>
<td>35</td>
<td>60</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>43</td>
<td>26</td>
<td>22</td>
<td>19</td>
<td>73</td>
<td>27</td>
<td>24</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>76</td>
<td>25</td>
<td>17</td>
<td>74</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>32</td>
<td>22</td>
<td>53</td>
<td>28</td>
<td>22</td>
<td>75</td>
<td>40</td>
<td>37</td>
<td>20</td>
<td>44</td>
</tr>
<tr>
<td>33</td>
<td>26</td>
<td>29</td>
<td>27</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>50</td>
<td>40</td>
<td>19</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>55</td>
<td>50</td>
<td>10</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>54</td>
<td>10</td>
<td>12</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>60</td>
<td>15</td>
<td>14</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>66</td>
<td>24</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>30</td>
<td>60</td>
<td>16</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>30</td>
<td>50</td>
<td>33</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>12</td>
<td>17</td>
<td>15</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>15</td>
<td>14</td>
<td>11</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>16</td>
<td>19</td>
<td>18</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Central Depot is at $x_o = 40, y_o = 40$

Vehicle Capacity is $Q = 140$
### TABLE XXVI

#### 50 NODE PROBLEM

**TEST PROBLEM #3**

<table>
<thead>
<tr>
<th>i</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$\mu_i$</th>
<th>$\sigma^2_i$</th>
<th>Mean Unload Time*</th>
<th>i</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$\mu_i$</th>
<th>$\sigma^2_i$</th>
<th>Mean Unload Time*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>52</td>
<td>7</td>
<td>3</td>
<td>7</td>
<td>26</td>
<td>27</td>
<td>68</td>
<td>7</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>49</td>
<td>30</td>
<td>100</td>
<td>30</td>
<td>27</td>
<td>30</td>
<td>48</td>
<td>25</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>52</td>
<td>64</td>
<td>16</td>
<td>5</td>
<td>16</td>
<td>28</td>
<td>43</td>
<td>67</td>
<td>14</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>26</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>29</td>
<td>58</td>
<td>48</td>
<td>6</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>30</td>
<td>21</td>
<td>18</td>
<td>21</td>
<td>30</td>
<td>58</td>
<td>27</td>
<td>19</td>
<td>40</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>47</td>
<td>15</td>
<td>6</td>
<td>15</td>
<td>31</td>
<td>37</td>
<td>69</td>
<td>11</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>63</td>
<td>19</td>
<td>14</td>
<td>19</td>
<td>32</td>
<td>38</td>
<td>46</td>
<td>12</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td>62</td>
<td>23</td>
<td>15</td>
<td>23</td>
<td>33</td>
<td>46</td>
<td>10</td>
<td>23</td>
<td>11</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>52</td>
<td>33</td>
<td>11</td>
<td>3</td>
<td>11</td>
<td>34</td>
<td>61</td>
<td>33</td>
<td>26</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td>51</td>
<td>21</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>35</td>
<td>62</td>
<td>63</td>
<td>17</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>42</td>
<td>41</td>
<td>19</td>
<td>14</td>
<td>19</td>
<td>36</td>
<td>63</td>
<td>69</td>
<td>6</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>31</td>
<td>32</td>
<td>20</td>
<td>30</td>
<td>32</td>
<td>37</td>
<td>32</td>
<td>22</td>
<td>9</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>25</td>
<td>23</td>
<td>33</td>
<td>33</td>
<td>38</td>
<td>45</td>
<td>35</td>
<td>15</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>42</td>
<td>21</td>
<td>49</td>
<td>21</td>
<td>39</td>
<td>59</td>
<td>15</td>
<td>14</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>36</td>
<td>16</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>40</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
<td>41</td>
<td>15</td>
<td>9</td>
<td>15</td>
<td>41</td>
<td>10</td>
<td>17</td>
<td>27</td>
<td>81</td>
<td>27</td>
</tr>
<tr>
<td>17</td>
<td>27</td>
<td>23</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>42</td>
<td>21</td>
<td>10</td>
<td>13</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>33</td>
<td>41</td>
<td>67</td>
<td>41</td>
<td>43</td>
<td>5</td>
<td>64</td>
<td>11</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>13</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>44</td>
<td>30</td>
<td>15</td>
<td>16</td>
<td>28</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>57</td>
<td>58</td>
<td>28</td>
<td>87</td>
<td>28</td>
<td>45</td>
<td>39</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>21</td>
<td>62</td>
<td>42</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>46</td>
<td>32</td>
<td>39</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>42</td>
<td>57</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>47</td>
<td>25</td>
<td>32</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>57</td>
<td>16</td>
<td>7</td>
<td>16</td>
<td>48</td>
<td>25</td>
<td>55</td>
<td>17</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>52</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>49</td>
<td>48</td>
<td>28</td>
<td>18</td>
<td>13</td>
<td>18</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>38</td>
<td>28</td>
<td>16</td>
<td>28</td>
<td>50</td>
<td>56</td>
<td>37</td>
<td>10</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

Central Depot is at $x_0 = 30$, $y_0 = 40$

Vehicle Capacity is $Q = 160$

*Unload time is poisson distributed (mean = variance)
TABLE XXVII
50 NODE PROBLEM
TEST PROBLEM #3
MEAN TRAVEL TIME

