THE SOLUTION OF VEHICLE ROUTING PROBLEMS
IN A MULTIPLE OBJECTIVE ENVIRONMENT

By

YANG BYUNG PARK

Bachelor of Engineering
Han Yang University
Seoul, Korea
1978

Master of Science
Pennsylvania State University
University Park, Pennsylvania
1981

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, 1984

THE SOLUTION OF VEHICLE ROUTING PROBLEMS

IN A MULTIPLE OBJECTIVE ENVIRONMENT

Thesis Approved:

C. Patrick Koelling
Thesis Adviser

Carl B. Estes

M. P. Terrell

Philip M. Wolfe

Norman N. Durham
Dean of the Graduate College

PREFACE

This research is concerned with obtaining the most satisfactory or favorable vehicle routes of multicriteria VRPs. The specific model considered consists of three relevant objectives which are, more often than not, conflicting. These are the minimization of total travel distance of vehicles, the minimization of total deterioration of goods during transportation, and the maximization of total fulfillment of emergent services and conditional dependencies of stations.

A heuristic algorithm is developed to determine the most satisfactory vehicle routes of multicriteria VRPs where the three objectives are to be achieved. Computational experiments are performed on three test problems incorporating multiple objectives, in order to evaluate and justify the proposed algorithm. An interactive procedure is developed that implements the proposed algorithm and relies on the progressive definition of a Decision Maker's preferences along with the exploration of the criterion space, in order to reach the most favorable vehicle routes of multicriteria VRPs.

I wish to express my sincere appreciation and respect to my major adviser, Dr. C. Patrick Koelling, and the chairman of my Ph. D. committee, Dr. Philip M. Wolfe, for their guidance and support throughout this research and during my doctoral program. I also appreciate their personal concern for my career and in no way can their advice and care be adequately recognized. Dr. Koelling first brought the problem to my attention and continued to lend a great deal of insight, encouragement, and assistance.

Appreciation is also extended to my committee members Dr. Carl B. Estes, Dr. William D. Warde, and Dr. M. Palmer Terrell, Head of the School of Industrial Engineering and Management, for their interest and assistance during this work and my stay at Oklahoma State University.

I especially thank and acknowledge the faculty and Head of the School of Industrial Engineering and Management who have committed themselves to develop and maintain an outstanding academic and research standard that has provided national recognition and reputation for Oklahoma State University.

I owe a great thanks to Mrs. Teresa Tanner for her virtually faultless typing and suggestions concerning this dissertation.

I owe a special thanks to my parents, Mr. and Mrs. Dongsik Park, for encouraging and supporting me during my study in the United States.

Finally to my wife, Giebong, and my son, Siehyun, I owe my deepest thanks for the many sacrifices they had to make to allow me to pursue my dreams. Without their patience and understanding, the long hours and nights spent in the office and preparing this dissertation would have been unbearable.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
	Statement of the Problem	1
	Research Objectives	5
	Research Procedure	5
	Phase I	5
	Phase II	6
	Outline of Succeeding Chapters	6
II.	BACKGROUND OF THE RESEARCH	8
	Introduction	8
	Vehicle Routing Problem	8
	Example	13
	Literature Review of VRP Solving Techniques	13
	Optimal Seeking Algorithms	15
	Heuristic Algorithms	20
	Summary	34
III.	MULTIPLE OBJECTIVE OPTIMIZATION ANALYSIS	37
	Introduction	37
	Set of Nondominated Solutions	39
	Goal Programming	42
	Interactive Methods for Multiobjective Decision Making	46
	Summary	48
IV.	ALGORITHM FOR MULTICRITERIA VEHICLE ROUTING PROBLEMS	50
	Introduction	50
	Notation	52
	Assumptions	54
	Cluster Method	55
	Iterative Goal Programming Heuristic Procedure	63
	Initial Development of An Exact GP Model	63
	Heuristic Procedure	70
	Example Problem	76
	Clustering Stage	77
	Routing Stage	81
	Summary	86

Chapter Page

V. COMPUTATIONAL RESULTS AND ANALYSIS. 88

Introduction 88
Programming. 88
Test Problems. 89
Computational Experience 93
Summary. 99

VI. USING THE INTERACTIVE COMPUTER PROGRAM. 100

Introduction 100
Interactive Procedure. 101
Procedure on the Goal Priority Structure Model I 103
Procedure on the Goal Priority Structure Model II. 113
Procedure on the Goal Priority Structure Model III 120
Summary. 123

VII. CONCLUSIONS AND RECOMMENDATIONS 125

Conclusions. 125
Recommendations. 129

BIBLIOGRAPHY 130

APPENDICES 135

APPENDIX A - FORTRAN PROGRAM LISTING. 136

APPENDIX B - DATA INPUTS FOR THREE TEST PROBLEMS. . . . 167

LIST OF TABLES

Table Page

I. Data for the Sample Problem: Distance Matrix and Service Quantity. 14

II. Model Description of Algorithms Mentioned in Literature Review 35

III. Priority Structures of Three Alternative Models in The Research. 57

IV. Priority Structures of Three Alternative GP Models. . 70

V. Distance Matrix of Example Problem. 78

VI. Summary of the Outcomes of Example Problem for Three Models. 86

VII. List of Test Problems 94

VIII. Comparison of Algorithms With Model I Priority Structure 95

IX. Results of Test Problem 1 for Three Models. 98

X. Test Problem 1. 168

XI. Test Problem 2. 169

XII. Test Problem 3. 170

LIST OF FIGURES

Figure		Page
1.	A Layout of Stations in a VRP.	2
2.	Graphical Depiction of the Solution to the Sample Problem	15
3.	Link Replacement Scheme Leading to Potential Saving in a Route Structure.	22
4.	A Sequence of Steps for Multiobjective Analysis.	40
5.	Set of Nondominated Solutions.	41
6.	The Logic Flow Chart for an Interactive Procedure. . . .	47
7.	The Logic Flow Chart of the Cluster Method For Multicriteria VRPs	59
8.	The Logic Flow Chart of the Iterative GP Procedure For Multicriteria VRPs	73
9.	Graphical Configuration of a Depot and Stations in Example Problem.	77
10.	Logic Flow Chart of the Iterative GP Procedure With an Application of SLGP Approach for Model I	90
11.	Logic Flow Chart of the Iterative GP Procedure With an Application of SLGP Approach for Model II.	91
12.	Logic Flow Chart of the Iterative GP Procedure With an Application of SLGP Approach for Model III	92
13.	The Logic Flow Chart of the Proposed Interactive Procedure.	102

CHAPTER I

INTRODUCTION

Statement of the Problem

The Vehicle Routing Problem (VRP) is a generic name given to a whole class of problems involving the visiting of "stations" by "vehicles." The VRP is also referred to as "vehicle scheduling" [9, 17, 22, 23, 29, 38, 61, 62], "truck or vehicle dispatching" [13, 19, 24, 48, 52], or "multiple delivery" problem [3,57,60]. The VRP was originally posed by Dantzig and Ramser [19] and can be stated as follows:

> The number of stations at known locations are to be serviced exactly once by a set of vehicles with both capacity and distance restrictions, starting from a central depot and eventually returning to the depot through stations such that all stations with a known quantity of some commodity are fully serviced and that any restrictions are kept. The objective is to build up a schedule of routes minimizing a total distance traveled (time or cost), while satisfying the restrictions given.

Figure 1 shows a layout of the stations dispersed around a central depot, as an example.

Manifestations of this problem appear in many diverse sectors of the economy including the public and private sectors. In the public

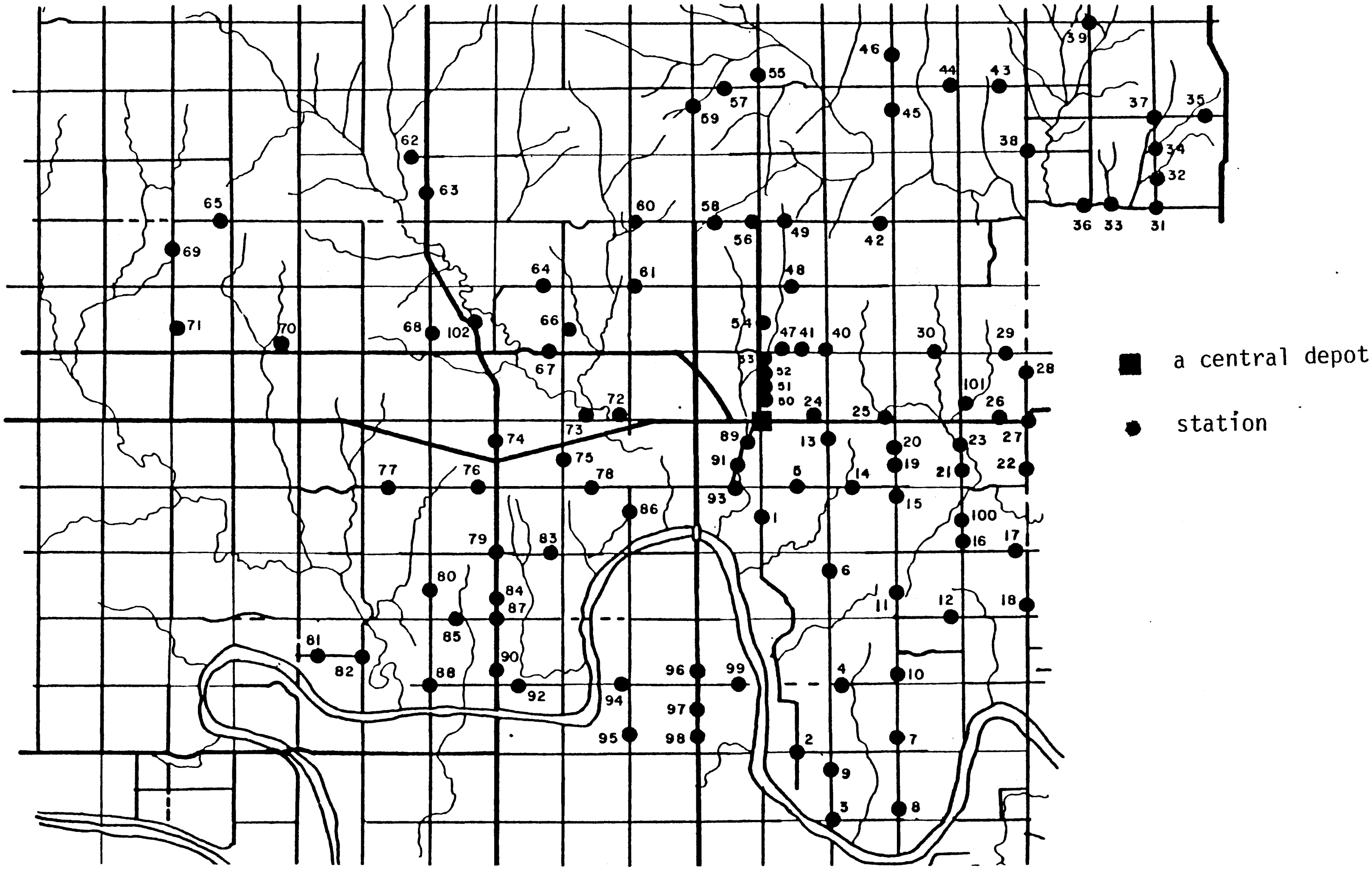

Figure 1. A Layout of Stations in a VRP

[49, p. 49]

sector, for example, analysts are constantly routing street sweepers, snow plows, mail-box collection vehicles, school buses and other service vehicles. In the private sector, for example, industries route vehicles to collect raw materials, to serve warehouses or branch stores, and to perform preventive maintenance inspection in manufacturing systems. The operation in all VRPs may be one of collection, delivery, both collection and delivery, or one involving neither. In this day and age of severe economic conditions, the VRPs become a real concern to practitioners of operations research as management becomes increasingly aware of the need to control the rising costs of the service activities by vehicles. The systematic construction of efficient vehicle route structures for operations provides an important management tool for the control of costs in the short-term, for adapting the vehicle fleet size and composition in the medium-term, and even for the location of depots in the longer-term [40].

Due to these attractive points, in recent years many researchers have been concerned not only with obtaining an optimal solution but also with developing practical and economical heuristic methods for VRPs. Each of the studies performed has a common feature of a single objective, either the minimization of cost, time, or distance traveled, while meeting the given restrictions. However, the collection or delivery problems inherent in VRPs may not lend themselves to a model construction concerning only one objective and may involve relevant multiple objectives like many other resource allocation or scheduling problems, creating multicriteria VRPs.

Deterioration of certain perishable or decaying goods, for example, vegetable, food, fish, medicine, hide, and so on, has become of major concern in the collection or delivery activity by vehicles because it may cause a significant loss of profit [1]. In some cases, there may be stations that should be serviced urgently or that are contingent upon others. Two stations are said to be contingent when there is a conditional dependency between them. A station is conditionally dependent on another when its service is operationally, functionally, or economically dependent on the service of the other [8].

Hence the VRP, like many other real life problems, involves relevant multiple objectives which are, more often than not, conflicting:

1. Minimization of total distance traveled.
2. Minimization of total deterioration of goods during transportation.
3. Maximization of fulfillment of emergent services.
4. Maximization of fulfillment of conditional dependencies of stations.

The conflict arises because improvement in one objective can only be made to the detriment of one or more of the rest of the objectives. It is noted that there may be more possible objectives that are not considered explicitly in this research.

It is desirable to study how to make an intelligent trade-off between the objectives and determine the most satisfactory or favorable vehicle routes. The successful consideration of the VRP in a multiple objective environment will provide an important management tool in many vehicle operations, bringing about a savings of resources and the increase of service satisfaction from customers.

Research Objectives

The objectives of this research are three fold. The first objective is to propose a VRP model for the multiple-vehicle, single-depot case where the conflicting multiple objectives are treated explicitly, and to develop an algorithm and an interactive procedure to determine the most satisfactory vehicle routes for it. The second is to develop a computer program of the algorithm that can solve the multiple criteria VRP and to perform computational experiments to evaluate and justify it with respect to some criteria corresponding to the multiple objectives. The third one is to develop a computer program of the interactive procedure that allows Decision Maker (DM) involvement in the solution process. The primary result of this research will provide management with more realistic and practical solutions for VRPs through multiple objective analysis. In addition, the results from this research can be extended to consider other important objectives to be accomplished in VRPs.

Research Procedure

In order to accomplish the research objectives, two phases are described as follows:

Phase I

Addressing Multiple Criteria VRP through Goal Programming.

1. Construct a mathematical model of multicriteria VRP in a Goal Programming framework and develop an algorithm to apply it to VRPs in a multiple objective environment.

2. Develop a computer program of the algorithm.
3. Carry out the computational experiemnt of the algorithm on three test problems of VRP, incorporating multiple objectives, and evaluate its performance by comparing the results with those obtained by savings algorithms for VRPs with a single objective, with respect to some criteria corresponding to the multiple objectives.

Phase II

Designing an Interactive Procedure.

1. Develop an interactive procedure for multicriteria VRP that relies on the progressive definition of DM's preferences along with the exploration of the criterion space, in order to reach the most favorable solution of the VRP with respect to the DM's preference.
2. Develop a computer program of the interactive procedure.

Outline of Succeeding Chapters

Chapter I, this chapter, defines the problem and states the objectives and the procedure of the research. Chapter II introduces the VRP and reviews the existing literature on VRP solution techniques. Chapter III discusses the concept of set of nondominated solutions, and introduces Goal Programming and interactive methods for multiple objective decision making. In Chapter IV, the algorithm for multicriteria VRPs is proposed. The algorithm consists of two major stages. Results of the evaluation study are presented in Chapter V. Chapter VI proposes the interactive procedure for multicriteria VRPs and its use. In Chapter VII, summary, conclusions, and recommendations for future study are offered.

CHAPTER II

BACKGROUND OF THE RESEARCH

Introduction

The basic routing problem is to construct a low-cost, feasible set of routes for a set of stations (nodes) and/or arcs by a fleet of vehicles. The VRP was first formulated by Dantzig and Ramser [19]. Since then, many researchers have been concerned with developing the solution methods for the VRPs. In this chapter, a brief review of the VRP is given, followed by a review of vehicle routing literature.

Vehicle Routing Problem

The effective management of vehicles for collection and/or delivery activities gives rise to a variety of problems generally known as "routing or scheduling problems." In its standard form the Vehicle Routing Problem (VRP) is to design a set of routes starting from, and ending at, a central depot, to service once only a number of geographically dispersed stations with a known quantity of some commodity, such that all stations are satisfied and that any restrictions on the capacity of vehicles, the duration of a route, or the times of visits to various stations are met. The "capacity of vehicles," "duration of a route," and "the times of visits" refer respectively to the maximum load allowed on each vehicle, the maximum distance each vehicle can travel in a day, and a given span of time within which services are allowed.

The objective of the VRP is to construct a sequence of routes optimizing an objective of either a total distance, time, cost, safety, or convenience. For example, in school bus routing, the objective is to minimize the total number of student-minutes on the bus since this measure is perceived to be highly correlated with safety [7]. In dial-a-ride services for the elderly or the handicapped, the primary objective is to provide convenient service to all users [7]. Measures of both safety and convenience have been identified in a quantifiable form to allow the problem to be viewed as an optimization problem.

It should be known, however, that in any practical VRP its basic form may be complicated by the presence of one or more added characteristics both to the constraints and to the factors contributing to the objective. Bodin et al. , [7] classifies VRP into seven catagories in terms of their characteristics:

1. The Traveling Salesman Problem (TSP), where no physical constraints regarding vehicles are involved, or the total distance and load are within the limits of one vehicle.
2. The Chinese Postman Problem, where the determination of the minimal distance cycle, that passes through every arc of a network at least one time, is required. No physical constraints are involved.
3. The Multiple Traveling Salesman Problem, where there is a need to account for more than one vehicle with a capacity constraint.
4. The Single-Depot, Multiple-Vehicle, Node Routing Problem, where all the stations scattered around a central depot are

required to be serviced by vehicles. The demand at each station is assumed to be deterministic and the physical and temporal constraints are involved. The problem is generally known as a standard VRP.