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>19</th>
<th>28</th>
<th>42</th>
<th>23</th>
<th>19</th>
<th>16</th>
<th>34</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>17</td>
<td>12</td>
<td>37</td>
<td>24</td>
<td>32</td>
<td>29</td>
<td>23</td>
<td>20</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>41</td>
<td>27</td>
<td>29</td>
<td>33</td>
<td>30</td>
<td>36</td>
<td>12</td>
<td>38</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>38</td>
<td>14</td>
<td>43</td>
<td>41</td>
<td>50</td>
<td>55</td>
<td>24</td>
<td>21</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>19</td>
<td>40</td>
<td>44</td>
<td>32</td>
<td>40</td>
<td>5</td>
<td>14</td>
<td>21</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>17</td>
<td>26</td>
<td>40</td>
<td>29</td>
<td>23</td>
<td>30</td>
<td>16</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>17</td>
<td>28</td>
<td>53</td>
<td>35</td>
<td>46</td>
<td>25</td>
<td>39</td>
<td>36</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>35</td>
<td>11</td>
<td>28</td>
<td>37</td>
<td>42</td>
<td>25</td>
<td>12</td>
<td>22</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>23</td>
<td>10</td>
<td>54</td>
<td>39</td>
<td>35</td>
<td>40</td>
<td>39</td>
<td>25</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>56</td>
<td>56</td>
<td>44</td>
<td>40</td>
<td>53</td>
<td>19</td>
<td>30</td>
<td>17</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>17</td>
<td>0</td>
<td>21</td>
<td>47</td>
<td>28</td>
<td>36</td>
<td>44</td>
<td>29</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>15</td>
<td>32</td>
<td>63</td>
<td>48</td>
<td>45</td>
<td>13</td>
<td>43</td>
<td>45</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>15</td>
<td>43</td>
<td>52</td>
<td>55</td>
<td>37</td>
<td>25</td>
<td>25</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>30</td>
<td>16</td>
<td>49</td>
<td>26</td>
<td>25</td>
<td>32</td>
<td>41</td>
<td>20</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>63</td>
<td>60</td>
<td>58</td>
<td>49</td>
<td>51</td>
<td>26</td>
<td>38</td>
<td>32</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>26</td>
<td>21</td>
<td>0</td>
<td>62</td>
<td>46</td>
<td>45</td>
<td>45</td>
<td>28</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>33</td>
<td>48</td>
<td>76</td>
<td>57</td>
<td>63</td>
<td>30</td>
<td>60</td>
<td>59</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>31</td>
<td>17</td>
<td>47</td>
<td>57</td>
<td>65</td>
<td>33</td>
<td>35</td>
<td>14</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>21</td>
<td>30</td>
<td>68</td>
<td>41</td>
<td>15</td>
<td>17</td>
<td>58</td>
<td>38</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>78</td>
<td>77</td>
<td>59</td>
<td>67</td>
<td>69</td>
<td>41</td>
<td>53</td>
<td>37</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>40</td>
<td>47</td>
<td>62</td>
<td>0</td>
<td>27</td>
<td>28</td>
<td>47</td>
<td>48</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>34</td>
<td>18</td>
<td>21</td>
<td>24</td>
<td>25</td>
<td>45</td>
<td>12</td>
<td>12</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>56</td>
<td>48</td>
<td>40</td>
<td>37</td>
<td>24</td>
<td>54</td>
<td>31</td>
<td>59</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>58</td>
<td>35</td>
<td>39</td>
<td>52</td>
<td>70</td>
<td>75</td>
<td>18</td>
<td>34</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>19</td>
<td>22</td>
<td>52</td>
<td>20</td>
<td>32</td>
<td>24</td>
<td>12</td>
<td>38</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>29</td>
<td>28</td>
<td>33</td>
<td>0</td>
<td>33</td>
<td>51</td>
<td>42</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>14</td>
<td>45</td>
<td>39</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>30</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>33</td>
<td>35</td>
<td>46</td>
<td>49</td>
<td>43</td>
<td>51</td>
<td>27</td>
<td>47</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>49</td>
<td>22</td>
<td>28</td>
<td>50</td>
<td>57</td>
<td>16</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>42</td>
<td>36</td>
<td>61</td>
<td>24</td>
<td>27</td>
<td>17</td>
<td>21</td>
<td>37</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>23</td>
<td>36</td>
<td>45</td>
<td>28</td>
<td>33</td>
<td>0</td>
<td>22</td>
<td>24</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>29</td>
<td>24</td>
<td>35</td>
<td>15</td>
<td>44</td>
<td>40</td>
<td>40</td>
<td>32</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>52</td>
<td>30</td>
<td>16</td>
<td>19</td>
<td>22</td>
<td>29</td>
<td>13</td>
<td>38</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>35</td>
<td>23</td>
<td>56</td>
<td>53</td>
<td>55</td>
<td>59</td>
<td>35</td>
<td>35</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>41</td>
<td>47</td>
<td>31</td>
<td>42</td>
<td>52</td>
<td>19</td>
<td>21</td>
<td>13</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>30</td>
<td>44</td>
<td>45</td>
<td>47</td>
<td>51</td>
<td>22</td>
<td>0</td>
<td>19</td>
<td>58</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>67</td>
<td>42</td>
<td>43</td>
<td>50</td>
<td>28</td>
<td>63</td>
<td>52</td>
<td>52</td>
<td>38</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>62</td>
<td>33</td>
<td>10</td>
<td>20</td>
<td>35</td>
<td>16</td>
<td>26</td>
<td>34</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>28</td>
<td>35</td>
<td>75</td>
<td>66</td>
<td>56</td>
<td>58</td>
<td>58</td>
<td>55</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>58</td>
<td>66</td>
<td>17</td>
<td>62</td>
<td>71</td>
<td>36</td>
<td>41</td>
<td>16</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>16</td>
<td>29</td>
<td>28</td>
<td>48</td>
<td>42</td>
<td>24</td>
<td>19</td>
<td>0</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>31</td>
<td>38</td>
<td>57</td>
<td>36</td>
<td>58</td>
<td>38</td>
<td>50</td>
<td>41</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>47</td>
<td>17</td>
<td>21</td>
<td>33</td>
<td>43</td>
<td>11</td>
<td>19</td>
<td>18</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>14</td>
<td>23</td>
<td>67</td>
<td>53</td>
<td>40</td>
<td>42</td>
<td>51</td>
<td>39</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>62</td>
<td>66</td>
<td>34</td>
<td>59</td>
<td>66</td>
<td>30</td>
<td>39</td>
<td>14</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>32</td>
<td>22</td>
<td>40</td>
<td>42</td>
<td>17</td>
<td>43</td>
<td>58</td>
<td>45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>28</td>
<td>60</td>
<td>52</td>
<td>31</td>
<td>12</td>
<td>35</td>
<td>44</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>19</td>
<td>34</td>
<td>54</td>
<td>60</td>
<td>57</td>
<td>54</td>
<td>34</td>
<td>45</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>49</td>
<td>25</td>
<td>31</td>
<td>13</td>
<td>40</td>
<td>48</td>
<td>30</td>
<td>11</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>56</td>
<td>49</td>
<td>70</td>
<td>37</td>
<td>34</td>
<td>28</td>
<td>35</td>
<td>44</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>36</td>
<td>54</td>
<td>40</td>
<td>20</td>
<td>50</td>
<td>67</td>
<td>57</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>29</td>
<td>30</td>
<td>58</td>
<td>56</td>
<td>21</td>
<td>27</td>
<td>31</td>
<td>46</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>31</td>
<td>47</td>
<td>63</td>
<td>66</td>
<td>59</td>
<td>66</td>
<td>44</td>
<td>59</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>62</td>
<td>36</td>
<td>17</td>
<td>21</td>
<td>55</td>
<td>62</td>
<td>25</td>
<td>21</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>52</td>
<td>41</td>
<td>78</td>
<td>29</td>
<td>22</td>
<td>34</td>
<td>36</td>
<td>54</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>15</td>
<td>33</td>
<td>34</td>
<td>16</td>
<td>29</td>
<td>42</td>
<td>31</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>20</td>
<td>51</td>
<td>39</td>
<td>33</td>
<td>14</td>
<td>31</td>
<td>34</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>27</td>
<td>22</td>
<td>39</td>
<td>45</td>
<td>45</td>
<td>40</td>
<td>19</td>
<td>34</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>37</td>
<td>10</td>
<td>40</td>
<td>27</td>
<td>38</td>
<td>44</td>
<td>28</td>
<td>11</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>50</td>
<td>47</td>
<td>55</td>
<td>37</td>
<td>40</td>
<td>15</td>
<td>26</td>
<td>29</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>28</td>
<td>32</td>
<td>48</td>
<td>18</td>
<td>14</td>
<td>24</td>
<td>43</td>
<td>38</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>0</td>
<td>35</td>
<td>28</td>
<td>23</td>
<td>30</td>
<td>14</td>
<td>19</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>35</td>
<td>37</td>
<td>39</td>
<td>32</td>
<td>46</td>
<td>22</td>
<td>47</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>47</td>
<td>21</td>
<td>34</td>
<td>39</td>
<td>55</td>
<td>61</td>
<td>15</td>
<td>20</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>33</td>
<td>31</td>
<td>52</td>
<td>23</td>
<td>31</td>
<td>11</td>
<td>10</td>
<td>31</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>53</td>
<td>63</td>
<td>76</td>
<td>21</td>
<td>45</td>
<td>35</td>
<td>50</td>
<td>57</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>51</td>
<td>35</td>
<td>0</td>
<td>25</td>
<td>41</td>
<td>62</td>
<td>29</td>
<td>20</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>74</td>
<td>61</td>
<td>43</td>
<td>35</td>
<td>18</td>
<td>60</td>
<td>43</td>
<td>70</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>68</td>
<td>49</td>
<td>55</td>
<td>70</td>
<td>85</td>
<td>90</td>
<td>35</td>
<td>52</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>14</td>
<td>29</td>
<td>49</td>
<td>35</td>
<td>47</td>
<td>39</td>
<td>28</td>
<td>46</td>
<td>54</td>
<td></td>
</tr>
</tbody>
</table>
TABLE XXVII (continued)