5. The Single-Depot, Multiple-Vehicle, Node Routing Problem with Stochastic Demands is identical to the standard VRP except that the demands are not known with certainty.
6. The Multiple-Depot, Multiple-Vehicle, Node Routing Problem, where the fleet of vehicles must serve several depots rather than just one. All other constraints from the standard VRP still apply.
7. The Capacitated Arc Routing Problem, where the specified demands of arc in a network must be satisfied by one of a fleet of vehicles. The physical constraints are involved.

A formulation of the standard VRP as a 0-1 integer problem is given below. This formulation is a simple modification of the one introduced in [15].

Let $x_{ijk}=1$ if vehicle k visits station j immediately after visiting station i. $x_{ijk}=0$ otherwise. The central depot is represented as station 0. The VRP is:

Minimize

$$Z = \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} (d_{ij} \sum_{k=1}^{M} x_{ijk}) \qquad (1)$$

subject to

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} \sum_{k=1}^{M} x_{ijk} = 1, \qquad j=0, 1, \ldots, N \qquad (2)$$

$$\sum_{\substack{i=0 \\ i \neq p}}^{N} x_{ipk} - \sum_{\substack{j=0 \\ j \neq p}}^{N} x_{pjk} = 0, \qquad k=1,2,\ldots,M,\ p=0,1,\ldots,N \quad (3)$$

$$\sum_{j=1}^{N} x_{0jk} = 1, \qquad k=1,2,\ldots,M \quad (4)$$

$$\sum_{i=1}^{N} (q_i \sum_{\substack{j=0 \\ j \neq i}}^{N} x_{ijk}) \leq Q_k, \qquad k=1,2,\ldots,M \quad (5)$$

$$\sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} d_{ij}x_{ijk} \leq T_k, \qquad k=1,2,\ldots,M \quad (6)$$

$$y_i - y_j + (N+1) \sum_{k=1}^{M} x_{ijk} \leq N, \qquad i \neq j=1,2,\ldots,N \quad (7)$$

$$x_{ijk} = 0 \text{ or } 1, \qquad \text{for all } i, j, k \quad (8)$$

y_i, $i=1,2,\ldots,N$ are arbitrary real numbers

where

d_{ij} = distance from station i to station j

q_i = service quantity (supply or demand) at station i

Q_k = capacity of vehicle k

T_k = maximum distance allowed for a route of vehicle k

N = number of stations

M = number of vehicles

The objective function (1) represents the minimization of total distance traveled by M vehicles. Alternatively, costs could be minimized by replacing d_{ij} by a cost coefficient c_{ijk} which depends upon the vehicle type. Constraints (2) state that a station must be visited exactly once. Constraints (3) state that if a vehicle visits a station, it must also depart from it. Constraints (4) ensure that a vehicle must be used exactly once. Constraints (5) are the vehicle capacity limitations. Similarly, constraints (6) are the vehicle travel distance limitations. A route is said to constitute a tour if, starting from a central depot, stations are visited exactly once before returning to the depot. A subtour may be defined as a route comprising some stations without the depot. Constraints (7) eliminates subtours and forces each route to pass through the depot. N^2-N subtour-elimination constraints are required when N stations are to be served. Constraints (8) are integrality conditions.

It is quite clear that the formulation of the VRP becomes unwieldly even for a modestly-sized problems, comprising an enormous number of variables and constraints. The VRP is NP-Complete, that is, it is a member of a large class of hard combinatorial problems for which no efficient polynominally-bounded algorithms are available. Given that the VRP is NP-Complete, known approaches for solving these problems optimally suffer from an exponential growth in computational burden with problem size.

Much attention has been given over the years to the study of the VRPs as management became increasingly aware of the need to control the rising costs of the physical collection and/or delivery activities by vehicles. Bodin et al. [7] states that the costs associated with operating vehicles and crews for collection and/or delivery purposes form an important component of total distribution costs and consequently small percentage savings in these expenses could result in substantial total savings over a number of years. When coupled with an effective management information system, the routing methodology can assume a crucial role in the operational planning of collection and/or delivery activities by vehicles. Mole [40] expresses the importance of VRPs in his survey report, in terms of "tactical" short-term viewpoints and "strategic' longer term concerns.

Due to these attractive points, many researchers, in recent years, have been concerned not only with obtaining an optimal solution but also with developing practical and economic heuristic methods for VRPs.

Example

In order to clarify the VRP further, consider a small problem involving five stations to serve and a single depot. A distance matrix is given in Table I, as is the list of service quantities that are to be collected for all stations. It is assumed that there are an unlimited number of 16-unit capacity vehicles available and that the travel distance by each vehicle is limited to 90 units. The objective is to construct a sequence of routes minimizing a total distance while meeting the restrictions given.

The optimal solution obtained is with routes 0-1-2-0 and 0-3-4-5-0. The distance of each is 45 and 85 units, respectively, yielding a total of 130 units. The routes are depicted graphically in Figure 2.

Literature Review of VRP Solving Techniques

Since the first mathematical formulation of the VRP by Dantzig and Ramser in 1959 [19], many researchers have been engaged in solving the problem of determining an optimum or near optimum solution for VRPs.

TABLE I

DATA FOR THE SAMPLE PROBLEM: DISTANCE MATRIX AND SERVICE QUANTITY

	0	1	2	3	4	5	Station	Quantity
0	-	20	30	50	60	40	0 (depot)	-
1	10	-	5	10	20	15	1	6
2	20	10	-	30	10	20	2	2
3	30	15	20	-	10	10	3	5
4	40	15	5	10	-	5	4	5
5	20	10	30	20	10	-	5	6

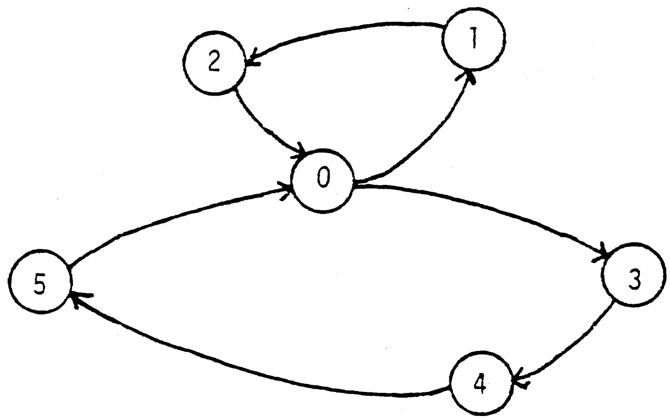

Figure 2. Graphical Depiction of the Solution to the Sample Problem

Basically, there are two types of algorithms that can be used to solve VRPs; optimal seeking and heuristic. The literature review concentrates mostly on the single-depot, multiple-vehicle and multiple-depot, multiple-vehicle cases.

Optimal Seeking Algorithms

Optimal seeking algorithms are ones that, in the absence of round-off or other errors, yield an exact solution in a finite number of steps. Since the VRP is NP-Complete in nature, however, iptimal seeking procedures cause excessive computational burden in solving problems. The nature of the growth in computation time and storage requirements is a function of problem size. If this growth is too rapid, the computational burden soon becomes prohibitive, even for moderate problem sizes, thereby limiting the applicability of a solution technique in a realistic environment where the problems encountered are typically large scale. The optimal seeking algorithms have been developed mainly on the basis of the branch-and-bound procedure of Little et al. [45], dynamic programming [4], and integer programming [55].

Christofides and Eilon [13] developed an optimal seeking algorithm based on the branch-and-bound technique of Little et al. [45] for solving the TSP. They transformed the VRP into a TSP by eliminating the real depot and replacing it by N artificial depots, all located in the same positions. The lower bound of the number of artificial depots N is determined by

$$N \geq \sum_{i=1}^{n} q_i/Q$$

where q_i is the quantity for station i (i=1,2,...,n) and Q is the vehicle capacity. Traveling from one artificial depot to another is prohibited by setting the distance between any two depots equal to infinity. The lower bounds for nodes of the decision tree are computed from the minimal spanning tree plus the shortest link, while checking the constraints on the capacity of vehicles and the duration of a route at each branch. A spanning tree is a configuration of n-1 straight lines passing through the n points and a minimal spanning tree is one with the shortest sum of links. Therefore, a lower bound for the minimal traveling salesman tour can be obtained by adding a suitable link, such as the shortest link in the network. The problem may be solved for several values of N and the best solution chosen. Though optimality can be guaranteed for small-size problems by this algorithm, the problem size is expanded as the number of artificial depots N are increased, resulting in a heavy computational burden. In fact, the largest size VRPs solved involve problems with ten or twelve stations.

Pierce in 1969 [48] extended the branch-and-bound technique of Little et al. [45] to a single cyclic VRP involving delivery time constraints such as due dates and earliest times for stations, and a more general cost objective function that considers a total variable cost reflecting additional time-independent costs dependent on the subsequences of pair of stations included in the route. These costs, for instance, might represent vehicle toll charges incurred in traveling from station i to j. At each branch, feasibility, bounding, and dominance tests are performed to eliminate dominated and nonfeasible branches from explicit elaboration, by incorporating the lower and upper bounds corresponding to each constraint. Though this procedure is limited to single-route problems, it could be extended to the multiple-route problems with additional computational effort.

Pierce also showed that the solution of the VRP could be found by a dynamic programming approach based on the procedure for solving TSP due to Bellman [4]. As in many dynamic programming approaches, computer storage would quickly become a problem, so only relatively small-sized VRPs could be solved.

Christofides, Mingozzi, and Toth [15] developed another exact branch-and-bound algorithm incorporating the improved computation method of lower bounds derived from the shortest spanning tree with a fixed degree at a central depot. In the solution of M-Traveling Salesman Problem (M-TSP) where M is a number of salesmen, the k-degree center tree (k-DCT) is defined by removing $y \leq M$ arcs adjacent to a central depot and M - y arcs not adjacent to a central depot from each of the remaining M-y routes-- one arc from each route --the resulting graph is k-DCT with k = 2M - y. A lower bound of the M-TSP is computed from the shortest spanning k-DCT for several k values and it is then employed for the lower bound of the VRP at each branch. The shortest spanning k-DCT is calculated efficiently using the Lagrangean penalty procedure.

This algorithm is based on the idea that the value of the solution to the M-TSP is a lower bound to the value of the solution to the VRP using M vehicles, because the VRP may be considered as the M-TSP with additional constraints. The computation procedures, however, are further complicated in the nonsymmetric case, where the distances between two stations are different upon direction. The computational results showed that the standard VRPs up to 25 stations could be solved exactly. The basic difference between this and Christofides and Eilon's algorithm is that, in the computation method of lower bounds, the former separates the problem into several possible tours and the latter considers it as the large single tour. However, it is still not clear that this improvement of lower bounds can contribute significantly to guarantee an optimal solution to the VRP in reasonable computation time [15].

Two procedures have been developed with cutting plane algorithms. Balinski and Quandt [3] formulated a delivery problem as a 0-1 integer programming model. Their problem consists entirely of common carrier route. For n stations and a set of permissible routes J, the formulation is as follows:

$$\text{Minimize} \quad Z = \sum_{j \varepsilon J} c_j x_j$$

subject to

$$\sum_{j \varepsilon J} a_{ij} x_j = 1, \qquad i = 1,2,\ldots,n$$

$$x_j = 0 \text{ or } 1, \qquad j \varepsilon J$$

where

c_j = the cost incurred with the jth route

a_{ij} = 1 if station i is included as a stop in the jth route and

a_{ij} = 0 otherwise

In their problem, the set J represents permissible alternative routes satisfying the restrictions about the vehicle, and cost c_j is determined as a function of total weight shipped over the route, the number of stops on the route, and the most distant stop. This formulation is, unfortunately, not very useful as there is likely to be an enormous number of feasible routes or variables x_j, $j \varepsilon J$. However, the authors managed to reduce this number by employing the concept of "dominated tours" -- tours which could never be part of an optimal solution. Using Gomory's cutting plane method [55, pp 178-205], they found approximate solutions to problems of up to 270 stations and 15 feasible routes. However, any realistic application is likely to contain considerably more. This formulation was further extended by Foster and Ryan in 1976 [22], to incorporate restrictions on work load, coverage, and service that occur in real world VRPs.

Another integer programming formulation has been introduced by Christofides, Mingozzi, and Toth [15]. The formulation is as described in equations of (1) - (8) in page 11. The formulation given has an enormous number of variables and constraints, even for a small-size VRP. Thus its value lies not in its practicality as a way of solving the VRP directly, but more in its ability to yield insights which may be useful in the development of heuristics.

In summary, it may be true that finding an efficient optimal seeking algorithm is an impossible task, because the VRP is an NP-Complete problem. It is noted that any heuristic procedure which can provide good lower bounds on the optimal value of the VRP can be embedded within a branch-and-bound approach to yield an exact procedure.

Heuristic Algorithms

As mentioned earlier, optimal seeking algorithms have severe limitations when employed in practical situations due to their computation requirements. Therefore, various heuristic approaches have been developed during the past twenty-five years. Another reason to investigate approximate methods is that procedural steps can be kept simple enough so that the problem solver does not lose sight of the overall view of the problem, thus enabling him to make the best use of his intuition and judgment [46].

Heuristics for the VRP can be classified into two classes: (1) Route First (RF) and (2) Cluster First (CF). In the RF methods, routes are sequentially constructed initially. This is done by either accepting links successively as part of the initial solution or inserting new stations one at a time into existing partial routes, on the basis of a special evaluation system which indicates the potential worth of each possible choice. The initial solution constructed may then be subject to some improvement strategies. In the CF methods, instead of attempt-ing to initially complete routes, the set of stations is clustered into subsets. Once the stations have been clustered, each cluster is subjected to a TSP method in order to determine the best sequence of stations for each route.

Route First Methods.

An early method is that of Dantzig and Ramser [19]. It starts from connecting each station with a central depot and excluding permanently the links which may cause routes to exceed the vehicle capacity during the aggregation process. The procedure continues the successive aggregation of a large number of elementary partial routes without exceeding the vehicle capacity, based on the criterion of the Delta-function that indicates how much the total distance will decrease by linking two seperated partial routes, achieving a reduction in a travel distance at each stage. Each partial route is considered as a station with a shortest distance, at each stage of the aggregation procedure. The shortest distance is obtained by solving the partial route as a TSP. As a result of initial exclusion of the links to prevent any routes from exceeding the vehicle capacity, their heuristic tends to lay more emphasis on filling vehicles to near capacity than on minimizing the total distance. It has failed in obtaining good solutions also because when any two stations become linked in the aggregation, they remain aggregated during the procedure.

Following this work, Clarke and Wright [17] introduced a way of quantifying the direct link between any two stations, according to the potential "savings" involved. Their heuristic, which is still one of the most widely used today [9, 59], begins by designating a seperate vehicle to each station. The total distance is progressively shortened, by repeatedly joining the point-pair of maximum "saving," providing this is feasible, at the same time dispatching one less vehicle. The "saving," s_{ij}, is computed by:

$$s_{ij} = d_{i0} + d_{0j} - d_{ij}$$

where d_{ij} represents the travel distance from station i to j and i,j = 0 denotes a central depot. Figure 3 illustrates the "saving" s_{ij} by joining two stations i and j to form one route.

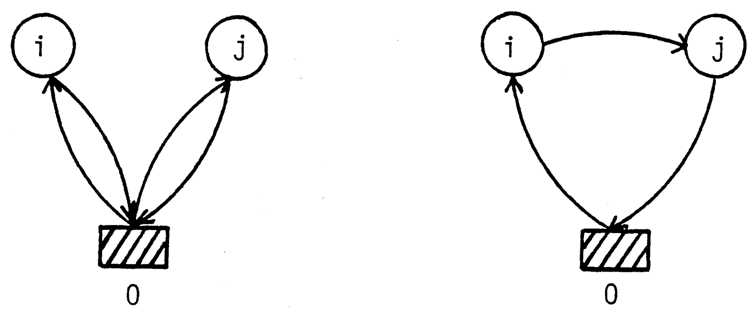

Figure 3. Link Replacement Scheme Leading to Potential Saving in a Route Structure

This heuristic has, however, three major deficiences. First, it does not look ahead to discover the consequence of taking advantage of a particular "saving" which is not a maximum. Secondly, its decisions are permanent. Once a link is accepted as part of a route it is never discarded, which results in an under-utilized vehicle and consequently a poor solution. Thirdly, it typically requires a prior calculation of a "savings" file consisting of all pairs of points at a considerable expense. There have been a number of attempts to overcome these shortcomings.

Gaskell [23] suggested slightly different methods of "savings" calculation which placed different emphasis upon the spatial distribution of stations. Two measures of s_{ij} are:

1. $s_{ij} = (d_{0i} + d_{j0} - d_{ij})\ (\overline{d} + |d_{0i} - d_{j0}| - d_{ij})$

   where $\overline{d}$ is the average of all d_{0k}

2. $s_{ij} = d_{0i} + d_{j0} - 2d_{ij}$.

These methods are intended to give greater priority to stations on the depot side and lead to the generation of predominantly narrow petal-shaped routes. He also proposed two versions of the Clarke and Wright procedure [17], the "multiple," in which many routes are developed in parallel, and the "sequential," in which each route is completed before the next is started. Robbins et al. [50] have shown, however, using randomly generated problems, the Clarke and Wright method [17] to be at least as good as Gaskell's "savings" calculations on the problems examined.

A variation on the Clarke and Wright method was produced by Yellow [63], which eliminates the need for a precomputed "savings" file. Instead, it incorporates a geometric search technique on an ordered list of the polar coordinates of the stations, to search for the link of the highest "saving,"

$$s_{ij} = d_{0i} + d_{j0} - u\, d_{ij}$$

where u represents a route shape parameter. The algorithm generates only one route at a search. A computational advantage was recognized over Gaskell's method.

An approach to incorporate "look ahead" schemes into the Clarke and Wright method where the selection of a particular link may cause its stations to remain permanently in a particular route, was employed by Tillman and Hering [57]. They extended their decision horizon to consider in advance some of the later effects of linking stations, by choosing two pairs of stations with the best "saving" such that the second best feasible pair may also be chosen. This way of choosing the best two feasible pairs of stations maximizes the "savings" over four stations, not two. This could be extended to three or more. However, this modification may require an inordinate amount of computational time.

A similar approach was also adopted by Homes and Parker [27]. They explored the consequences of choosing each of several high "savings" links at each stage for use, by temporarily prohibiting the links of certain stations that yield high "savings" but adversely affect subsequent links, in a partial tree search guided by the "savings" rationale. They justified, also through computational experiment, a common property in VRPs that the reduction of total distance always leads to the subsequent reduction of the number of routes.

Buxey [9] modified the savings approach by introducing a probabilistic element. Rather than always accepting a link representing the next biggest "saving" on the file, he selected the next link on the basis of a Monte Carlo simulation and assigned it a specific direction of travel. In the simulation, a random choice of the links is made according to the probability distribution, that is,

$$\text{probability}(I) = (s_I)^M / \sum_{I=1}^{J} (s_I)^M$$

where I represents a station-pair (i,j), M is a weighting factor, and s_I is a "saving" of I. The method appeared to yield improved results for certain well-known test problems. However, it has been found from several computational results that these elaborations of the savings methods produce marginal improvements as compared with substantially increased computation times.

Mole and Jameson [41], also applied a "savings" based selection rule in a generalized form, that picks the most promising new station and describes the distance reduction of inserting it between two existing stations in a partial route. The generalized "savings," s_c (i,j), by including a station c between stations i and j in a route, is given by:

$$s_c\,(i,j) = v\,d_{0c} + u\,d_{ij} - d_{ic} - d_{jc}$$

where v and u represent route shape parameters. The positive parameters ensure that each partial tour does not intersect itself, a condition which obviously holds in any good solutions. This sequential approach preserves the computational advantage associated with the simple ranked selection procedure since it does not require a precomputed "savings" file. Finally, a refinement phase is employed to improve the final routes by reassigning a station to a different route, owing much to the earlier work of Wren and Holliday [62] to be described later.

Golden, Magnanti and Nguyen [26] divided the area containing all stations into a series of identical rectangles and applied a modified savings method, utilizing only those "savings" which result from linking stations within the same or neighboring rectangles. They also attempted to improve the final routes constructed.

Christofides and Eilon [13] proposed a method which builds an initial solution using the basic "savings" scheme. This is then improved by using a concept called r-optimality. Basically, it involves replacing r links in the solution by another r links if the total distance is reduced and feasibility is maintained. When it is impossible to find such an improvement the routine is terminated. This can be done for progressively increasing values of r. The r-optimality method was developed for the TSP by Lin and Kernighan [41]. This refinement procedure has been applied to the VRP by many researchers [14, 41, 50, 52]. A feasible starting route is, however, required, and the results are initial-solution-dependent.

Russell [52] presented an effective heuristic MTOUR for the M-TSP with strict side conditions of due dates or time intervals for stations as well as total load or distance associated with each tour, which is directly applicable to the VRP. The MTOUR applies Lin's 3-optimality procedure [44] to the initial feasible routes constructed in several ways such as random routes, the Clarke and Wright method [17], or the SWEEP method [24]. The essential modification that MTOUR imparts to Lin's procedure is the explicit enforcement of the various side conditions.

Tillman and Cain [56] proposed a solution technique for multi-depot VRPs using the "savings" concept. The prodedure starts with an initial solution consisting of servicing each station exclusively by one route from the closest depot. It successively links pairs of points in order to decrease the total cost. One basic rule assumed in the algorithm is that the initial assignment of stations to the nearest depot is temporary, but once two or more stations have been assigned to a common route from a depot, the stations are not reassigned to another depot. In addition, as in the original savings algorithm, stations i and j can be linked only if neither i nor j is interior to an existing tour. At each step, the choice of linking a pair of stations i and j on a route from depot k is made in terms of the "savings," ${s_{ij}}^k$, when linking i and j at k. Stations i and j can be linked only if no constraints are violated. The formula for "savings" is given by :

$$s^k_{ij} = \bar{d}^k_i + \bar{d}^k_j - d_{ij}$$

where

$$\bar{d}^k_i = \begin{cases} 2 \min\limits_t \left\{ d^t_i \right\} - d^k_i & \text{if } i \text{ has not yet been given a permanent assignment} \\ d^k_i & \text{otherwise} \end{cases}$$

d^k_i = the distance between station i and depot k.

It should be noted that the performance of many "savings" based algorithms varies considerably with the characteristics of problems tested, such as size, journey restrictions, spatial distribution of stations and depot location, and therefore no algorithm has been praised in absolute terms of its quality [9, 20, 35, 38, 40, 60]. However, the "savings" based heuristics have yielded acceptable results and proved commercially popular due to an advantage in speed and ease of application [35].

Using an approach that is completely different from the Clarke and Wright method, Williams [66] presented a proximity priority searching method. The method is based on joining stations furthest from the depot to the closest feasible stations within the immediate proximity, producing circumferential routes. Because stations are added sequentially, problems involving service time restrictions can also be effectively handled. It was concluded, on the basis of optimality and computation time, that the method was as good as other "savings" based techniques.

Most heuristics for the VRP are primal in that the solution is built up by retaining feasibility while gradually approaching optimality. By contrast, Cheshire, Malleson and Naccache [11] presented a dual technique that retains local optimality at each iteration while gradually approaching feasibility. The cost, that is made up of a distance function and a penalty function against the violation of constraints on the capacity of vehicles, the duration of a route and the delivery time for stations, is employed as the objective function to be minimized. Once the complete but infeasible solution is constructed by including promising stations one at a time in the partial routes that are locally optimized through an improvement procedure of repositioning of any station already included, the proportionality constants of the penalty function, associated with each violated constraint, are increased in value. The proportionality constants are initially set to some low value. Each route of the solution is then checked for cost reduction using the increased proportionality constants. This complete process is repeated until a feasible solution of routes is obtained.

Numerical results were comparable with those of Foster and Ryan [22].

Finally, Doll [20] proposed the simplest RF procedure of all, on the basis of his general rules. According to the procedure, a scheduler estimates the number of schedules required per day and the number of vehicles, using equations, identifies any geographical barriers, and creates a route as much like a tear drop as reasonable -- shaped routes on a scale map of the service area.

Cluster First Methods.

Wren and Holliday [62] presented a method which uses information about the spatial layout of the stations in scheduling vehicles from one or more depots to a number of stations. Each station is provisionally assigned to its nearest depot for the purpose of ordering stations. An axis for each depot is determined which passes through the most sparsely populated area and the stations are then sorted according to the order of the angular coordinates from their assigned depots. The stations in order are considered one at a time starting from any axis, and are either added to existing routes, used to create new ones, or assigned to another depot, in order to minimize the distance increase with the consideration that feasibility must be maintained. The initial routes produced are then passed through an exhaustive refinement process that reassigns stations to different routes and resequences stations on a route. Finally, the axes are rotated through 90°, 180° and 270°, and the process is repeated at each position until the best solution is obtained. The computer time required was about 50 times that of the Clarke and Wright approach.

A similar heuristic was suggested for a single depot by Gillet and Miller [24]. In their so-called SWEEP algorithm, the stations 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 or decreasing list of these angles until any given constraints are violated. The procedure of the sweep is repeated until the last station in the list is assigned. After a 360° sweep is completed, the stations in each route are sequenced by a TSP method. The computer time increased linearly or quadratically with the average number of stations per route, restricting the algorithm to problems as small as 60 stations when there were about 30 stations per route.

A formulation equivalent to that given in Balinski and Quandt [3] was employed by Foster and Ryan [22]. The formulation is:

$$\text{Minimize } Z = \sum_{j \varepsilon J} (V + c_j) x_j$$

subject to

$$\sum_{j \varepsilon J} a_{ij} x_j = 1, \qquad i = 1,2,\ldots,n$$

where

J = a set of all feasible routes

V = the mileage-equivalent cost of each vehicle

c_j = the cost incurred with jth route

a_{ij} = 1 if station i is included as a stop on the jth route and

a_{ij} = 0 otherwise

To avoid enumerating all feasible routes x_j over a vast feasible region in the Integer Linear Programming (ILP) model of Balinski and Quandt, the authors relax the solution space by enumerating only routes with special characteristics derived from the observation that the optimal solution is generally composed of the radial contiguous routes about a central depot (termed "petal" routes).

In the solution approach used, they relax the integrality requirement of decision variables x_j and define the reduced set of feasible tours that follow "petal" routes, thus providing a much faster rate of convergence to the solution of the over-constrained LP model. For a solution to the resulting LP to be interpreted as a schedule, one must ensure that the variables have values of only 0 or 1. Though this can be done using a standard branch-and-bound technique, they applied cutting planes [55, pp. 177-223] to the revised simplex method [16, pp. 100-102]. Using information provided by the LP solution of the over-constrained problem, the over-constraints are then progressively relaxed to expand the set of feasible routes. The authors were able to find approximate solutions to problems with up to 100 stations in reasonable computing time.

Though these CF methods may generate good solutions, they have two important drawbacks in application. First, they cannot be adopted in the case where the distances between stations are nonsymmetrical because the initial clustering process is carried out by using information about the spatial layout of the stations, i.e., polar coordinates with the depot as origin. Secondly, they usually exhibit much longer computation times than RF methods while it is uncertain that their solutions are of high quality. However, on the other hand, a great advantage when groups of neighboring stations are preselected for a single route in the CF methods is that the VRP becomes a set of seperate TSPs for which many successful algorithms are available.

The interactive use of a computer program combined with a powerful VRP algorithm can be a valuable tool in the hands of a skilled scheduler with detailed knowledge of the particular requirements of his customers, and so some successful programming packages have been developed very recently. In real situations, the successful result of vehicle operation depends critically on the judgment of the scheduler, who can apply his own skills and knowledge to full effect in conjunction with the speed and flexibility of the computer program.

Interactive computerized vehicle algorithms have been developed by Fisher et al. [21], Christofides [12], and Cheshire et al. [11]. For depots with a small number of service stations, however, there may be merit in providing improved simple tools for use by the human scheduler, without employing a computerized or a specific algorithm (see Robertson [51], and Krolek et al. [36]). The methods may not guarantee optimal routes, but they can usually be relied upon to produce cost improvements in even small collection or distribution systems. The human involvement in the VRP is also supported by Doll's argument [20] that any saving achieved in vehicle operations have been due to the careful, systematic review of operations by schedulers, not to the quality of the solution heuristic.

Other Heuristic Methods.

The heuristics for VRPs mentioned so far have been developed for the deterministic case. Recently, the stochastic situation, where demands or supplies at stations are probabilistic, has been considered in the literature. All vehicles must leave from and eventually return to a central depot, while satisfying certain constraints and probabilistic station demands.

Golden and Stewart [27] assumed that the demand at each station i could be modeled by a Poisson distribution with mean λ_i and that demands at stations were mutually independent. They then developed an efficient heuristic solution procedure for generating a set of fixed vehicle routes. This algoithm first determines the artificial vehicle capacity $\bar{u}$ based on the degree of risk allowance that the total route demand exceeds the actual vehicle capacity c, probability $(x \geq c)$, where x is the total route demand. The Clarke and Wright method is then applied with λ_i $(i = 1,2,\ldots,n)$ as fixed demands and $\bar{u}$ as vehicle capacity in order to determine a fixed set of routes.

Golden and Yee [28] extended the previous work to the case where other appropriate probability distributions, such as binominal, negative binominal and gamma distributions, were assumed and demands were correlated due to factors such as seasonality or competition. The solution procedures are the same as in the case of a Poisson distribution, while using the different equations for determining $\bar{u}$ for each distribution.

Cook and Russell [18] performed a simulation study to evaluate the effectiveness of the deterministically generated routes based on mean values, using Russell's MTOUR method [52], when demands and travel times varied stochastically. The simulation analysis implied that the heuristics developed for deterministic VRPs can also generate an effective solution to the stochastic case.

In summary, a significantly large proportion of the researchers have examined the Clarke and Wright method and proposed variations to overcome its shortcomings. The reason for this may be related to the simplicity of the procedure and ease of application. Whereas the single-depot VRP has been studied widely, the multi-depot problem has attracted less attention. The relevant literature is represented by only a few papers. Relatively little research has been conducted on the stochastic VRP. Not surprisingly, the available reports [22, 24, 62] give an indication that the RF methods are inferior to the CF methods with regard to the minimization of an objective. However, the former have an advantage in speed, and also in ease of application, and have proved commercially popular. In applying one of the algorithms to a VRP in a real situation, consideration must be given to the algorithm because some rigid restrictions or assumptions have already been given to the procedure. Finally, it is noted that there are now many interactive computer programs available commercially and more attention should be given to the development of efficient interactive programs for VRPs.

Table II gives a general discription of models of both exact and heuristic algorithms mentioned in the Literature Review. Starting from Dantzig and Ramser's method in 1959, all of the algorithms have been developed with regard to the minimization of a single objective, either distance traveled, cost, or time, while strictly holding the constraints given. However, the collection or delivery problems inherent in the VRP issue may not lend themselves to a model construction concerning only one objective and may involve multiple objectives. As Table II illustrates, no algorithm for obtaining solutions for VRPs in a multiple objective environment has been developed.

Summary

A brief review and literature survey of the VRP is presented. The survey demonstrates an increasing importance of the VRP. VRPs can be

TABLE II

MODEL DESCRIPTION OF ALGORITHMS MENTIONED IN LITERATURE REVIEW

	Algorithm (Prog.) Developer /Reference number	Single-objective Deter-ministic	Single-objective Stocha-stic	Multi-objective Deter-ministic	Multi-objective Stocha-stic	Constraints*	Published Year
Opitmal seeking algo.	Balinski & Quandt [3]	x				1,2,4	1964
	Christofides and Eilon [13]	x				1,2,4	1969
	Pierce [48]	x				3	1969
	Christofides et al. [15]	x				1,2,4	1981
Heuristic algo.	Dantizig and Ramser [19]	x				1,4	1959
	Clarke and Wright [17]	x				1,4	1964
	Gaskell [23]	x				1,2,4	1967
	Christofides and Eilon [13]	x				1,2,4	1969
	Yellow [63]	x				1,2,4	1970
	Tillman and Hering [57]	x				1,2,4	1971
	Gillet and Miller [24]	x				1,2,4	1971
	Tillman and Cain [56]	x				1,2,5	1971
	Wren and Holliday [62]	x				1,2,3,5	1972
	Homes and Parker [29]	x				1,2,4	1976
	Mole and Jameson [41]	x				1,2,4	1976
	Foster and Ryan [22]	x				1,2,4	1976
	Golden et al. [26]	x				1,2,4	1977
	Russell [52]	x				1,2,3,4	1977
	Golden and Stewart [21]		x			1,4	1978
	Buxey [9]	x				1,2,4	1979
	Golden and Yee [28]		x			1,4	1979
	Doll [20]	x				1,4	1980
	Cheshire et al. [11]	x				1,2,3,4	1982
	William [61]	x				1,2,4	1982
Inter-active prog.	Cheshire et al. [11]	x				1,2,3,4	1982
	Fisher et al. - [21]	x				1,2,3,5	1982

*1. Vehicle capacity

2. Vehicle travel distance

3. Due date or time interval for stations

4. Single-depot

5. Multi-depot

solved using many algorithms. Some procedures are exact while others are heuristic. Optimal seeking procedures generate optimal solutions but are only practical for small-size problems. Large-scale problems must be solved by heuristic techniques. Of the heuristics, Clarke and Wright's [17] and Gillet and Miller's [24] methods have been given much attention. Many researchers have extended the concepts of the two methods to produce their own procedures. Recently, interactive computer programs have been developed. However, all of the studies have been concerned with only a single objective. No algorithm has been developed for obtaining solutions for VRPs with relevant multiple objectives to be achieved. The following chapter discusses the multiple objective optimization analysis.

CHAPTER III

MULTIPLE OBJECTIVE OPTIMIZATION ANALYSIS

Introduction

Since the advance of operations research as a scientific approach to decision making in the military operations of World War II, a variety of mathematical tools or systematic procedures have been developed and applied to problems in many areas which are largely characterized by the need to allocate limited resources to a collection of activities in application areas [64]. These techniques share a common feature: the formulation of a single criterion or objective function, and the optimization of an objective function subject to a set of prescribed constraints. As such, a large number of problems can be considered, where it is of interest to do one of the following: maximize profits, minimize total distance traveled, minimize costs, and so on.