<p>| | | | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>35</td>
<td>48</td>
<td>57</td>
<td>24</td>
<td>39</td>
<td>15</td>
<td>28</td>
<td>36</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>39</td>
<td>28</td>
<td>25</td>
<td>0</td>
<td>45</td>
<td>51</td>
<td>32</td>
<td>15</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>62</td>
<td>43</td>
<td>21</td>
<td>15</td>
<td>10</td>
<td>39</td>
<td>25</td>
<td>50</td>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>47</td>
<td>34</td>
<td>59</td>
<td>62</td>
<td>68</td>
<td>72</td>
<td>36</td>
<td>43</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>33</td>
<td>42</td>
<td>30</td>
<td>41</td>
<td>53</td>
<td>27</td>
<td>22</td>
<td>25</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>46</td>
<td>45</td>
<td>63</td>
<td>25</td>
<td>20</td>
<td>44</td>
<td>63</td>
<td>58</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>33</td>
<td>23</td>
<td>41</td>
<td>45</td>
<td>0</td>
<td>38</td>
<td>16</td>
<td>33</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>47</td>
<td>52</td>
<td>57</td>
<td>57</td>
<td>46</td>
<td>66</td>
<td>42</td>
<td>64</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>66</td>
<td>39</td>
<td>16</td>
<td>39</td>
<td>67</td>
<td>74</td>
<td>11</td>
<td>28</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>34</td>
<td>22</td>
<td>71</td>
<td>10</td>
<td>11</td>
<td>31</td>
<td>26</td>
<td>51</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>25</td>
<td>13</td>
<td>30</td>
<td>45</td>
<td>22</td>
<td>40</td>
<td>52</td>
<td>38</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>14</td>
<td>30</td>
<td>62</td>
<td>51</td>
<td>38</td>
<td>0</td>
<td>39</td>
<td>46</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>15</td>
<td>25</td>
<td>50</td>
<td>57</td>
<td>57</td>
<td>47</td>
<td>30</td>
<td>36</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>41</td>
<td>20</td>
<td>40</td>
<td>17</td>
<td>31</td>
<td>39</td>
<td>36</td>
<td>14</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>61</td>
<td>55</td>
<td>65</td>
<td>43</td>
<td>43</td>
<td>27</td>
<td>37</td>
<td>39</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>23</td>
<td>39</td>
<td>43</td>
<td>60</td>
<td>12</td>
<td>20</td>
<td>32</td>
<td>52</td>
<td>50</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>14</td>
<td>29</td>
<td>32</td>
<td>16</td>
<td>39</td>
<td>0</td>
<td>19</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>50</td>
<td>47</td>
<td>45</td>
<td>44</td>
<td>32</td>
<td>56</td>
<td>33</td>
<td>59</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>59</td>
<td>33</td>
<td>30</td>
<td>45</td>
<td>66</td>
<td>73</td>
<td>9</td>
<td>28</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>20</td>
<td>58</td>
<td>13</td>
<td>24</td>
<td>23</td>
<td>14</td>
<td>41</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>36</td>
<td>45</td>
<td>59</td>
<td>12</td>
<td>30</td>
<td>20</td>
<td>38</td>
<td>41</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>34</td>
<td>19</td>
<td>20</td>
<td>15</td>
<td>33</td>
<td>46</td>
<td>19</td>
<td>0</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>58</td>
<td>44</td>
<td>31</td>
<td>28</td>
<td>16</td>
<td>46</td>
<td>26</td>
<td>54</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>52</td>
<td>32</td>
<td>47</td>
<td>55</td>
<td>67</td>
<td>73</td>
<td>25</td>
<td>36</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>23</td>
<td>31</td>
<td>42</td>
<td>29</td>
<td>41</td>
<td>22</td>
<td>12</td>
<td>31</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>57</td>
<td>64</td>
<td>80</td>
<td>20</td>
<td>41</td>
<td>44</td>
<td>63</td>
<td>65</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>51</td>
<td>34</td>
<td>20</td>
<td>37</td>
<td>30</td>
<td>60</td>
<td>23</td>
<td>27</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>71</td>
<td>66</td>
<td>55</td>
<td>50</td>
<td>33</td>
<td>71</td>
<td>49</td>
<td>77</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>76</td>
<td>52</td>
<td>42</td>
<td>65</td>
<td>87</td>
<td>93</td>
<td>28</td>
<td>49</td>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>13</td>
<td>64</td>
<td>23</td>
<td>34</td>
<td>41</td>
<td>29</td>
<td>55</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>28</td>
<td>17</td>
<td>12</td>
<td>61</td>
<td>42</td>
<td>48</td>
<td>51</td>
<td>34</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>30</td>
<td>47</td>
<td>76</td>
<td>60</td>
<td>59</td>
<td>24</td>
<td>58</td>
<td>59</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>21</td>
<td>52</td>
<td>62</td>
<td>67</td>
<td>40</td>
<td>37</td>
<td>22</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>30</td>
<td>29</td>
<td>62</td>
<td>33</td>
<td>11</td>
<td>18</td>
<td>55</td>
<td>34</td>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>77</td>
<td>74</td>
<td>65</td>
<td>63</td>
<td>64</td>
<td>40</td>
<td>52</td>
<td>41</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>20</td>
<td>31</td>
<td>56</td>
<td>33</td>
<td>52</td>
<td>62</td>
<td>47</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>27</td>
<td>42</td>
<td>74</td>
<td>62</td>
<td>47</td>
<td>15</td>
<td>50</td>
<td>58</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>32</td>
<td>61</td>
<td>68</td>
<td>69</td>
<td>55</td>
<td>42</td>
<td>40</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>47</td>
<td>32</td>
<td>45</td>
<td>13</td>
<td>28</td>
<td>35</td>
<td>46</td>
<td>25</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>72</td>
<td>65</td>
<td>76</td>
<td>53</td>
<td>50</td>
<td>39</td>
<td>48</td>
<td>50</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>11</td>
<td>15</td>
<td>17</td>
<td>48</td>
<td>35</td>
<td>30</td>
<td>33</td>
<td>17</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>22</td>
<td>35</td>
<td>61</td>
<td>43</td>
<td>52</td>
<td>25</td>
<td>47</td>
<td>44</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>32</td>
<td>0</td>
<td>34</td>
<td>44</td>
<td>50</td>
<td>25</td>
<td>20</td>
<td>15</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>18</td>
<td>17</td>
<td>59</td>
<td>39</td>
<td>28</td>
<td>32</td>
<td>46</td>
<td>29</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>64</td>
<td>64</td>
<td>48</td>
<td>55</td>
<td>59</td>
<td>27</td>
<td>39</td>
<td>23</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>28</td>
<td>43</td>
<td>47</td>
<td>40</td>
<td>46</td>
<td>16</td>
<td>10</td>
<td>21</td>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>39</td>
<td>37</td>
<td>43</td>
<td>21</td>
<td>57</td>
<td>50</td>
<td>45</td>
<td>31</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>61</td>
<td>34</td>
<td>0</td>
<td>14</td>
<td>28</td>
<td>21</td>
<td>22</td>
<td>37</td>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>32</td>
<td>69</td>
<td>64</td>
<td>58</td>
<td>61</td>
<td>49</td>
<td>46</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>51</td>
<td>59</td>
<td>18</td>
<td>56</td>
<td>65</td>
<td>31</td>
<td>34</td>
<td>14</td>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>37</td>
<td>52</td>
<td>57</td>
<td>37</td>
<td>49</td>
<td>19</td>
<td>20</td>
<td>33</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>45</td>
<td>39</td>
<td>35</td>
<td>15</td>
<td>57</td>
<td>57</td>
<td>44</td>
<td>28</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>68</td>
<td>44</td>
<td>14</td>
<td>0</td>
<td>19</td>
<td>32</td>
<td>29</td>
<td>48</td>
<td>63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>43</td>
<td>39</td>
<td>70</td>
<td>70</td>
<td>69</td>
<td>72</td>
<td>49</td>
<td>51</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>45</td>
<td>55</td>
<td>17</td>
<td>54</td>
<td>65</td>
<td>35</td>
<td>34</td>
<td>23</td>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>42</td>
<td>55</td>
<td>65</td>
<td>24</td>
<td>43</td>
<td>22</td>
<td>35</td>
<td>43</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>45</td>
<td>32</td>
<td>18</td>
<td>10</td>
<td>46</td>
<td>57</td>
<td>32</td>
<td>16</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>69</td>
<td>50</td>
<td>28</td>
<td>19</td>
<td>0</td>
<td>46</td>
<td>33</td>
<td>58</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>54</td>
<td>41</td>
<td>60</td>
<td>68</td>
<td>75</td>
<td>79</td>
<td>38</td>
<td>48</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>28</td>
<td>40</td>
<td>34</td>
<td>42</td>
<td>54</td>
<td>33</td>
<td>25</td>
<td>32</td>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>25</td>
<td>37</td>
<td>33</td>
<td>54</td>
<td>51</td>
<td>29</td>
<td>16</td>
<td>11</td>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>40</td>
<td>46</td>
<td>60</td>
<td>39</td>
<td>66</td>
<td>47</td>
<td>56</td>
<td>46</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>55</td>
<td>25</td>
<td>21</td>
<td>32</td>
<td>46</td>
<td>0</td>
<td>27</td>
<td>22</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>15</td>
<td>32</td>
<td>76</td>
<td>61</td>
<td>45</td>
<td>46</td>
<td>58</td>
<td>48</td>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>67</td>
<td>72</td>
<td>29</td>
<td>66</td>
<td>74</td>
<td>38</td>
<td>46</td>
<td>18</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>25</td>
<td>35</td>
<td>31</td>
<td>27</td>
<td>13</td>
<td>26</td>
<td>19</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>19</td>
<td>22</td>
<td>43</td>
<td>25</td>
<td>42</td>
<td>30</td>
<td>33</td>
<td>26</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>42</td>
<td>20</td>
<td>22</td>
<td>29</td>
<td>33</td>
<td>27</td>
<td>0</td>
<td>30</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>29</td>
<td>12</td>
<td>52</td>
<td>44</td>
<td>45</td>
<td>49</td>
<td>34</td>
<td>26</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>47</td>
<td>39</td>
<td>38</td>
<td>42</td>
<td>49</td>
<td>14</td>
<td>23</td>
<td>13</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE XXVII (continued)