In the last two decades there has been an increased awareness of the need to identify and consider several objectives simultaneously, many of which are in conflict, in the analysis and solution of many problems. In particular, some of these problems are those derived from the study of large-scale systems such as the complex resource-allocation systems in the areas of industrial production, urban transportation, health delivery, layout and landscaping of new cities, energy production and distribution, wildlife management, operation and control of the firm, local government administration, and so on. The multiple objective formulation of the problems have provided a more realistic modeling approach and afforded the Decision Maker (DM) in charge the ability to make intelligent trade-off decisions about the different objectives. Mathematically, the problems can be represented as:

$$\text{Maximize } [f_1(\overline{x}),\ f_2(\overline{x}),\ldots,f_k(\overline{x})]$$

subject to

$$g_i(\overline{x}) \leq 0, \qquad i=1,2,\ldots,m$$

where $\overline{x}$ is an n dimensional decision variable vector. The problem consists of n decision variables, m constraints and k objectives. Any or all of the functions may be nonlinear. Because of the conflicting nature, there is usually no solution to the problem which optimizes all k objectives simultaneously. Thus for multiple objective optimization problems, one may be interested in selecting one of the possible "non-dominated" solutions as the best compromise solution.

In turn, the recognition of multiple objectives in systems analysis has motivated the development of many multiple objective (criterion) decision making techniques. These may be classified into four catagories in terms of their characteristics [25]:

1. Techniques for generating the nondominated solutions set.
2. Continuous and discrete techniques that rely on prior articulation of preferences by the DM.
3. Techniques that rely on progressive articulation of preferences.
4. Techniques with posterior articulation of preferences.

Such classification recognizes the comparative advantage of bringing the DM's preferences into the different stages of an analysis in order to generate or rank the various alternative solutions. The applications of multiple objective models in the process of decision analysis, as opposed to a single objective in past practice, will be broadly and rapidly expanded. Figure 4 depicts a sequence of steps to follow in multiobjective analysis, suggested by Goicoechea et al. in 1982 [25].

In this chapter, the concept of the nondominated solutions set and the introduction of Goal Programming and interactive methods for multiobjective decision making, which are referred in the next chapters, are briefly described.

Set of Nondominated Solutions

A nondominated solution is one in which no one objective function can be improved without a simultaneous detriment to at least one of the other objectives in a multiple objective optimization problem. That is, given a set of feasible solutions X, the set of nondominated solutions is denoted S and defined as follows (assuming more of each objective function is desirable):

$$S = \Big\{ x\text{: } x \in X, \text{ there exists no other } x' \in X \text{ such that } f_i(x') > f_i(x) \text{ for some } i = 1,2,\ldots,p \text{ and } f_j(x') \geq f_j(x) \text{ for all } j \neq i \Big\} .$$

Thus it is evident from the definition of S that as one moves from one nondominated solution to another nondominated solution and one objective function improves, then one or more of the other objective func-

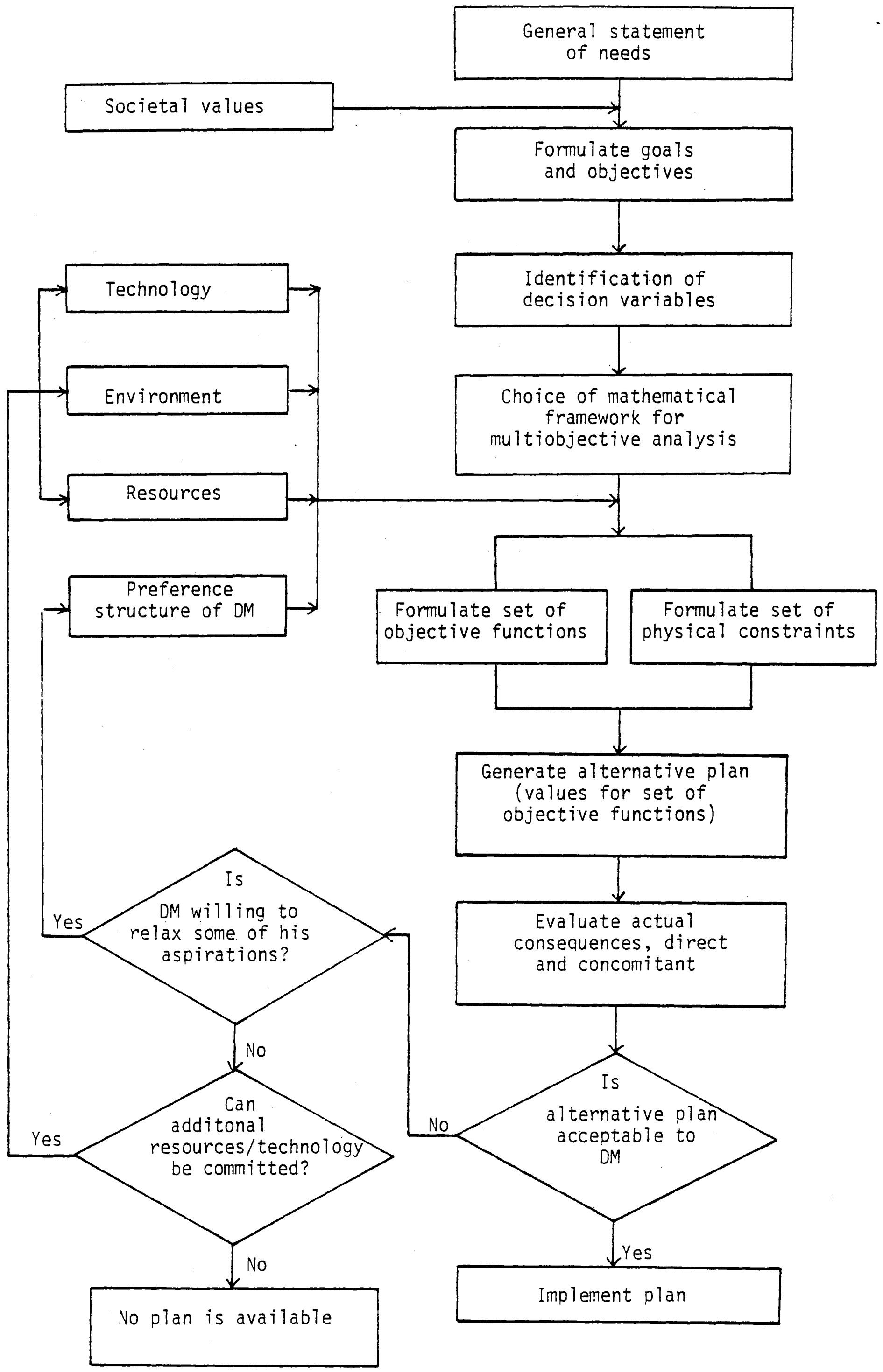

Figure 4. A Sequence of Steps for Multiobjective Analysis

ions must decrease in value.

Figure 5 [64] provides some graphical explanation of the concept of a "nondominated solutions set," using the maximization problem with two objective functions, f_1 and f_2. Observe that the point x in a set of feasible solutions X, is dominated by all points in the shaded subregion of X, indicating that the levels of both objective functions can be increased simultaneously. Only for points in N does this subregion of improvement extend beyond the boundaries of X into the infesible region. Thus the points in N are only the set of non-dominated solutions and they make up the heavy boundary of X. All other points of X are dominated.

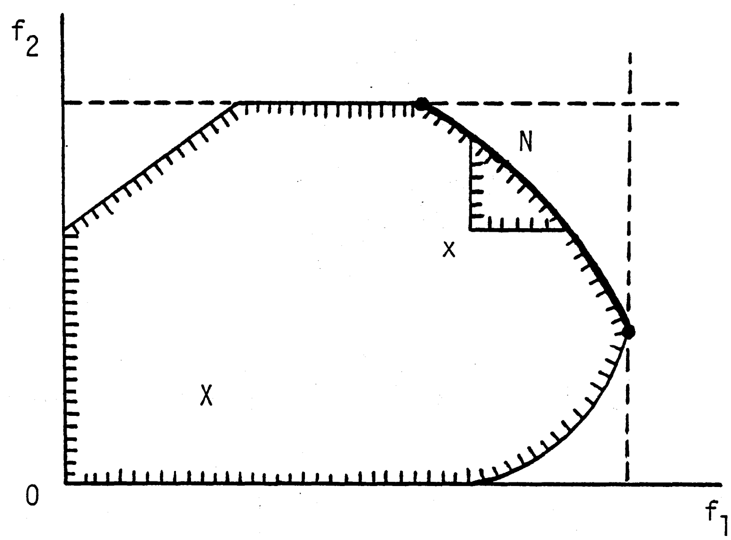

Figure 5. Set of Nondominated Solutions [64, p.70]

The methodology of multiparametric decomposition [64] projects various combination of preferences of multiple objectives in terms of corresponding nondominated solutions obtained. This allows the DM to apply his preferences imprecisely in terms of weights or rates in objectives and form a base for an interactive decision making procedure.

Goal Programming

A decision situation is generally characterized by multiple objectives. Some of these objectives may be complementary, while others may be conflicting in nature. Goal Programming (GP), a continuous method with prior articulation of preferences, requires the DM to specify a goal for each objective function and a priority structure of the various goals. A preferred solution is then defined as the one which minimizes the sum of the deviations from the prescribed set of goal values, on the basis of the preemptive goal priority. Therefore, the model implemented by GP is especially useful in providing the capability of evaluating different strategies under various assumed goal levels and/or varying the DM's policies with regard to the goal priority structure.

GP was originally proposed by Charnes and Cooper in 1961 [10] for a linear model. It has been further developed by Ijiri [34], Lee [42], and Ignizio [32]. Ignizio in 1976 extended the formulation of GP to linear integer and nonlinear forms.

The typical GP model is stated as follows:

$$\text{Minimize } S_0 = \sum_{i=1}^{k} P_i \ (w_i^- n_i + w_i^+ p_i)$$

subject to $x \in X$

$$f_i(x) + n_i - p_i = T_i$$

$$n_i p_i = 0$$

$$n_i,\ p_i \geq 0, \quad i = 1,2,\ldots,k$$

where

T_i = the goal (target) set by DM for the objective i

n_i = the negative deviation from the goal i

p_i = the positive deviation from the goal i

w_i^-, w_i^+ = the relative weights to the negative and positive deviations from the goal i.

To express preference for deviations, the DM can assign relative weights w_i^-, w_i^+ to negative and positive deviations, respectively, for each target, T_i. Since we are minimizing, choosing the w_i^+ to be larger than w_i^- would be expressing preference for under-achievement of a goal.

In addition, GP allows the DM to have the flexibility needed to deal with cases with conflicting multiple goals [25]. Essentially the DM can rank goals in order of importance to him. That is, the goals are classified into k ranks and a priority level P_i (i = 1,2,...,k) is assigned to the deviation variables associated with the goals. The P_is in the achievement function S_0 are preemptive priorities such that $P_i >>> P_{i+1}$. This implies that no number L, however large, can make $LP_{i+1} \geq P_i$ and so goal i has absolute priority over goal i + 1.

The solution procedure for the GP model consists of first minimizing the deviational variable(s) with the highest priority level, P_1, to the fullest possible extent, and when no further improvement is possible in a higher priority order variable(s) then the next priority order variable(s) is considered for minimization. This process continues until the variable(s) with the lowest priority level P_k is minimized. Thus, a solution is obtained in terms of a given hierarchy of the goals and is called a satisfactory solution.

Typically, there are two approaches for solving the GP problem. The one which has probably received the most attention in the literature involves the use of an approach which is basically an extension of the so-called Two Phase method of conventional linear programming. This modification of the simplex method, the Multiphase technique, is discussed in detail in [31, 32]. The second approach is called Sequential Linear Goal Programming (SLGP). The underlying basis for this method is the sequential solution to a series of conventional linear programming models.

The SLGP procedure is somewhat like dynamic programming where a complex multiple objective optimization problem is decomposed into a series of single objective optimization sub-problems according to priority levels [54]. Ignizio [31, p. 403] summarizes the procedure: Given the linear GP model, first consider just the portion of the achievement function and the goals associated with priority level 1. This results in the establishment of a single objective linear programming model given as:

Minimize $a_1 = P_1\ (w_i^- n_i + w_i^+ p_i)$

subject to

$$x \in X$$

$$f_i(x) + n_i - p_i = T_i$$

$$n_i p_i = 0$$

$$n_i, \; p_i \geq 0, \quad \text{for } i \in P_1.$$

That is, the first term in the achievement function is minimized, subject only to those goals in priority level 1. Once this is done, the best solution to the model is obtained, designated as a_1*. The next priority level is considered next. Here the second term in the achievement function, a_2, is minimized. However, it must be done subject to:

1. All goals at priority 1.
2. All goals at priority 2.
3. Plus an extra goal (or rigid constraint) that assures that any solution to priority 2 cannot degrade the achievement level previously obtained in priority 1, that is, a_1*.

This procedure is continued until all priorities have been considered. There are ways to shorten the procedure, as discussed in [31]. The solution to the final linear programming model is then also the solution to the equivalent linear GP. Sharif [54] points out that (1) in SLGP the objective functions are optimized directly, while in the Multiphase technique the objective functions are converted into constraints and the deviations from set goals are minimized and (2) for SLGP various solution methods are applicable depending on the characteristics of the objective functions, constraints, and decision variables, while for the Multiphase technique the application of the modified simplex method is restricted to certain GP problems.

Interactive Methods for Multiobjective Decision Making

This class of methods does not assume a global optimization but rather relies on the progressive articulation of the DM's preferences along with the exploration of the criterion space. Much work has been done recently on this class of methods [30, pp. 9-10]. Goicoechea [25] points out that the methods of progressive articulation of preferences are essentially predicated on certain assumptions about the psychology of the decision-making process.

The progressive articulation takes place through a DM-Machine or an Analyst-Machine dialogue at each iteration. At each such dialogue, the DM is asked about trade-offs or preferences on specific achievement levels of the objectives based on the current solution (or the set of current solutions) obtained by an algorithm. This information is used by the algorithm to generate a new solution. The DM then has an opportunity to provide new information which again serves as input to the algorithm. This process is repeated until the DM accepts a current achievement level of the objectives as the most favorable solution. Consequently, the methods require greater DM's involvement in the solution process than other techniques. Figure 6 depicts a general sequence of steps to follow in an interactive procedure.

These methods assume that the DM is not able to provide "a priori" preference information because of the complexity of the system, but that he is able to indicate preference information on a local level to a particular solution. As the solution process continues, the DM not

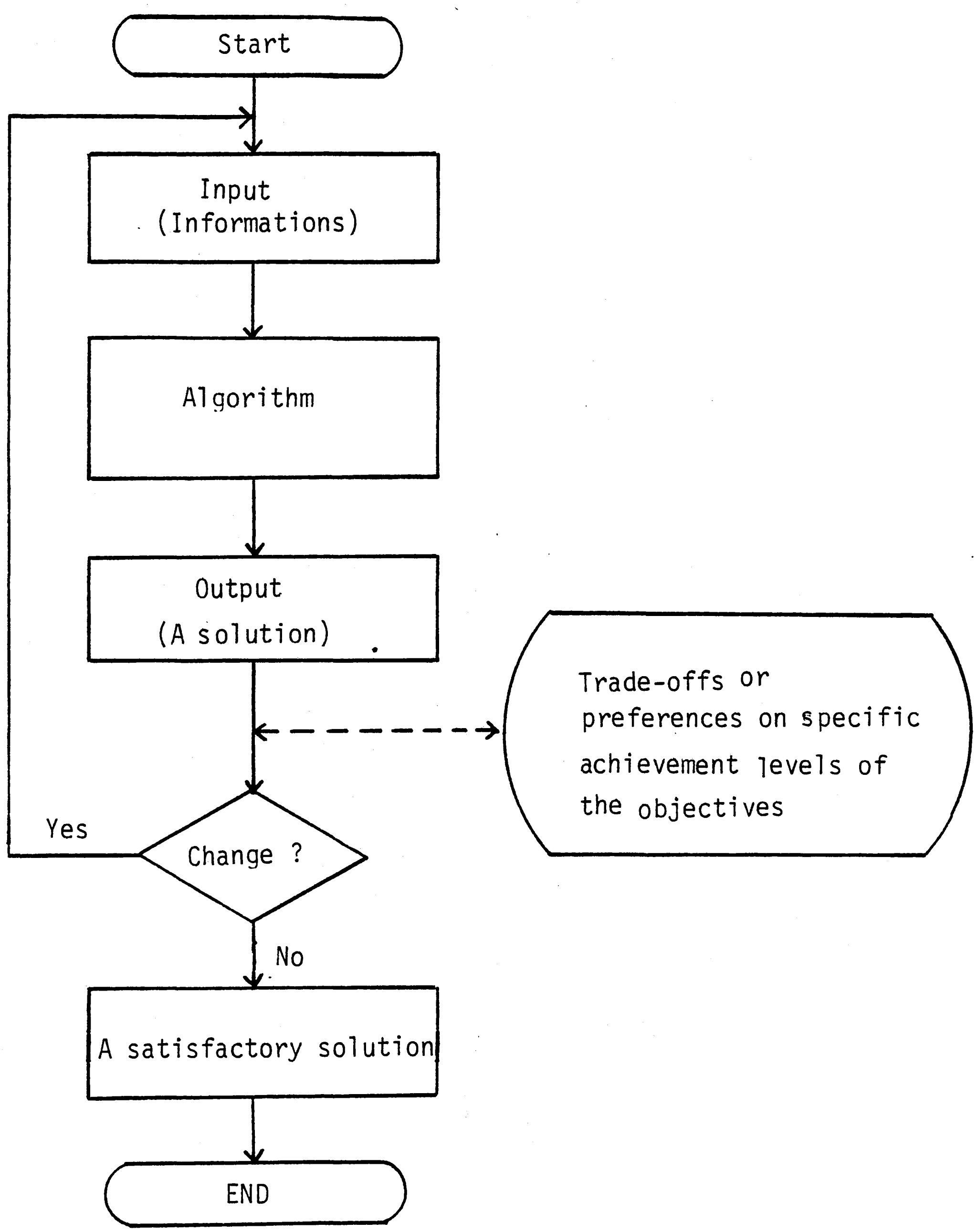

Figure 6. The Logic Flow Chart for an Interactive Procedure

only provides his preferences, but also gains a greater understanding and feeling for the structure of the system.