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>22</td>
<td>25</td>
<td>14</td>
<td>59</td>
<td>47</td>
<td>38</td>
<td>34</td>
<td>18</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>34</td>
<td>47</td>
<td>70</td>
<td>50</td>
<td>64</td>
<td>36</td>
<td>59</td>
<td>54</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>40</td>
<td>15</td>
<td>37</td>
<td>48</td>
<td>58</td>
<td>22</td>
<td>30</td>
<td>0</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>10</td>
<td>28</td>
<td>71</td>
<td>49</td>
<td>26</td>
<td>27</td>
<td>58</td>
<td>41</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>74</td>
<td>76</td>
<td>48</td>
<td>67</td>
<td>71</td>
<td>39</td>
<td>50</td>
<td>28</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>28</td>
<td>13</td>
<td>23</td>
<td>55</td>
<td>.33</td>
<td>47</td>
<td>55</td>
<td>39</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>23</td>
<td>40</td>
<td>72</td>
<td>58</td>
<td>49</td>
<td>14</td>
<td>50</td>
<td>55</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>25</td>
<td>54</td>
<td>63</td>
<td>65</td>
<td>47</td>
<td>36</td>
<td>32</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>38</td>
<td>27</td>
<td>50</td>
<td>21</td>
<td>21</td>
<td>28</td>
<td>47</td>
<td>25</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>71</td>
<td>66</td>
<td>69</td>
<td>54</td>
<td>53</td>
<td>36</td>
<td>47</td>
<td>43</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>42</td>
<td>31</td>
<td>47</td>
<td>48</td>
<td>24</td>
<td>53</td>
<td>68</td>
<td>56</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>28</td>
<td>35</td>
<td>66</td>
<td>61</td>
<td>32</td>
<td>21</td>
<td>40</td>
<td>52</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>21</td>
<td>43</td>
<td>64</td>
<td>70</td>
<td>65</td>
<td>64</td>
<td>44</td>
<td>54</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>59</td>
<td>36</td>
<td>27</td>
<td>11</td>
<td>46</td>
<td>53</td>
<td>34</td>
<td>21</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>61</td>
<td>51</td>
<td>80</td>
<td>39</td>
<td>33</td>
<td>37</td>
<td>43</td>
<td>54</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>23</td>
<td>30</td>
<td>21</td>
<td>58</td>
<td>49</td>
<td>35</td>
<td>28</td>
<td>14</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>37</td>
<td>47</td>
<td>68</td>
<td>47</td>
<td>66</td>
<td>41</td>
<td>59</td>
<td>52</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>47</td>
<td>18</td>
<td>32</td>
<td>43</td>
<td>54</td>
<td>15</td>
<td>29</td>
<td>10</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>0</td>
<td>30</td>
<td>74</td>
<td>54</td>
<td>33</td>
<td>34</td>
<td>59</td>
<td>44</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>73</td>
<td>76</td>
<td>41</td>
<td>68</td>
<td>73</td>
<td>39</td>
<td>49</td>
<td>25</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>16</td>
<td>30</td>
<td>35</td>
<td>22</td>
<td>23</td>
<td>35</td>
<td>23</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>10</td>
<td>21</td>
<td>49</td>
<td>34</td>
<td>39</td>
<td>20</td>
<td>33</td>
<td>32</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>32</td>
<td>17</td>
<td>32</td>
<td>39</td>
<td>41</td>
<td>32</td>
<td>12</td>
<td>28</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>30</td>
<td>0</td>
<td>47</td>
<td>34</td>
<td>38</td>
<td>43</td>
<td>32</td>
<td>18</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>51</td>
<td>50</td>
<td>48</td>
<td>41</td>
<td>46</td>
<td>14</td>
<td>25</td>
<td>21</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>54</td>
<td>49</td>
<td>68</td>
<td>39</td>
<td>28</td>
<td>56</td>
<td>75</td>
<td>67</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>40</td>
<td>34</td>
<td>55</td>
<td>59</td>
<td>16</td>
<td>40</td>
<td>30</td>
<td>47</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>45</td>
<td>59</td>
<td>69</td>
<td>70</td>
<td>60</td>
<td>76</td>
<td>52</td>
<td>71</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>74</td>
<td>47</td>
<td>0</td>
<td>35</td>
<td>69</td>
<td>76</td>
<td>25</td>
<td>33</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>47</td>
<td>32</td>
<td>84</td>
<td>23</td>
<td>11</td>
<td>41</td>
<td>39</td>
<td>62</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>39</td>
<td>26</td>
<td>41</td>
<td>52</td>
<td>28</td>
<td>53</td>
<td>66</td>
<td>53</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>27</td>
<td>39</td>
<td>70</td>
<td>62</td>
<td>39</td>
<td>17</td>
<td>45</td>
<td>55</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>13</td>
<td>39</td>
<td>64</td>
<td>70</td>
<td>68</td>
<td>61</td>
<td>44</td>
<td>49</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>54</td>
<td>34</td>
<td>35</td>
<td>0</td>
<td>39</td>
<td>46</td>
<td>40</td>
<td>22</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>67</td>
<td>58</td>
<td>79</td>
<td>46</td>
<td>41</td>
<td>38</td>
<td>46</td>
<td>53</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**TABLE XXVII (continued)**

<table>
<thead>
<tr>
<th></th>
<th>35</th>
<th>25</th>
<th>15</th>
<th>70</th>
<th>50</th>
<th>55</th>
<th>56</th>
<th>40</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>35</td>
<td>25</td>
<td>15</td>
<td>70</td>
<td>50</td>
<td>55</td>
<td>56</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>55</td>
<td>38</td>
<td>55</td>
<td>85</td>
<td>68</td>
<td>67</td>
<td>31</td>
<td>66</td>
<td>67</td>
<td>87</td>
</tr>
<tr>
<td>11</td>
<td>28</td>
<td>28</td>
<td>58</td>
<td>69</td>
<td>75</td>
<td>45</td>
<td>45</td>
<td>26</td>
<td>21</td>
</tr>
<tr>
<td>46</td>
<td>33</td>
<td>38</td>
<td>69</td>
<td>39</td>
<td>0</td>
<td>10</td>
<td>63</td>
<td>42</td>
<td>60</td>
</tr>
<tr>
<td>99</td>
<td>86</td>
<td>83</td>
<td>71</td>
<td>72</td>
<td>72</td>
<td>49</td>
<td>60</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>35</td>
<td>40</td>
<td>32</td>
<td>17</td>
<td>75</td>
<td>57</td>
<td>59</td>
<td>58</td>
<td>42</td>
<td>48</td>
</tr>
<tr>
<td>55</td>
<td>44</td>
<td>61</td>
<td>90</td>
<td>72</td>
<td>74</td>
<td>39</td>
<td>73</td>
<td>73</td>
<td>93</td>
</tr>
<tr>
<td>18</td>
<td>35</td>
<td>32</td>
<td>61</td>
<td>72</td>
<td>79</td>
<td>46</td>
<td>49</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>53</td>
<td>34</td>
<td>43</td>
<td>76</td>
<td>46</td>
<td>10</td>
<td>0</td>
<td>70</td>
<td>49</td>
<td>67</td>
</tr>
<tr>
<td>05</td>
<td>92</td>
<td>89</td>
<td>72</td>
<td>78</td>
<td>79</td>
<td>54</td>
<td>66</td>
<td>51</td>
<td>55</td>
</tr>
<tr>
<td>42</td>
<td>24</td>
<td>39</td>
<td>41</td>
<td>58</td>
<td>18</td>
<td>16</td>
<td>35</td>
<td>55</td>
<td>51</td>
</tr>
<tr>
<td>25</td>
<td>28</td>
<td>15</td>
<td>35</td>
<td>36</td>
<td>11</td>
<td>36</td>
<td>9</td>
<td>25</td>
<td>28</td>
</tr>
<tr>
<td>55</td>
<td>46</td>
<td>46</td>
<td>49</td>
<td>49</td>
<td>38</td>
<td>58</td>
<td>34</td>
<td>58</td>
<td>47</td>
</tr>
<tr>
<td>34</td>
<td>59</td>
<td>32</td>
<td>25</td>
<td>40</td>
<td>63</td>
<td>70</td>
<td>0</td>
<td>25</td>
<td>36</td>
</tr>
<tr>
<td>40</td>
<td>30</td>
<td>22</td>
<td>62</td>
<td>11</td>
<td>19</td>
<td>23</td>
<td>17</td>
<td>43</td>
<td>23</td>
</tr>
<tr>
<td>36</td>
<td>21</td>
<td>20</td>
<td>38</td>
<td>34</td>
<td>11</td>
<td>35</td>
<td>50</td>
<td>39</td>
<td>11</td>
</tr>
<tr>
<td>21</td>
<td>11</td>
<td>20</td>
<td>52</td>
<td>43</td>
<td>28</td>
<td>14</td>
<td>28</td>
<td>36</td>
<td>49</td>
</tr>
<tr>
<td>34</td>
<td>25</td>
<td>29</td>
<td>46</td>
<td>51</td>
<td>48</td>
<td>48</td>
<td>26</td>
<td>41</td>
<td>25</td>
</tr>
<tr>
<td>21</td>
<td>44</td>
<td>18</td>
<td>33</td>
<td>22</td>
<td>42</td>
<td>49</td>
<td>25</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>62</td>
<td>50</td>
<td>44</td>
<td>62</td>
<td>32</td>
<td>33</td>
<td>19</td>
<td>27</td>
<td>36</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>48</td>
<td>54</td>
<td>45</td>
<td>62</td>
<td>51</td>
<td>32</td>
<td>62</td>
<td>79</td>
<td>68</td>
</tr>
<tr>
<td>48</td>
<td>54</td>
<td>45</td>
<td>62</td>
<td>51</td>
<td>32</td>
<td>62</td>
<td>79</td>
<td>68</td>
<td>26</td>
</tr>
<tr>
<td>14</td>
<td>40</td>
<td>42</td>
<td>68</td>
<td>68</td>
<td>30</td>
<td>35</td>
<td>42</td>
<td>57</td>
<td>58</td>
</tr>
<tr>
<td>54</td>
<td>35</td>
<td>57</td>
<td>75</td>
<td>78</td>
<td>71</td>
<td>77</td>
<td>55</td>
<td>68</td>
<td>42</td>
</tr>
<tr>
<td>17</td>
<td>72</td>
<td>47</td>
<td>19</td>
<td>24</td>
<td>60</td>
<td>67</td>
<td>36</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>68</td>
<td>61</td>
<td>48</td>
<td>90</td>
<td>37</td>
<td>27</td>
<td>46</td>
<td>48</td>
<td>65</td>
<td>23</td>
</tr>
<tr>
<td>29</td>
<td>53</td>
<td>70</td>
<td>76</td>
<td>92</td>
<td>32</td>
<td>53</td>
<td>55</td>
<td>72</td>
<td>77</td>
</tr>
<tr>
<td>53</td>
<td>70</td>
<td>76</td>
<td>92</td>
<td>32</td>
<td>53</td>
<td>55</td>
<td>72</td>
<td>77</td>
<td>68</td>
</tr>
<tr>
<td>61</td>
<td>64</td>
<td>47</td>
<td>25</td>
<td>47</td>
<td>42</td>
<td>73</td>
<td>36</td>
<td>38</td>
<td>15</td>
</tr>
<tr>
<td>91</td>
<td>83</td>
<td>78</td>
<td>65</td>
<td>58</td>
<td>41</td>
<td>81</td>
<td>61</td>
<td>89</td>
<td>84</td>
</tr>
<tr>
<td>71</td>
<td>87</td>
<td>65</td>
<td>52</td>
<td>77</td>
<td>99</td>
<td>105</td>
<td>40</td>
<td>62</td>
<td>68</td>
</tr>
<tr>
<td>0</td>
<td>17</td>
<td>22</td>
<td>72</td>
<td>34</td>
<td>44</td>
<td>54</td>
<td>42</td>
<td>66</td>
<td>60</td>
</tr>
<tr>
<td>74</td>
<td>39</td>
<td>56</td>
<td>63</td>
<td>78</td>
<td>19</td>
<td>42</td>
<td>41</td>
<td>58</td>
<td>62</td>
</tr>
<tr>
<td>39</td>
<td>56</td>
<td>63</td>
<td>78</td>
<td>19</td>
<td>42</td>
<td>41</td>
<td>58</td>
<td>62</td>
<td>56</td>
</tr>
<tr>
<td>52</td>
<td>50</td>
<td>33</td>
<td>14</td>
<td>33</td>
<td>34</td>
<td>61</td>
<td>24</td>
<td>23</td>
<td>8</td>
</tr>
<tr>
<td>77</td>
<td>72</td>
<td>64</td>
<td>51</td>
<td>45</td>
<td>28</td>
<td>67</td>
<td>47</td>
<td>74</td>
<td>71</td>
</tr>
<tr>
<td>61</td>
<td>73</td>
<td>51</td>
<td>47</td>
<td>67</td>
<td>86</td>
<td>92</td>
<td>30</td>
<td>50</td>