Hwang and Masud [30] summarize the advantages and disadvantages of the interactive methods. The advantages of the methods are listed as follows:

1. There is no need for "a priori" preference information and only progressive local preference information is required.
2. It is a learning process for the DM to understand the behavior of the system.
3. Since the DM is part of the solution, the solution obtained has a better prospect of being implemented.

On the other hand, the disadvantages are listed as follows:

1. Solutions depend on the accuracy of the local preference the DM can indicate.
2. For some methods, there is no guarantee that the preferred solution can be obtained within a finite number of interactive cycles and the procedure may be time-consuming.
3. Much effort is required of the DM.

Summary

Multiple objective optimization analysis is introduced. In particular, the nondominated solutions set, Goal Programming, and interactive methods for multiple objective decision making are discussed. It is emphasized that the multiple objective formulation of the problems in systems analysis provide a more realistic modeling approach and afford the DM in charge the ability to make intelligent trade-off decisions about the different objectives.

In the next chapter, a development of an algorithm for multicriteria VRPs is presented.

CHAPTER IV

ALGORITHM FOR MULTICRITERIA VEHICLE ROUTING PROBLEMS

Introduction

This chapter presents a heuristic algorithm to determine the most satisfactory vehicle routes for the multiple-vehicle, single-depot case where the conflicting multiple objectives are treated explicitly. The algorithm is illustrated by a simple example.

The version of the VRP examined in this research is concerned with the multiple-vehicle, single-depot case with multiple objectives to be achieved where stations at known locations are scattered around a single depot, each with a known quantity to be collected by multiple vehicles. Each vehicle must be assigned a route beginning at the depot, visiting a number of stations in a prescribed sequence and ending at the depot, with the guarantee that the total collection service on a route does not exceed the vehicle capacity and duration limit. The vehicle duration limit is determined by the smaller value of the maximum allowable vehicle travel distance and the transportation duration until complete goods deterioration.

The objective is to assign at least one route to each vehicle so that each station is collected by exactly one vehicle and three goals, such as the minimization of total travel distance, the minimization of total deterioration of goods during transportation and the maximization of the fulfillment of emergent services and conditional dependencies of stations are achieved. These three goals represent multiple objectives in different dimensions. Furthermore, these objectives are often conflicting, because improvement in one objective can only be made to the detriment of one or all of the rest of the objectives. To analyze these conflicting values and objectives, a technique capable of handling multiple criteria VRPs was developed.

To develop an algorithm for VRPs in a multiple objective environment, the prospect of stations scattered around a central depot has to be carefully examined. Figure 1 shows an example of a layout. Due to the complexity inherent in the problem to solve, that mainly depends on the number of stations in the prospect, a set of stations needs to be partitioned into smaller subsets without losing sight of the overall view of the problem; thus enabling the application of a multiple objective decision making technique to each smaller subset. This logic of the Cluster First approach is further supported by an indication that it is superior to the Route First approach with respect to the optimization of a single objective.

The algorithm developed consists of two major stages:

1. A clustering stage to partition a set of stations into subsets by the "Cluster Method," thus each subset ultimately comprises the stations for a single route. This process is carried out by using information about the spatial layout of the stations, e.g., polar coordinates with the depot as the origin.
2. A routing stage is required to sequence the stations on each route, by applying the "iterative Goal Programming Procedure."

The algorithm yields an optimum or near-optimum solution to multi-criteria VRPs.

Notation

The following terms and definitions were employed in developing the algorithm:

M = total number of stations to be served, excluding a central depot.

N = the number of stations in a route, excluding a central depot.

S = the set of stations in a route, including a central depot.

d_{ij} = the shortest distance between stations i and j.

Q = the vehicle capacity.

MT = the maximum allowable travel distance of vehicles (this is usually a legal or a contractual condition).

T = the upper bound for the constraint on vehicle travel distance.

q_i = the amount of supply at station i.

PL = the predetermined level of transportation duration for the starting point of goods deterioration.

UL = the upper limit of transportation duration until the complete goods deterioration ($PL < UL$).

$(X(i), Y(i))$ = the rectangular coordinates of station i.

$An(i)$ = the polar coordinate angle of station i defined as

$$An(i) = \arctan[(Y(i)-Y(0))/(X(i)-X(0))]$$

where $-\pi \leq An(i) < 0$ if $Y(i)-Y(0) < 0$,

$0 \leq An(i) \leq \pi$ if $Y(i)-Y(0) \geq 0$, and

the central depot is denoted as station 0.

R(i) = the distance (radius) from depot to station i.

TVTT = the target value of a vehicle travel distance.

TVTD = the target value of the transportation duration for goods deterioration.

TT = a vehicle travel distance on a route. (GTT = the grand total distance on the routes.)

TD = a total degree of deterioration generated on a route. (GTD = the grand total deterioration on the routes.)

FR = a total fulfillment of emergent services and conditional dependencies of stations on a route. (GFR = the grand total fulfillment of service requirements on the routes.)

OBTT = an objective: the minimization of total travel distance of vehicles.

OBTD = an objective: the minimization of total deterioration of goods during transportation.

OBFR = an objective: the maximization of total fulfillment of emergent services and conditional dependencies of stations.

SUM(i) = the tentative vehicle travel distance when station i is assigned to the link in the clustering procedure.

TOT(i) = the tentative vehicle load when station i is assigned to the link in the clustering procedure.

$n_{(i)}$ = a set of negative deviations adhered to constraints (i).

$p_{(i)}$ = a set of positive deviations adhered to constraints (i).

Assumptions

The following assumptions were made:

1. The commodity that is to be collected is homogeneous.
2. There exist the known constraints on the capacity of vehicles and the duration of a route.
3. The type of vehicles is homogeneous.
4. The rectangular coordinates of stations are known.
5. The shortest distances between stations are defined as Euclidean distances.
6. Quantities of supply at stations are known and approximately equal.
7. Quantities of supply at stations do not exceed the capacity of vehicles.
8. The degree of deterioration is proportional to an excessive transportation duration over the predetermined level for goods deterioration, after the commodity is loaded into a vehicle at a station. Hence, the total degree of deterioration on a route, TD, is defined by

$$TD = \sum_{\substack{i \in S \\ i \neq 0}} \max \left\{ (RTD_i - PL),\ 0 \right\}$$

   where RTD_i is the remaining transportation duration of the commodity loaded at station i to a depot.
9. There is a known upper limit of transportation duration for the commodity collected until its complete deterioration. Hence, the predetermined level of deterioration may be considered as a starting point of goods deterioration.

The above assumptions are consistent with the problem statement previously given.

Cluster Method

The technique to be presented is based on the heuristic ideas of Gillet and Miller's [24], Clarke and Wright's [17], and William's [61] algorithms that could be used in attaining visual solutions. That is, the method is based on joining stations furthest from the depot to the closest feasible stations within the immediate proximity. The final solution of clustering would be a set of routes. Each route maintains feasibility with regard to the vehicle capacity and duration limit.

The method implies different upper bounds for the constraint on the vehicle travel distance, according to the preemptive goal priority structure. When the first priority is given to the minimization of total travel distance, the smaller value of the maximum allowable vehicle travel distance, MT, and the transportation duration until the complete deterioration of goods, UL, is used as the basis of the upper bound. The transportation duration to the depot on a route should not exceed UL because the goods collected are completely spoiled and become worthless beyond UL. The condition that travel distance on a route minus minimum distance from the depot to any station in the subset does not exceed UL, that is,

$$TT - \min_{\substack{i \in S \\ i \neq 0}} \left\{ d_{0i} \right\} < UL$$

guarantees no complete deterioration of goods during transportation.

When the first priority is placed on the minimization of the total deterioration of goods, the condition that travel distance on a route, minus minimum distance from the depot to any station in the subset, does not exceed the target value of the transportation duration for goods deterioration, TVTD, that is,

$$TT - \min_{\substack{i \in S \\ i \neq 0}} \left\{ d_{0i} \right\} < TVTD$$

is employed to guarantee that no deterioration is caused during transportation. TVTD is usually set equal to PL. However, it may be relaxed to a certain degree, depending upon the DM's preference.

On the other hand, when the first priority is placed on the maximization of the fulfillment of emergent services and conditional dependencies of stations, the procedure should take into account the fact that the stations requiring urgent services are separated into different subsets and the conditionally dependent stations are placed in the same subset. In this study, the goal priority structure with the fulfillment of requirements as the first priority was not treated, because its consideration may result in very poor achievement of the rest of the goals. However, this type of goal priority structure can be employed depending upon the DM's preference. In this research, three models with different goal priority structures were considered in order to demonstrate the flexibility of the proposed algorithm in dealing with unique situations in multicriteria VRPs. Table III presents the descriptive summary of each model's objectives and their preemptive priorities.

TABLE III

PRIORITY STRUCTURES OF THREE ALTERNATIVE MODELS IN THE RESEARCH

Objectives	Model I	Model II	Model III
Minimize total travel distance	P_1	P_2	P_1
Minimize total deterioration of goods during transportation	P_2	P_1	P_3
Maximize the fulfillment of emergent services and conditional dependencies of stations	P_3	P_3	P_2

The clustering procedure starts with an unassigned station at an extreme point in the area in order to form the beginning of a feasible link. A feasible link is a route of one or more stations which does not violate any restrictions, and the link has two ends to which stations can be assigned. Two ends represent two stations newly assigned to the link and connected temporarily to the depot. At the beginning of the feasible link, only the end that is the furthest station from the depot exists.

In the clustering procedure, each of the ends of the link pseudo-assigns (temporarily assigns) the closest two feasible stations within the immediate proximity. This involves the concept of William's Proximity Priority Searching algorithm [61]. A station under competition from two different ends is pseudo-assigned to the closer end. The losing end pseudo-assigns the next closest feasible station. Then, among pseudo-assigned station(s), a station to be assigned to the link is obtained by maximizing a function of the radius R(i) and minimizing the angular difference between the end and its station. This provides a station that is far from the depot and also close to an end of the link in terms of both distance and polar coordinate angle. The remaining pseudo-assigned station(s) are released from their ends.

Based on the above idea that is mainly due to the concepts of the Clarke and Wright method [17] and the Gillet and Miller's SWEEP algorithm [24], a function was developed. The function is:

$$CRT(i) = R(i) + \frac{\bar{d}}{|An(i) - An(j)| * \alpha}$$

where

$\bar{d}$ = the average of the radii of all stations

j = the end to which station i is pseudo-assigned

α = a shape parameter.

Maximizing the function provides a station to be added to a feasible link. In the function CRT(i), the shape parameter α represents a weighting factor to an angular difference between an end and its station. When α is close to zero, a great emphasis is placed on the polar coordinate angle of station. This involves the basic concept of the SWEEP algorithm. On the other hand, when α is large, a great emphasis is given to the distance from a depot to a station. This involves the concept of the Clarke and Wright method. Thus, these two factors can be traded off in the clustering procedure by simply altering α.

The travel distance of the link, for the purpose of the feasibility test, is determined by computing the distance increase when a station is assigned to the link. Let this tentative travel distance of the link be SUM. Then,

$$\text{new SUM} = \text{old SUM} + (d_{ji} + d_{i0} - d_{j0})$$

where j is the end to which station i is to be assigned.

The flow chart shown in Figure 7 outlines the procedural steps for the method developed for clustering a set of stations in multicriteria VRPs and these steps can be summarized as follows:

Step 1:

1) Evaluate the polar coordinates for stations with the depot.
2) Construct the symmetrical distance matrix which gives the distance of stations from one another.
3. Compute the polar coordinate angles of stations, An(i).
4. List all stations in descending distance from the depot.
5. Determine the DM's goal priority structure.

Step 2: Determine the basis of the upper bound for the constraint on vehicle travel distance, T, based on the DM's preference on the goal priority structure.

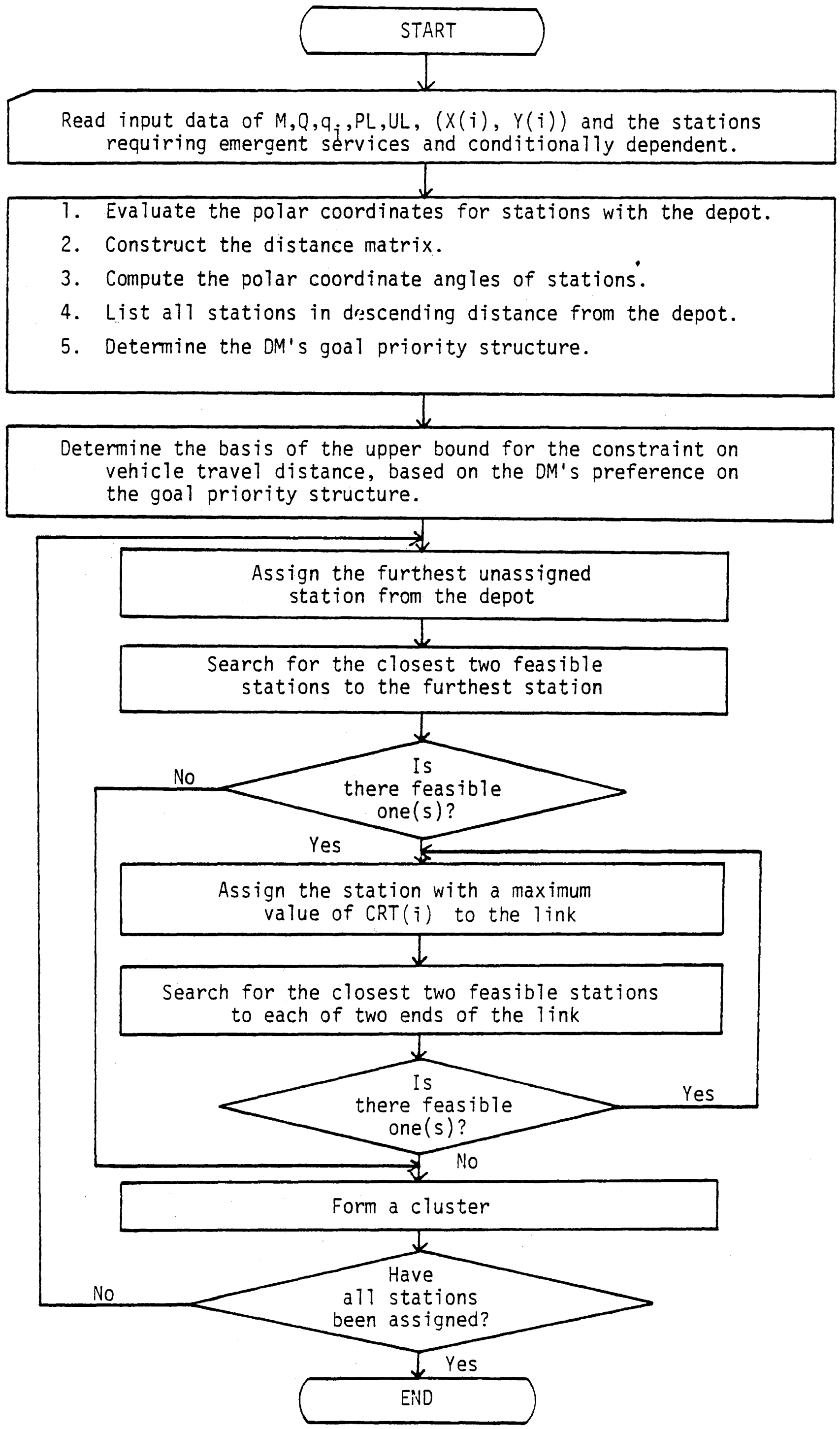

Figure 7. The Logic Flow Chart of the Cluster Method for Multicriteria VRPs

1. If the first priority is placed on the minimization of total travel distance,

$$T = MT \qquad \text{if } MT \le UL + \min_{\substack{i \in S \\ i \ne 0}} \{d_{0i}\}$$

$$T = UL + \min_{\substack{i \in S \\ i \ne 0}} \{d_{0i}\} \qquad \text{if } MT > UL + \min_{\substack{i \in S \\ i \ne 0}} \{d_{0i}\}.$$

2. If the first priority is placed on the minimization of total deterioration of goods,

$$T = TVTD + \min_{\substack{i \in S \\ i \ne 0}} \{d_{0i}\} .$$

TVTD is set equal to PL. It is noted that TVTD may be relaxed to a certain degree by DM.

Step 3: Assign the furthest unassigned station from the depot to form the beginning of the feasible link. A feasible link is a route of one or more stations which does not exceed any restrictions, such as distance and capacity.

Step 4: From the distance matrix, pseudo-assign the closest two feasible stations to the furthest station.

1. If no feasible station exists, go to Step 6.
2. Otherwise, compute CRT(i) for the station(s) and assign the station with a maximum value of CRT(i) to the link. The link now has two ends to which stations can be assigned.

Step 5: Pseudo-assign the closest two feasible stations to each of two ends of the link. A station under competition from two ends is pseudo-assigned to the closer end. The losing end pseudo-assigns the next closest feasible station.

1. If no feasible station exists, go to Step 6.
2. Otherwise, compute CRT(i) for the station(s) and assign the station with a maximum value of CRT(i) to the link. Repeat Step 5.

Step 6: Form a cluster. The completed subset is part of the final solution in the clustering stage and need not be considered during further clustering procedures.

Step 7: Go to Step 3 for continuation, until all stations have been assigned. The solution is the set of created subsets.

A number of comments can be made in order to clarify or justify each of the above procedural steps.