<td>61</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>18</td>
<td>59</td>
<td>27</td>
<td>38</td>
<td>40</td>
<td>28</td>
<td>52</td>
<td>50</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>18</td>
<td>59</td>
<td>27</td>
<td>38</td>
<td>40</td>
<td>28</td>
<td>52</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>56</td>
<td>60</td>
<td>77</td>
<td>22</td>
<td>36</td>
<td>47</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>41</td>
<td>47</td>
<td>31</td>
<td>29</td>
<td>42</td>
<td>22</td>
<td>55</td>
<td>20</td>
<td>31</td>
<td>13</td>
</tr>
<tr>
<td>74</td>
<td>65</td>
<td>64</td>
<td>59</td>
<td>55</td>
<td>40</td>
<td>72</td>
<td>49</td>
<td>76</td>
<td>66</td>
</tr>
<tr>
<td>51</td>
<td>76</td>
<td>50</td>
<td>32</td>
<td>58</td>
<td>83</td>
<td>89</td>
<td>22</td>
<td>44</td>
<td>48</td>
</tr>
<tr>
<td>22</td>
<td>18</td>
<td>0</td>
<td>70</td>
<td>15</td>
<td>24</td>
<td>40</td>
<td>29</td>
<td>57</td>
<td>41</td>
</tr>
<tr>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>44</td>
<td>58</td>
<td>59</td>
<td>52</td>
<td>61</td>
<td>31</td>
<td>17</td>
<td>34</td>
<td>70</td>
</tr>
<tr>
<td>78</td>
<td>55</td>
<td>52</td>
<td>49</td>
<td>30</td>
<td>71</td>
<td>65</td>
<td>58</td>
<td>42</td>
<td>64</td>
</tr>
<tr>
<td>65</td>
<td>76</td>
<td>48</td>
<td>18</td>
<td>17</td>
<td>34</td>
<td>29</td>
<td>38</td>
<td>48</td>
<td>69</td>
</tr>
<tr>
<td>80</td>
<td>41</td>
<td>48</td>
<td>84</td>
<td>79</td>
<td>71</td>
<td>72</td>
<td>62</td>
<td>62</td>
<td>90</td>
</tr>
<tr>
<td>72</td>
<td>59</td>
<td>70</td>
<td>0</td>
<td>69</td>
<td>79</td>
<td>47</td>
<td>48</td>
<td>29</td>
<td>70</td>
</tr>
<tr>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>48</td>
<td>49</td>
<td>67</td>
<td>20</td>
<td>24</td>
<td>42</td>
<td>62</td>
<td>59</td>
<td>37</td>
</tr>
<tr>
<td>29</td>
<td>37</td>
<td>23</td>
<td>35</td>
<td>41</td>
<td>10</td>
<td>43</td>
<td>13</td>
<td>29</td>
<td>23</td>
</tr>
<tr>
<td>63</td>
<td>53</td>
<td>55</td>
<td>56</td>
<td>54</td>
<td>42</td>
<td>66</td>
<td>42</td>
<td>67</td>
<td>54</td>
</tr>
<tr>
<td>39</td>
<td>68</td>
<td>41</td>
<td>23</td>
<td>46</td>
<td>72</td>
<td>78</td>
<td>11</td>
<td>32</td>
<td>37</td>
</tr>
<tr>
<td>34</td>
<td>27</td>
<td>15</td>
<td>69</td>
<td>0</td>
<td>15</td>
<td>31</td>
<td>24</td>
<td>51</td>
<td>29</td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>53</td>
<td>51</td>
<td>69</td>
<td>32</td>
<td>27</td>
<td>52</td>
<td>71</td>
<td>66</td>
<td>34</td>
</tr>
<tr>
<td>22</td>
<td>40</td>
<td>31</td>
<td>47</td>
<td>53</td>
<td>11</td>
<td>43</td>
<td>24</td>
<td>41</td>
<td>34</td>
</tr>
<tr>
<td>64</td>
<td>50</td>
<td>59</td>
<td>65</td>
<td>65</td>
<td>54</td>
<td>74</td>
<td>49</td>
<td>71</td>
<td>53</td>
</tr>
<tr>
<td>33</td>
<td>73</td>
<td>46</td>
<td>11</td>
<td>41</td>
<td>72</td>
<td>79</td>
<td>19</td>
<td>33</td>
<td>27</td>
</tr>
<tr>
<td>44</td>
<td>38</td>
<td>24</td>
<td>79</td>
<td>15</td>
<td>0</td>
<td>38</td>
<td>34</td>
<td>59</td>
<td>27</td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>26</td>
<td>41</td>
<td>24</td>
<td>17</td>
<td>19</td>
<td>36</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>34</td>
<td>15</td>
<td>11</td>
<td>39</td>
<td>27</td>
<td>31</td>
<td>17</td>
<td>23</td>
<td>22</td>
<td>41</td>
</tr>
<tr>
<td>40</td>
<td>39</td>
<td>27</td>
<td>31</td>
<td>35</td>
<td>33</td>
<td>38</td>
<td>14</td>
<td>39</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>39</td>
<td>14</td>
<td>41</td>
<td>38</td>
<td>49</td>
<td>54</td>
<td>23</td>
<td>19</td>
<td>46</td>
</tr>
<tr>
<td>54</td>
<td>40</td>
<td>40</td>
<td>47</td>
<td>31</td>
<td>38</td>
<td>0</td>
<td>14</td>
<td>23</td>
<td>26</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>30</td>
<td>38</td>
<td>53</td>
<td>12</td>
<td>21</td>
<td>21</td>
<td>41</td>
<td>39</td>
<td>35</td>
</tr>
<tr>
<td>36</td>
<td>26</td>
<td>10</td>
<td>28</td>
<td>22</td>
<td>26</td>
<td>37</td>
<td>14</td>
<td>12</td>
<td>29</td>
</tr>
<tr>
<td>52</td>
<td>48</td>
<td>39</td>
<td>34</td>
<td>34</td>
<td>25</td>
<td>46</td>
<td>23</td>
<td>50</td>
<td>47</td>
</tr>
<tr>
<td>43</td>
<td>49</td>
<td>25</td>
<td>39</td>
<td>46</td>
<td>60</td>
<td>66</td>
<td>17</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td>42</td>
<td>28</td>
<td>29</td>
<td>48</td>
<td>24</td>
<td>34</td>
<td>14</td>
<td>0</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>17</td>
<td>32</td>
<td>37</td>
<td>38</td>
<td>37</td>
<td>13</td>
<td>16</td>
<td>14</td>
<td>44</td>
</tr>
<tr>
<td>54</td>
<td>29</td>
<td>31</td>
<td>46</td>
<td>25</td>
<td>51</td>
<td>39</td>
<td>41</td>
<td>31</td>
<td>55</td>
</tr>
<tr>
<td>41</td>
<td>50</td>
<td>23</td>
<td>14</td>
<td>23</td>
<td>32</td>
<td>18</td>
<td>13</td>
<td>28</td>
<td>43</td>
</tr>
<tr>
<td>54</td>
<td>25</td>
<td>21</td>
<td>62</td>
<td>53</td>
<td>48</td>
<td>51</td>
<td>43</td>
<td>36</td>
<td>65</td>
</tr>
<tr>
<td>66</td>
<td>52</td>
<td>57</td>
<td>29</td>
<td>51</td>
<td>59</td>
<td>23</td>
<td>30</td>
<td>0</td>
<td>45</td>
</tr>
</tbody>
</table>