1. The algorithm takes into account the DM's goal priority structure.
2. It is reasonable, intuitively, to start with stations at extreme points in the area in order to avoid single long journeys and to minimize total distance as stations are added to the link.
3. A great emphasis is primarily placed on the distance between an end of the link and a station, rather than position relative to the depot in selecting an addition to the link. Assigning the closest feasible station to the end would generally minimize the distance traveled to service the station.

4. Assigning the station with a maximum CRT(i) to the link has two useful properties:
   (i) A station among pseudo-assigned station(s) is assigned to its end, bringing about a very good saving in terms of travel distance. This involves similar techniques to those used in the "savings" algorithms.
   (ii) The completed subsets are forced to follow a "petal" shape that rarely crosses adjacent subsets.
5. To determine the station to be assigned to the link, only the closest two feasible stations are searched at each of the ends as the candidates. Hence, the effort for sorting the distance matrix is significantly reduced, without the need to create any precomputed file or matrix such as the "savings" file in savings methods.
6. The method does not require the routing procedure. Therefore, the computation burden is very low.

Iterative Goal Programming Heuristic Procedure

<u>Initial Development of An Exact GP Model</u>

Once a set of stations are clustered into subsets in the first stage, the second stage of the algorithm sequences the stations in each subset by applying the GP approach to each cluster. The reasons for utilizing the GP approach in addressing multicriteria VRPs are:

1. It allows the optimization of the desired goal attainments while permitting an explicit consideration of the multiple conflicting objectives.
2. It is useful in providing the capability of evaluating different strategies under various assumed goal levels and/or varying the DM's policies about the goal priority structures.
3. It is expected to require a sizeable effort to search for all of the nondominated solutions.

The development of a GP model requires a sequence of several steps [55].

1. Determination of model objectives and their priorities.
2. Identification of the decision variables.
3. Formulation of model constraints.
4. Analysis of the model solution and its implications.

The first three items are discussed in detail.

Model Objectives and Their Priorities.

The multicriteria VRP involves multiple objectives and implications. Their importance and priority may vary according to the conditions under consideration. In the research, three different GP models were developed. Table III presents a descriptive summary of each model's objectives and their preemptive priorities. The objectives are:

1. Minimize total travel distance of vehicles (OBTT).
2. Minimize total degree of deterioration of goods during transportation (OBTD).

3. Maximize the fulfillment of emergent services and conditional dependencies of stations (OBFR).

These three goals represent multiple objectives in different dimensions. Furthermore, they are often in conflict.

Decision Variables.

The primary objective of the multicriteria VRP is to determine route sequences that should be followed by vehicles in order to service the customers. The decision variable $x_{ij} = 1$ if the vehicle visits station j immediately after visiting station i, and $x_{ij} = 0$ otherwise.

Model Constraints.

The GP model usually has two types of constraints, system and goal constraints. The former represent a set of fact-of-life type constraints which must be adhered to before an optimal solution can be considered. The latter represent a set of constraints which include the objectives of the problem. The following constraints are to be considered:

1. Only one station must immediately follow station i in a given route. The system constraints are:

$$\sum_{\substack{j \varepsilon S \\ j \neq i}} x_{ij} + n - p = 1, \qquad \text{for } i \varepsilon S.$$

These constraints can be achieved by minimizing both negative (n) and positive (p) deviations for each station i.

2. Only one station must immediately precede station j in a given route. The system constraints are:

$$\sum_{\substack{i \in S \\ i \neq j}} x_{ij} + n - p = 1, \qquad \text{for } j \in S.$$

These constraints can be achieved by minimizing both n and p for each station j.

3. A constraint must be imposed to ensure that a selection of x_{ij} actually represents a feasible, complete route without subtours. To accomplish this task, N additional variables, u_i, are defined. The desired results can be achieved by minimizing $p_{(3)}$ from the system constraints:

$$u_i - u_j + (N+1)\, x_{ij} + n - p = N, \text{ for } i,j \in S,\ i \neq j, \text{ and } i,j \neq 0$$

where u_i, i=1,2,...,N, are arbitrary real numbers.

4. A primary objective of the VRP is the minimization of the total distance traveled by vehicles. The total travel distance must be kept within a reasonable bound, i.e., target value, with the consideration of the legal or contractual condition and/or goods deterioration. This goal constraint can be expressed by :

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} d_{ij}\, x_{ij} + n - p = TVTT$$

where n represents the amount of duration shortened below bound, TVTT. The minimization of total travel distance can be achieved by assuming the bound as zero and minimizing p.

5. An important consideration in some VRPs is the minimization of total deterioration of goods during transportation. Based on the definition given in assumption (8), the degree of deterioration of the goods collected at the kth stop in a route sequence is determined by computing an excessive transportation duration from the kth visited station to the central depot over the predetermined starting point for deterioration PL. Thus, the minimization of the degree of deterioration of the goods loaded at the kth stop can be accomplished by minimizing the remaining transportation duration to the depot. A faster transportation of goods than the predetermined starting point for deterioration does not give any value in view of the deterioration minimization. The goal constraints are now formulated for each stop with the objective of minimizing $p_{(5)}$. TVTD is set equal to PL. However, it may be relaxed to a certain degree, depending upon the DM's preference.

$$\sum_{i \varepsilon S} \sum_{\substack{j \varepsilon S \\ j \neq i}} d_{ij} x_{ij} - \sum_{\substack{j \varepsilon S \\ j \neq 0}} d_{0j} x_{0j} + n - p = \text{TVTD, for the 1st stop}$$

$$\sum_{i \varepsilon S} \sum_{\substack{j \varepsilon S \\ j \neq i}} d_{ij} x_{ij} - \sum_{\substack{j \varepsilon S \\ j \neq 0}} \sum_{\substack{k \varepsilon S \\ k \neq j \\ k \neq 0}} (d_{0j} + d_{jk})(x_{0j} x_{jk}) + n - p = \text{TVTD},$$

for the 2nd stop

$$\sum_{i \varepsilon S} \sum_{\substack{j \varepsilon S \\ j \neq i}} d_{ij} x_{ij} - \overbrace{\sum_{\substack{j \varepsilon S \\ j \neq 0}} \sum_{\substack{k \varepsilon S \\ k \neq j \\ k \neq 0}} \cdots \sum_{q} \sum_{\substack{r \varepsilon S \\ r \neq q \neq \ldots \neq k \neq j \\ r \neq 0}}}^{w} (d_{0j} + d_{jk} + \ldots + d_{qr})$$

$$(x_{0j}\, x_{jk} \cdots x_{qr}) + n - p = TVTD, \qquad \text{for the wth stop}$$

$$\sum_{\substack{i \varepsilon S \\ i \neq 0}} d_{i0}\, x_{i0} + n - p = TVTD, \qquad \text{for the last stop}$$

where n denotes a faster delivery of goods than TVTD and p represents the degree of goods deterioration.

6. Another important consideration is the treatment of emergent stations that should be serviced with the first stop, and conditional dependencies of stations. The degree of fulfillment of these requirements can be determined by the number of the requirements to be satisfied in a solution. If station m requests an urgent service and station n is conditionally dependent on station m, the goal constraints are:

$$x_{0m} + n - p = 1$$
$$x_{mn} + n - p = 1.$$

These goal constraints can be achieved by minimizing both $n_{(6)}$ and $p_{(6)}$.

7. Since the decision variables require 0 or 1 integer values, the system constraints for integrality have to be provided. This is accomplished by minimizing $p_{(7)}$ from the system constraints

$$x_{ij} + n - p = 1, \qquad \text{for } i,\ j \varepsilon S \text{ and } i \neq j.$$

However, these constraints may not be expressed explicitly in the GP model when a computer code for integer programming is employed as the solution method, because constraints (1) and (2) restrict the decision variables to 0 or 1. Therefore, these system constraints will not be further considered in the model.

The Achievement Function.

The achievement function of the GP model includes minimizing deviations, either negative or positive, or both, from a set of goals, with certain preemptive priority weights P_j assigned by the DM. However, a primal priority should be given to the first three system constraints, because those are the basic constraints for defining the VRP before an optimal solution can be considered in the model. The remaining three goal constraints may be assigned certain preemptive priorities by the DM. Table IV presents the goal priority structures of three alternative GP models. The achievement functions for the three models are formulated as follows:

For Model I,

$$\text{min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [p_{(4)}] + P_3 [p_{(5)}] + P_4 [n_{(6)} + p_{(6)}].$$

For Model II,

$$\text{min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [p_{(5)}] + P_3 [p_{(4)}] + P_4 [n_{(6)} + p_{(6)}].$$

For Model III,

$$\min.\ P_1\ [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2\ [p_{(4)}] + P_3\ [n_{(6)} + p_{(6)}] + P_4\ [p_{(5)}].$$

TABLE IV

PRIORITY STRUCTURES OF THREE ALTERNATIVE GP MODELS

Goals	Model I	Model II	Model III
System constraints (1) - (3)	P_1	P_1	P_1
OBTT	P_2	P_3	P_2
OBTD	P_3	P_2	P_4
OBFR	P_4	P_4	P_3

Heuristic Procedure

The GP formulation for an exact solution as it stands has a serious computational difficulty in its application, due to constraint (5). That is, the GP model is a nonlinear integer GP for which no efficient and practical solution procedure has been developed. Though a nonlinear integer GP may be at least theoretically solved by transforming it into a linear integer GP, its size increases rather dramatically and quickly gets out of hand [33]. Furthermore, for constraint (5), the number of possible partial routes to be enumerated are greatly increased as the number of stations are increased, which causes a tremendous effort in formulating the constraints.

To overcome such problems, this author has developed an iterative procedure with linear integer GP applications, called the "Iterative GP Heuristic Procedure." This heuristic procedure is based on the following theoretical considerations of the deterioration definition:

1. The remaining transportation duration to the depot is decreased as the vehicle visits more stations. In other words, the commodity collected at the earlier visit would result in a higher degree of deterioration, if deterioration exists, than one collected later.
2. A route that gives the minimal deterioration of the commodity collected at the 1st station in the sequence tends to result in the minimal total deterioration, among all feasible alternatives.
3. The computation of the remaining transportation duration of the commodity from a certain station requires that the station(s) already stopped be known.

At each iteration in the algorithm, the next station to stop is determined by solving a linear integer GP model that is constructed on the basis of the known sequence of the stations determined at the previous iterations, instead of generating a complete route sequence at a time as in the exact GP method. Since the linear integer GP model is used to determine the station that should follow the current station immediately, constraint (5) in the model consists of only one linear 0-1 integer GP constraint. Consequently, the GP model is practically solvable without the tremendous effort of constraints formulation otherwise required.

The procedure is repeated until a complete route sequence is obtained. However, the number of iterations may be significantly shortened by employing another stopping rule:

> The procedure may be terminated when a station, at which the commodity collected is delivered to the depot without deterioration, is first found. In other words, there would be no deterioration generated by the commodity to be collected at the next station to stop, determined by solving the current GP model.

The complete route sequence that is obtained at the last iteration is considered as the most satisfactory solution to be employed. At this time, it cannot be guaranteed that this iterative GP heuristic procedure always generates an optimal solution in multicriteria VRPs. However, the solution obtained would be a good one. The logic flow chart of this heuristic is shown in Figure 8.

Let [k] be the kth station to stop in a route and [0] be equal to a central depot 0. The steps of the procedure can be stated as follows:

Step 1: Let $k = 0$ and $Q = \sum_{i \varepsilon S} \sum_{\substack{j \varepsilon S \\ j \neq i}} d_{ij} x_{ij}$.

Step 2: Solve the following GP model with the achievement function based on the DM's preference on the goal priority structure:

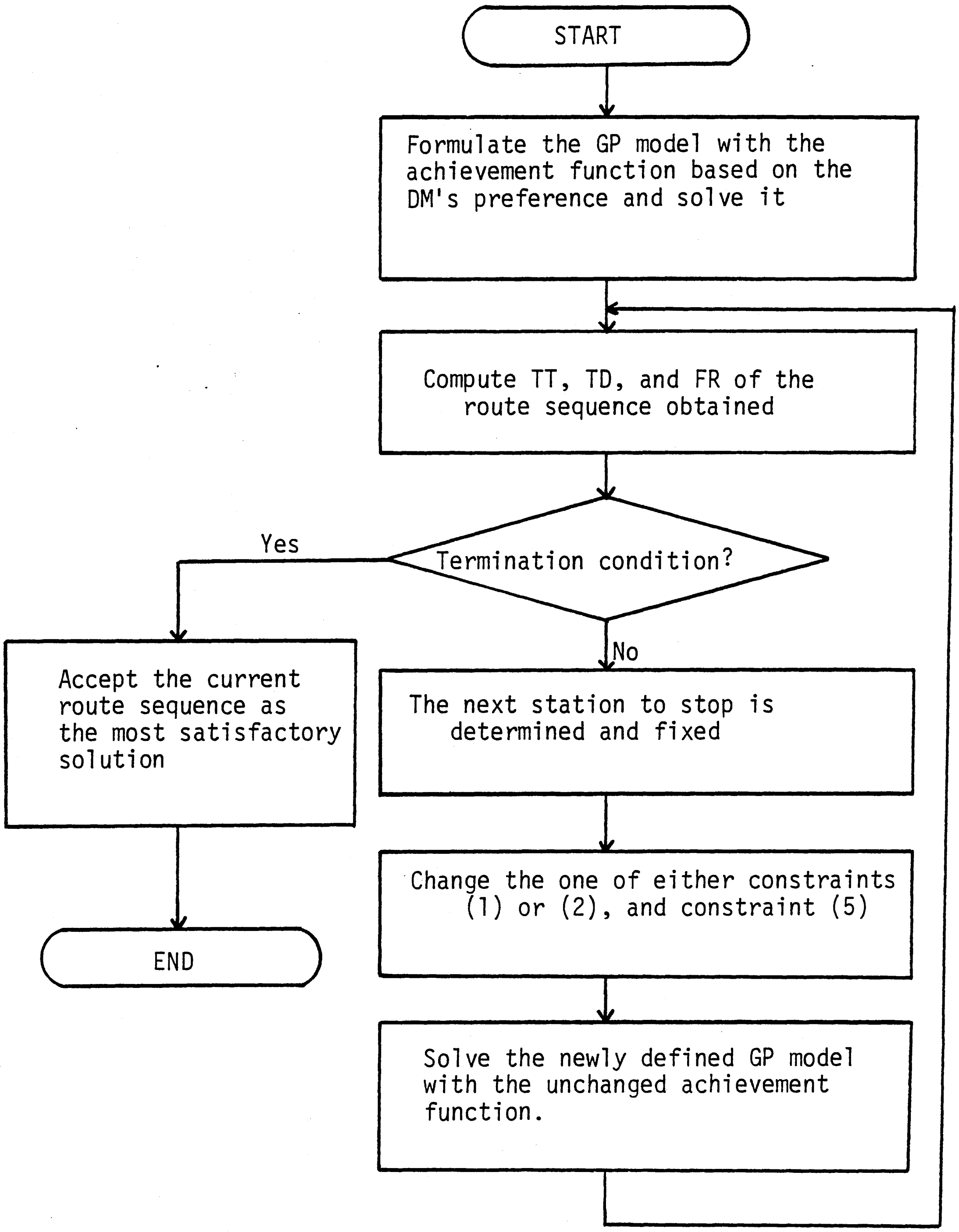

Figure 8. The Logic Flow Chart of the Iterative GP Procedure for Multicriteria VRPs.

$$\text{Min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [p_{(4)}] + P_3 [p_{(5)}] + P_4 [n_{(6)} + p_{(6)}] \quad \text{for Model I.}$$

$$\text{Min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [p_{(5)}] + P_3 [p_{(4)}] + P_4 [n_{(6)} + p_{(6)}] \quad \text{for Model II.}$$

$$\text{Min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [P_{(4)}] + P_3 [n_{(6)} + p_{(6)}] + P_4 [p_{(5)}] \quad \text{for Model III.}$$

subject to

$$\sum_{\substack{j \in S \\ j \neq i}} x_{ij} + n_{(1)} - p_{(1)} = 1, \qquad \text{for } i \in S \qquad (1)$$

$$\sum_{\substack{i \in S \\ i \neq j}} x_{ij} + n_{(2)} - p_{(2)} = 1, \qquad \text{for } j \in S \qquad (2)$$

$$u_i - u_j + (N+1)x_{ij} + n_{(3)} - p_{(3)} = N, \qquad \text{for } i,j \in S,\ i \neq j \text{ and } i,\ j \neq 0 \qquad (3)$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} d_{ij}\, x_{ij} + n_{(4)} - p_{(4)} = TVTT \qquad (4)$$

$$Q - \sum_{\substack{j \in S \\ j \neq [k]}} d_{[k]j}\, x_{[k]j} + n_{(5)} - p_{(5)} = TVTD \qquad (5)$$

$$x_{0m} + n_{(6)} - p_{(6)} = 1$$
$$x_{mn} + n_{(6)} - p_{(6)} = 1 \qquad (6)$$

Step 3: new Q = old Q - $\sum_{\substack{j \varepsilon S \\ j \neq [k]}} d_{[k]j} \, x_{[k]j}$

Step 4: Compute TT, TD, and FR of the route sequence obtained in step 2. Let k = k + 1.

Step 5: If either $p_{(5)} = 0$ or k = N-1, then accept the current route sequence as the most satisfactory solution and stop. Otherwise, 1) [k] is determined and

2) let $x_{[k-1][k]} = 1$.

Step 6: Change one of either constraints (1) or (2) according to the following principle; $x_{[k-1][k]}$ must be forced to be one, thus the achievement function should minimize both n and p from the corresponding constraint. Solve the newly defined GP model with the unchanged DM's preference on the goal priority structure and go to Step 3.