TABLE XXVII (continued)
TABLE XXVII (continued)

| 28 | 34 | 28 | 46 | 36 | 12 | 42 | 59 | 48 | 10 |
| 12 | 20 | 23 | 54 | 49 | 23 | 19 | 28 | 40 | 48 |
| 40 | 26 | 38 | 54 | 58 | 53 | 57 | 35 | 50 | 29 |
| 15 | 53 | 27 | 24 | 19 | 48 | 55 | 23 | 12 | 23 |
| 60 | 50 | 41 | 70 | 29 | 27 | 26 | 31 | 45 | 0 |
| 17 |   |   |   |   |   |   |   |   |   |
| 34 | 32 | 19 | 35 | 48 | 23 | 46 | 59 | 45 | 9 |
| 23 | 20 | 33 | 65 | 56 | 37 | 9  | 41 | 50 | 62 |
| 28 | 12 | 32 | 56 | 63 | 61 | 53 | 36 | 42 | 16 |
| 15 | 47 | 27 | 37 | 10 | 35 | 42 | 36 | 16 | 29 |
| 74 | 63 | 56 | 72 | 43 | 41 | 31 | 40 | 46 | 17 |
| 0  |   |   |   |   |   |   |   |   |   |
TABLE XXVIII
50 NODE PROBLEM
TEST PROBLEM #3
VARIANCE TRAVEL TIME

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>2</th>
<th>3</th>
<th>10</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>8</th>
<th>10</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>8</th>
<th>10</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>8</th>
<th>10</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
### TABLE XXVIII (continued)