In applying the Iterative GP Heuristic Procedure to each subset formed by the Cluster Method, a total of $N^2 + N + 6$ model constraints with a total of $N^2 + 2N$ decision variables should be formulated at each iteration. However, the effort of the constraints formulation is actually limited only to the first iteration. For the remaining iterations until termination only the very slight changes of two constraints are required. Once the GP model is formulated at each iteration, it can be solved using the computer code for integer GP [32].

Example Problem

The algorithm for multicriteria VRPs, consisting of the Cluster Method and the Iterative GP Heuristic Procedure, is illustrated by a simple example problem. Consider a small problem involving a single depot and six stations to serve by vehicles. In Figure 9 the rectangular coordinates of the stations and depot are expressed on the corresponding node denoted by the number inside each circle, and the net supply quantities are marked on the left side of each node. The following conditions are given:

1. The maximum allowable vehicle travel distance is limited to 190 units.
2. There are 200-unit capacity vehicles available.
3. The goods start to deteriorate after 115 distance units and are completely spoiled at 200 distance units.
4. The stations requiring emergent services are station 2, 5, and 6.
5. The stations that are conditionally dependent are stations 2 and 3, and stations 3 and 5.
6. For each stop, 10 distance units allowance is required for the operation.
7. The DM's goal priority structure follows Model I from Table III.

If all the assumptions being employed in this research are also applied to the example problem, then the problem can be solved by applying the proposed algorithm in order to determine the most satisfactory solution with respect to the DM's preference. The target value of the transportation duration for goods deterioration is set equal to the predetermined starting point for goods deterioration. The solution procedure is described step by step.

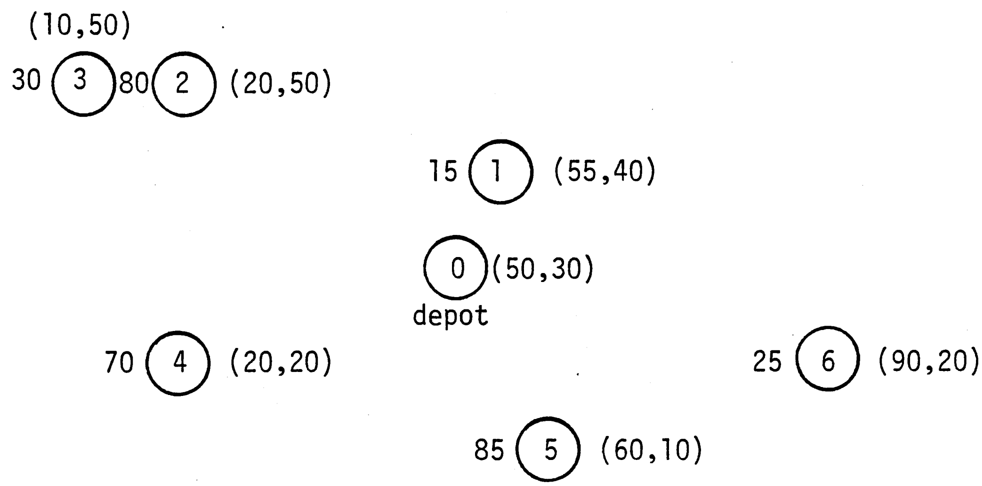

Figure 9. Graphical Configuration of a Depot and Stations in Example Problem

Clustering Stage

The set of stations are clustered into subsets by applying the Cluster Method.

1. Construct the distance matrix given in Table V.
2. Compute the polar coordinate angles of all stations as follows:

   An(1) = 1.11, An(2) = -0.59, An(3) = -0.46, An(4) = -0.32, An(5) = -1.11, and An(6) = -0.25.
3. Determine the basis of the upper bound for the constraint on vehicle travel distance, T.

   T = 190 because the first priority is placed on the minimization of vehicle travel distance and MT < UL.

$(T = 115 + \min_{\substack{i \in S \\ i \neq 0}} \{d_{0i}\}$ if the first priority is given to OBTD.)

TABLE V

DISTANCE MATRIX OF EXAMPLE PROBLEM

	0	1	2	3	4	5	6
0	-	11	36	44	31	22	41
1	11	-	36	46	40	30	40
2	36	36	-	10	30	56	76
3	44	46	10	-	31	64	85
4	31	40	30	31	-	41	70
5	22	30	56	64	41	-	31
6	41	40	76	85	70	31	-

4. Assign the furthest station from the depot, station 3. So the first link starts with $\{3\}$.
5. Select the closest two stations to station 3, and perform a feasibility test with them as follows:

   SUM(2) = 44 + 10 + 10 + 10 + 36 = 110 < 190

   TOT(2) = 30 + 80 = 110 < 200

   SUM(4) = 44 + 10 + 31 + 10 + 31 = 126 < 190

   TOT(4) = 30 + 70 = 100 < 200
6. Pseudo-assign stations 2 and 4 to station 3.
7. Compute CRT(i) for the two stations as follows

(α is assumed to be 2.0):

$$CRT(2) = 36 + \frac{30.8}{|-0.59 + 0.46| * 2.0} = 154.5$$

$$CRT(4) = 31 + \frac{30.8}{|-0.32 + 0.46| * 2.0} = 141.0$$

Assign station 2 to the link since CRT(2) > CRT(4). New link is $\{3,2\}$. The remaining pseudo-assigned station 4 is released from its end, station 3.

8. Select the closest two stations to stations 3 and 2, each, and perform a feasibility test with them as follows:

For station 3,

SUM(5) = 110 - 44 + 64 + 10 + 22 = 162 < 190

TOT(5) = 110 + 85 = 195 < 200

SUM(6) = 110 - 44 + 85 + 10 + 41 = 202 > 190 -- infeasible

TOT(6) = 110 + 25 = 135 < 200.

For station 2,

SUM(1) = 110 - 36 + 36 + 10 + 11 = 131 < 190

TOT(1) = 110 + 15 = 125 < 200

SUM(4) = 110 - 36 + 30 + 10 + 31 = 145 < 190

TOT(4) = 110 + 70 = 180 < 200.

9. Pseudo-assign station 5 to station 3, and stations 1 and 4 to station 2.

10. Compute CRT(i) for the three stations as follows:

$$CRT(5) = 22 + \frac{30.8}{|-1.11 + 0.46| * 2.0} = 45.7$$

$$CRT(1) = 11 + \frac{30.8}{|\,1.17 + 0.59\,| * 2.0} = 19.8$$

$$CRT(4) = 31 + \frac{30.8}{|-0.32 + 0.59\,| * 2.0} = 88.0 .$$

Hence, assign station 4 to station 2. New link is $\{3,2,4\}$.
The remaining pseudo-assigned stations are released.

11. Select the closest two stations to stations 3 and 4, each, and perform a feasibility test with them as follows:

For station 3,

SUM(6) = 145 - 44 + 85 + 10 + 41 = 237 > 190 -- infeasible

TOT(6) = 180 + 25 = 205 > 200 -- infeasible

For station 4,

SUM(1) = 145 - 31 + 40 + 10 + 11 = 176 < 190

TOT(1) = 180 + 15 = 195 < 200

SUM(5) = 145 - 31 + 41 + 10 + 22 = 187 < 190

TOT(5) = 180 + 85 = 265 > 200 --infeasible

SUM(6) = 145 - 31 + 70 + 10 + 41 = 235 > 190 --infeasible

TOT(6) = 180 + 25 = 205 > 200 --infeasible

Hence, assign station 1 to station 4. New link is $\{3,2,4,1\}$.

12. Select the closest two stations to stations 3 and 1, each, and perform a feasibility test with them as follows:

For station 3, none.

For station 1,

SUM(5) = 176 - 11 + 30 + 10 + 22 = 227 > 190 --infeasible

$TOT(5) = 195 + 85 = 280 > 200$ --infeasible

$SUM(6) = 176 - 11 + 40 + 10 + 41 = 256 > 190$ --infeasible

$TOT(6) = 195 + 25 = 220 > 200$ --infeasible

13. Since no feasible station exists, form a cluster $\{3,2,4,1\}$. Assign the furthest unassigned station from the depot, station 6, so the second link starts with $\{6\}$.

14. Perform a feasibility test with station 5 as follows:

    $SUM(5) = 41 + 10 + 31 + 10 + 22 = 114 < 190$

    $TOT(5) = 25 + 85 = 110 < 200$

15. Assign station 5 to station 6. Form the second cluster, $\{6,5\}$ and stop. The completed subsets are: $\{3,2,4,1\}$ and $\{6,5\}$.

Routing Stage

The stations in each subset are sequenced by applying the Iterative GP Heuristic Procedure. For convenience, the target value of vehicle travel distance was determined by adding 20 units to the minimal travel distance of a route which can be obtained by solving a Traveling Salesman Problem.

1. Let k=0 and [0] = 0

2. Formulate the GP model for subset 1, $\{3,2,4,1\}$ as follows: (a different achievement function would be employed for the different priority structure):

$$\text{Min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [p_{(4)}] + P_3 [p_{(5)}] + P_4 [n_{(6)} + p_{(6)}]$$

subject to

$$x_{01} + x_{02} + x_{03} + x_{04} + n_1 - p_1 = 1 \qquad (1)$$
$$x_{10} + x_{12} + x_{13} + x_{14} + n_2 - p_2 = 1$$
$$x_{20} + x_{21} + x_{23} + x_{24} + n_3 - p_3 = 1$$
$$x_{30} + x_{31} + x_{32} + x_{34} + n_4 - p_4 = 1$$
$$x_{40} + x_{41} + x_{42} + x_{43} + n_5 - p_5 = 1$$

$$x_{10} + x_{20} + x_{30} + x_{40} + n_6 - p_6 = 1 \qquad (2)$$
$$x_{01} + x_{21} + x_{31} + x_{41} + n_7 - p_7 = 1$$
$$x_{02} + x_{12} + x_{32} + x_{42} + n_8 - p_8 = 1$$
$$x_{03} + x_{13} + x_{23} + x_{43} + n_9 - p_9 = 1$$
$$x_{04} + x_{14} + x_{24} + x_{34} + n_{10} - p_{10} = 1$$

$$u_1 - u_2 + 5x_{12} + n_{11} - p_{11} = 4 \qquad (3)$$
$$u_1 - u_3 + 5x_{13} + n_{12} - p_{12} = 4$$
$$u_1 - u_4 + 5x_{14} + n_{13} - p_{13} = 4$$
$$u_2 - u_1 + 5x_{21} + n_{14} - p_{14} = 4$$
$$u_2 - u_3 + 5x_{23} + n_{15} - p_{15} = 4$$
$$u_2 - u_4 + 5x_{24} + n_{16} - p_{16} = 4$$
$$u_3 - u_1 + 5x_{31} + n_{17} - p_{17} = 4$$
$$u_3 - u_2 + 5x_{32} + n_{18} - p_{18} = 4$$
$$u_3 - u_4 + 5x_{34} + n_{19} - p_{19} = 4$$
$$u_4 - u_1 + 5x_{41} + n_{20} - p_{20} = 4$$
$$u_4 - u_2 + 5x_{42} + n_{21} - p_{21} = 4$$
$$u_4 - u_3 + 5x_{43} + n_{22} - p_{22} = 4$$

$$11x_{01} + 36x_{02} + 44x_{03} + 31x_{04} + 11x_{10} + 36x_{12} + 46x_{13} + 40x_{14}$$
$$+ 36x_{20} + 36x_{21} + 10x_{23} + 30x_{24} + 44x_{30} + 46x_{31} + 10x_{32}$$
$$+ 31x_{34} + 11x_{40} + 40x_{41} + 30x_{42} + 31x_{43} + n_{23} - p_{23} = 179 \quad (4)$$

$$11x_{10} + 36x_{12} + 46x_{13} + 40x_{14} + 36x_{20} + 36x_{21} + 10x_{23} + 30x_{24}$$
$$+ 44x_{30} + 46x_{31} + 10x_{32} + 31x_{34} + 11x_{40} + 40x_{41} + 30x_{42}$$
$$+ 31x_{43} + n_{24} - p_{24} = 115 \quad (5)$$

$$x_{02} + n_{25} - p_{25} = 1 \quad (6)$$
$$x_{23} + n_{26} - p_{26} = 1$$

3. Solve it by using the computer code for integer GP [28]. The solution obtained is the route 0-4-3-2-1-0, where the degree of deterioration of goods collected at the first station to stop is 3 units, i.e., $p_{24} = 3$. Let k = 1.
4. [1] = 4 and let $x_{04} = 1$. Formulate the following new GP Model for the second iteration and solve it:

$$\text{Min. } P_1 [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2 [p_{(4)}]$$
$$+ P_3 [p_{(5)}] + P_4 [n_{(6)} + p_{(6)}]$$

subject to

$$x_{04} + n_1 - p_1 = 1 \quad (1)$$
$$x_{10} + x_{12} + x_{13} + x_{14} + n_2 - p_2 = 1$$
$$x_{20} + x_{21} + x_{23} + x_{24} + n_3 - p_3 = 1$$
$$x_{30} + x_{31} + x_{32} + x_{34} + n_4 - p_4 = 1$$
$$x_{40} + x_{41} + x_{42} + x_{43} + n_5 - p_5 = 1$$

No change (2)

No change (3)

No change (4)

$$11x_{10} + 36x_{12} + 46x_{13} + 40x_{14} + 36x_{20} + 36x_{21} + 10x_{23} + 30x_{24}$$
$$+ 44x_{30} + 46x_{31} + 10x_{32} + 31x_{34} + n_{24} - p_{24} = 115 \qquad (5)$$

No change (6)

5. Since $p_{24} = 0$ for this solution, stop. The most satisfactory solution obtained is therefore the route 0-4-3-2-1-0 whose TT is 159 units, TD is 3 units, and FR is 1.
6. Let k=0 and [0] = 0.
7. Formulate the following GP model for subset 2, $\{6,5\}$, and solve it:

$$\text{Min. } P_1\ [n_{(1)} + p_{(1)} + n_{(2)} + p_{(2)} + p_{(3)}] + P_2\ [p_{(4)}]$$
$$+ P_3\ [p_{(5)}] + P_4\ [n_{(6)} + p_{(6)}]$$

subject to

$$x_{05} + x_{06} + n_1 - p_1 = 1 \qquad (1)$$
$$x_{50} + x_{56} + n_2 - p_2 = 1$$
$$x_{60} + x_{65} + n_3 - p_3 = 1$$

$$x_{50} + x_{60} + n_4 - p_4 = 1 \qquad (2)$$

$$x_{05} + x_{65} + n_5 - p_5 = 1$$

$$x_{06} + x_{56} + n_6 - p_6 - 1$$

$$u_5 - u_6 + 3x_{56} + n_7 - p_7 = 2 \qquad (3)$$

$$u_6 - u_5 + 3x_{65} + n_8 - p_8 = 2$$

$$22x_{05} + 41x_{06} + 22x_{50} + 31x_{56} + 41x_{60} + 31x_{65}$$

$$+ n_9 - p_9 = 134 \qquad (4)$$

$$22x_{50} + 31x_{56} + 41x_{60} + 31x_{65} + n_{10} - p_{10} = 115 \qquad (5)$$

$$x_{06} + n_{11} - p_{11} = 1 \qquad (6)$$

$$x_{05} + n_{12} - p_{12} = 1$$

8. Since $p_{24} = 0$, Stop. The most satisfactory solution obtained is therefore the route 0-5-6-0 whose TT is 114 units, TD is none, and FR is 1.
9. Routing for the two subsets is completed and the procedure for the proposed algorithm is ended.

Table VI shows the results of the example problem, for three Models with different goal priority structures. As would be expected, the outcomes for the Models differ, depending upon the DM's preference regarding the priority structure.

TABLE VI

SUMMARY OF THE OUTCOMES OF
EXAMPLE PROBLEM FOR
THREE MODELS

Model No.	Model I	Model II	Model III
No. of Routes	2	3	2
Routes Sequence	0-4-3-2-1-0 0-5-6-0	0-2-3-0 0-5-6-0 0-4-1-0	0-2-3-4-1-0 0-5-6-0
GTT	273	326	282
GTD	3	0	7
GFR	1	3	3

Summary

A heuristic algorithm is developed to determine the most satisfactory vehicle routes of the multicriteria VRP where three objectives, the minimization of total travel distance, minimization of total deterioration of goods, and maximization of the fulfillment of emergent services and conditional dependencies of stations are to be achieved. The algorithm consists of the Cluster Method to partition a set of stations into subsets and the Iterative GP Procedure to sequence the stations in each subset. A function is proposed in the Cluster Method which is used as the basis for clustering stations to a link. The development of the exact GP model and derivation of the Iterative GP Heuristic from it are discussed. A simple example problem is employed to illustrate the algorithm procedure.

The algorithm developed in this research has the capability of treating the conflicting multiple objectives simultaneously while previously proposed methods for VRPs concern only a single objective. Furthermore, it has the important capability of taking into account the DM's preference regarding the goal priority and the target value of the goal constraints. Therefore, it can provide the DM with the ability to make intelligent trade-off decisions about the different objectives. It is noted that the approach applied in this research could be extended to include any number of possible objectives that would make the model more realistic and adoptable.

In the next chapter, computational experiments and results for the proposed algorithm are presented. Its performance is also evaluated.

CHAPTER V

COMPUTATIONAL RESULTS AND ANALYSIS

Introduction

This chapter presents the computational experience of the algorithm developed in this research. The computational experiments of the proposed algorithm are carried out on three test problems. Its performance is evaluated by comparing the results with those obtained by the existing savings methods, which are for VRPs with a single objective, with respect to the criteria corresponding to the multiple objectives. Three savings methods, Clarke and Wright's savings, multiple and sequential approaches [17], and Gaskell's savings, multiple (λ) approach [23], are selected for the comparision because these methods have been generally considered as representative of the Route First methods and have also proved to be commercially popular.