<table>
<thead>
<tr>
<th>2</th>
<th>0</th>
<th>6</th>
<th>3</th>
<th>10</th>
<th>10</th>
<th>2</th>
<th>0</th>
<th>4</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>13</td>
<td>9</td>
<td>12</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>13</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>1</th>
<th>5</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>4</th>
<th>4</th>
<th>0</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>5</td>
<td>11</td>
<td>11</td>
<td>9</td>
<td>0</td>
<td>12</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>13</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>0</th>
<th>2</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>7</th>
<th>3</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>13</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>.8</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>1</th>
<th>7</th>
<th>13</th>
<th>5</th>
<th>4</th>
<th>2</th>
<th>7</th>
<th>12</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>12</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>9</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>2</th>
<th>2</th>
<th>6</th>
<th>8</th>
<th>3</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>0</th>
<th>9</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>3</th>
<th>11</th>
<th>9</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>3</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>13</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>10</td>
<td>19</td>
<td>6</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>9</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>11</td>
<td>15</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>15</td>
<td>17</td>
<td>0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>10</td>
<td>12</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>13</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>13</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>13</td>
<td>9</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>11</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>11</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>13</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>11</td>
<td>7</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>10</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3</td>
<td>14</td>
<td>15</td>
<td>5</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>13</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>13</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>7</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>14</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>5</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>6</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>19</td>
<td>11</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE XXVIII (continued)

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>14</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>6</td>
<td>11</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>13</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
<td>11</td>
<td>11</td>
<td>13</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>12</td>
<td>14</td>
<td>15</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>3</td>
<td>16</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>11</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>3</td>
<td>15</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>7</td>
<td>2</td>
<td>11</td>
<td>16</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>12</td>
<td>2</td>
<td>9</td>
<td>6</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>11</td>
<td>11</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>13</td>
<td>4</td>
<td>14</td>
<td>3</td>
<td>12</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>11</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>4</td>
<td>13</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>11</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>9</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>12</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>3</td>
<td>11</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>6</td>
<td>11</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
## TABLE XXVIII (continued)

<table>
<thead>
<tr>
<th></th>
<th>9</th>
<th>4</th>
<th>4</th>
<th>2</th>
<th>7</th>
<th>8</th>
<th>7</th>
<th>2</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>14</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>13</td>
<td>7</td>
<td>0</td>
<td>13</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>17</td>
<td>9</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>4</td>
<td>10</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>1</td>
<td>14</td>
<td>0</td>
<td>15</td>
<td>12</td>
<td>1</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>9</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>1</td>
<td>12</td>
<td>3</td>
<td>12</td>
<td>5</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>15</td>
<td>6</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>4</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>7</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>19</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>20</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>12</td>
<td>7</td>
<td>14</td>
<td>12</td>
<td>17</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TABLE XXVIII (continued)

<table>
<thead>
<tr>
<th>6</th>
<th>9</th>
<th>0</th>
<th>9</th>
<th>2</th>
<th>1</th>
<th>11</th>
<th>4</th>
<th>13</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>4</td>
<td>1 1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>3</td>
<td>13</td>
<td>7</td>
</tr>
</tbody>
</table>

| 2 | 8 | 13 | 5 | 7 | 12 | 4 | 1 | 3 | 12 |
| 11 | 5 | 1 | 7 | 0 | 7 | 15 | 7 | 8 | 12 |
| 13 | 17 | 9 | 3 | 3 | 4 | 5 | 11 | 16 |
| 15 | 2 | 13 | 14 | 7 | 14 | 2 | 8 | 0 |
| 1 | 12 | 16 | 0 | 7 | 6 | 10 | 11 | 3 | 16 |

| 4 | 1 | 0 | 15 | 0 | 2 | 5 | 6 | 5 | 5 |
| 5 | 1 | 2 | 2 | 10 | 1 | 1 | 4 | 0 | 0 |
| 9 | 1 | 7 | 10 | 3 | 3 | 6 | 5 | 11 | 10 |
| 6 | 3 | 1 | 7 | 0 | 1 | 0 | 2 | 8 | 3 |
| 1 | 7 | 8 | 6 | 10 | 7 | 5 | 0 | 13 | 5 | 3 |
| 14 | 3 | 7 | 1 | 2 | 1 | 7 | 5 | 1 | 7 |
| 13 | 6 | 6 | 2 | 7 | 13 | 15 | 1 | 13 | 8 |
| 3 | 1 | 3 | 6 | 1 | 0 | 1 | 3 | 12 | 5 |
| 9 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 6 | 6 | 0 |
| 11 | 0 | 0 | 1 | 8 | 5 | 6 | 5 | 5 | 2 | 5 |
| 1 | 2 | 0 | 4 | 7 | 1 | 9 | 0 | 7 | 7 |
| 13 | 1 | 0 | 4 | 3 | 5 | 12 | 4 | 3 | 4 |
| 11 | 0 | 8 | 10 | 0 | 1 | 0 | 3 | 4 | 2 |
| 4 | 0 | 1 | 8 | 8 | 0 | 0 | 2 | 3 | 7 | 7 |
| 6 | 1 | 0 | 6 | 3 | 4 | 5 | 2 | 1 | 2 |
| 11 | 9 | 3 | 3 | 11 | 2 | 3 | 3 | 0 | 4 | 2 |
| 11 | 5 | 1 | 5 | 2 | 8 | 6 | 2 | 0 | 3 | 6 |
| 12 | 1 | 6 | 7 | 3 | 4 | 3 | 9 | 7 | 6 |
| 1 | 1 | 4 | 1 | 5 | 3 | 3 | 3 | 1 | 8 |
| 17 | 6 | 4 | 0 | 7 | 3 | 1 | 3 | 8 | 8 |
| 11 | 0 | 13 | 3 | 8 | 12 | 4 | 4 | 0 | 4 |
TABLE XXVIII (continued)

<table>
<thead>
<tr>
<th>4</th>
<th>1</th>
<th>5</th>
<th>0</th>
<th>3</th>
<th>0</th>
<th>10</th>
<th>9</th>
<th>10</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
<td>16</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>13</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VITA

Yahia Zare-Mehrjerdi
Candidate for the Degree of
Doctor of Philosophy

Thesis: A GOAL PROGRAMMING MODEL OF THE STOCHASTIC VEHICLE ROUTING PROBLEM

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Yazd, Iran, March 30, 1953, the son of Mr. Mohammad Hasan Zare-Mehrjerdi and Mrs. Sedigheh Hashemi. Married to Mary Jo Fenske on January 9, 1982.

Education: Graduated from Iranshahr High School, Yazd, Iran, in May, 1971; received Bachelor of Science Degree in Mathematics from Iranian National University in May, 1976; received Bachelor of Science Degree in Civil Engineering from Texas A & I University in December, 1980; received Master of Science Degree in Operations Research from St. Mary's University in December, 1983; completed requirements for the Doctor of Philosophy Degree at Oklahoma State University in December, 1986.

Professional Experience: Teaching Associate, Department of Mathematics, Oklahoma State University, September, 1982 to May, 1983; Teaching Assistant, School of Industrial Engineering and Management, Oklahoma State University, September, 1983 to June, 1986.

Professional Organizations: Institute of Industrial Engineering, Institute of Management Sciences.