Programming

Initially, an attempt was made to solve the GP model, using the computer code available for integer GP [32]. However, the code frequently generated an infinite loop in the solution procedure, even for small problems. To overcome this difficulty, this author adopted the SLGP approach with the application of an algorithm for mixed integer programming (MINT algorithm) developed by Kuester and Mize [37], for a solution method.

The MINT algorithm is based on the Land and Doig [37] method. Its FORTRAN program is based on branch and bound mixed integer programming [55], and is available in [37]. Since SLGP decomposes the GP model into an ordered series of single objective mixed integer linear programming optimization problems according to the preemptive priority levels, the MINT algorithm is employed to solve each single objective optimization problem. The logic flow charts of the Iterative GP Heuristic Procedure with an application of the SLGP approach for three Models are shown in Figures 10, 11, and 12. The initial Traveling Salesman Problem in the flow chart of each model is required to provide the DM with the basic information in determining the target value of vehicle travel distance.

The proposed algorithm was coded in FORTRAN. A list of the source program with necessary documentation is included in Appendix A. The program can solve the following sizes of problems:

1. It can cluster an unlimited number of stations.
2. For each subset, it can route a maximum of 10 stations.

The capability of solving larger size multicriteria VRPs can be achieved by increasing the array dimensions in the computer program.

Test Problems

Three test problems are solved by the proposed algorithm. Of the three problems, the data for the first two were proposed by

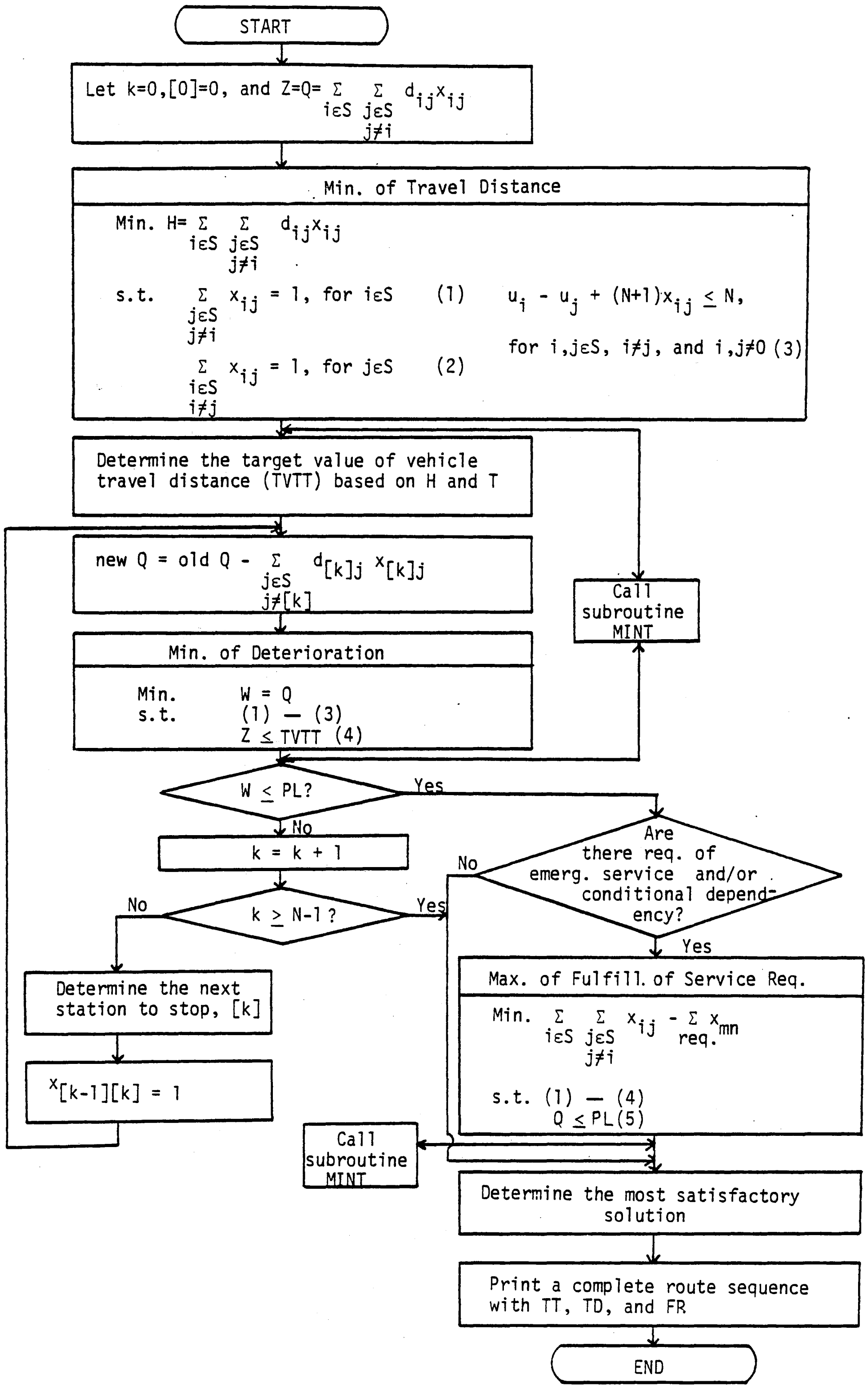

Figure 10. Logic Flow Chart of the Iterative GP Procedure with an Application of SLGP Approach for Model I

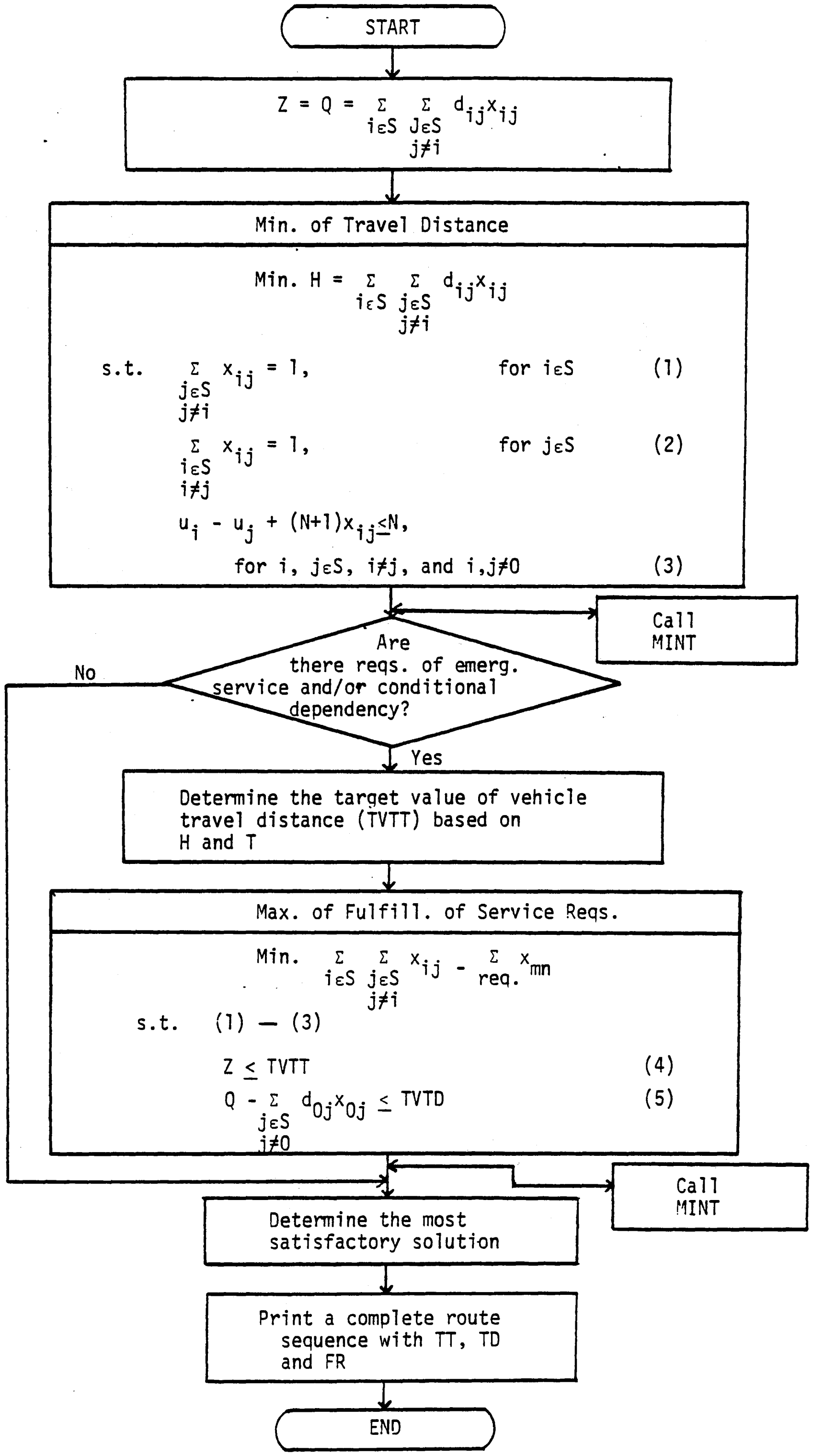

Figure 11. Logic Flow Chart of the Iterative GP Procedure with an Application of SLGP Approach for Model II

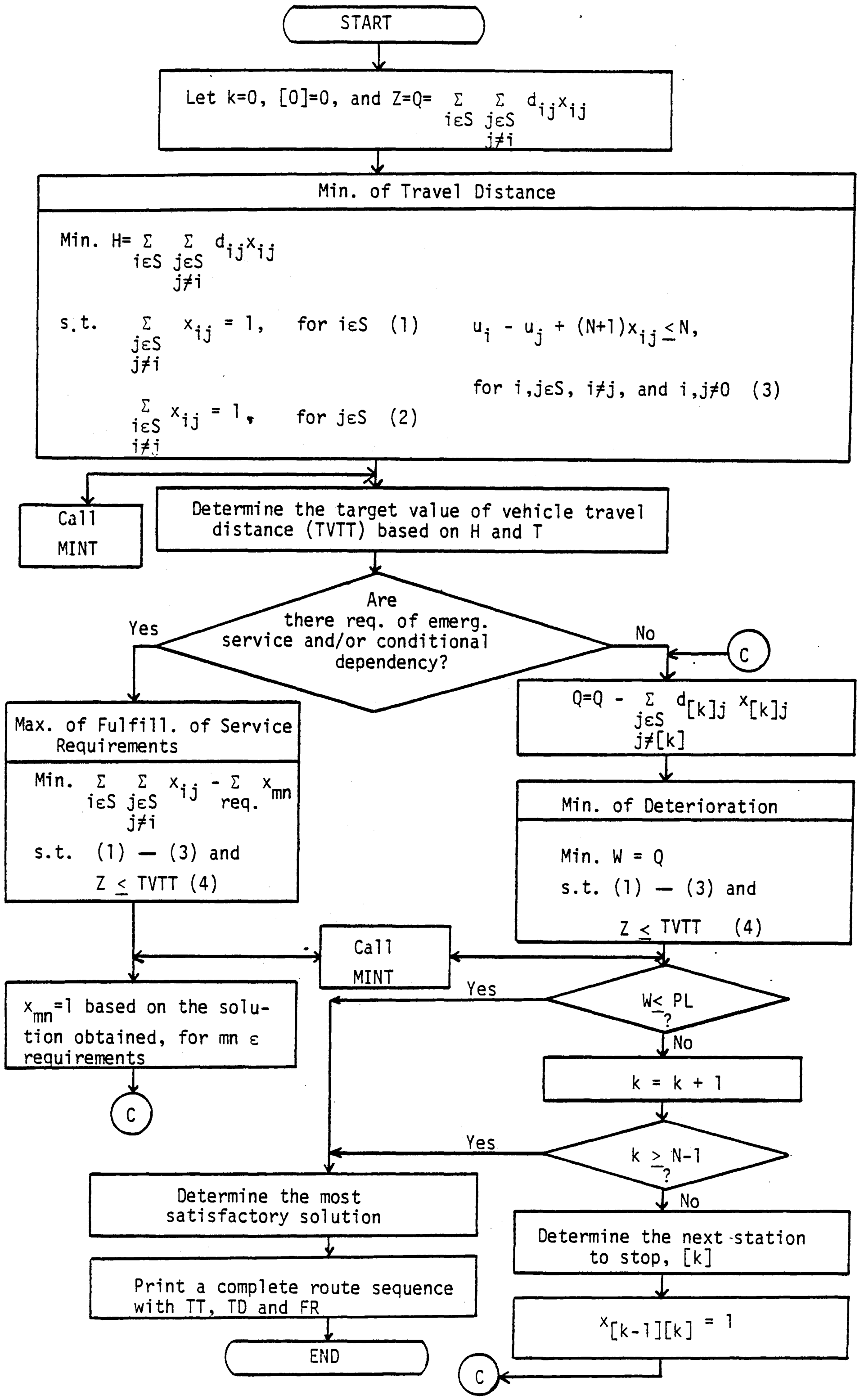

Figure 12. Logic Flow Chart of the Iterative GP Procedure with an Application of SLGP Approach for Model III

Gaskell [23], and the last one is the same as the one described by Christofides and Eilon [13] except that distance and capacity constraints are added. The detailed data for the three are reproduced in Appendix B. The data about the levels of transportation duration for goods deterioration, and stations requiring urgent services and conditionally dependent are given quite artificially, for each problem.

It is assumed that for each stop 10 distance units allowance is required. It is also assumed that the DM's goal priority structure follows Model I in problem 2 and 3. In problem 1, all three Models are considered. This is done to illustrate that the outcomes differ, depending upon the DM's preference on the goal priority structure. The target value of vehicle travel distance is reasonably determined by adding 20 units to the minimal travel distance of a route. The target value of transportation duration for goods deterioration is set equal to the predetermined level of transportation duration for goods deterioration, PL. The problem sources and conditions are presented in Table VII.

Computational Experience

Three problem sets were run on an IBM 3081D computer at Oklahoma State University. Table VIII, shows the results of four different solution procedures on the three problems. The results of the proposed algorithm in the table are based on the Model I priority structure, using an α value of 2.0 in clustering. The four procedures are:

1. The proposed algorithm,

TABLE VII

LIST OF TEST PROBLEMS

Test Problem No.	Problem Origin	No. of[a] Stations (M)	Vehicle Capacity (Q)	Maximum Allowable Vehicle Travel Distance (MT)	Predetermined Duration Level For Goods Deterioration (PL)	Upper Limit of Duration Until The Complete Goods Deterioration (UL)	Stations Requiring Emergent Services	Stations Conditionally Dependent	Models for Priority Structure
1	Gaskell [23]	21	6000	200	130	200	11,20	(2,9) (1,20)	I,II,III
2	Gaskell [23]	29	4500	240	160	235	3,9,15, 17,27	(10,5) (14,2) (4,1) (29,25) (19,8)	I
3	Christofides & Eilon [13]	50	130	160	130	180	13,15, 18,28, 42	(4,19) (8,32) (13,18) (25,14) (44,47)	I

[a]Excludes depot.

TABLE VIII

COMPARISON OF ALGORITHMS WITH
MODEL I PRIORITY STRUCTURE

Test Problem No.	Proposed Algorithm					Method A					Method B					Method C				
	Rts.	GTT	GTD	GFR	Time[a] (sec)	Rts.	GTT	GTD	GFR	Time[b] (sec)	Rts.	GTT	GTD	GFR	Time (sec)	Rts.	GTT	GTD	GFR	Time[b] (sec)
1	4	612	9	0	3.88	4	598	20	0	6.	4	648	91	0	-[c]	4	602	20	1	6.
2	5	1019	14	2	15.25	5	963	63	0	12.	5	1017	151	0	-	5	979	72	0	12.
3	8	1219	16	7	20.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

[a] IBM 3081D

[b] IBM 7090

[c] Not available

Note: Method A - Clarke and Wright's savings, multiple approach;
Method B - Clarke and Wright's savings, sequential approach;
Method C - Gaskell's savings, multiple (λ) procedure.

2. Clarke and Wright's savings, multiple approach (results available on only problems 1 and 2),
3. Clarke and Wright's savings, sequential approach (results available on only problems 1 and 2), and
4. Gaskell's savings, multiple (λ) procedure (results available on only problems 1 and 2).

While the grand total distance (GTT), grand total deterioration (GTD), and grand total fulfillment of requirements (GFR) are of concern, the number of vehicles utilized (Rts.) in all cases is also important to note. In addition, it should be pointed out that no attempt has been made to convert computing times to some comparable value. Hence, caution should be exercised in viewing solution times.

Based on solution optimality, in terms of minimum number of vehicles, minimum distance, minimum deterioration, and maximum fulfillment, the proposed algorithm produces the nondominated solutions in both cases 1 and 2. It is also seen that the proposed algorithm turns out the best results with respect to the deterioration and/or fulfillment of service requirements, without a considerable sacrifice to the distance optimality.

At the same time, the proposed technique produces routes requiring the same number of vehicles as those derived by the savings methods. It must be noted that the proposed algorithm may successively improve the solutions by changing α in the clustering stage and/or changing target values. This idea will be fully described in the next chapter. The shortcomings of the proposed algorithm lie in the fact that more than one run is necessary to solve SLGP problems during the routing procedure. The resultant computation time and computer memory requirement can therefore be substantial.

Computer times are difficult to contrast since the algorithms were programmed on a different computer. A fact of interest is the computer time of the proposed algorithm. Computer time for the algorithm may be increased linearly with an increase in the total number of stations if the number of stations per route remains relatively constant, and quadratically with the average number of stations per route if the total number of stations remains relatively constant. This is a general principle [24] applicable to Cluster First methods, including Gillet and Miller's SWEEP algorithm. This can be seen in Table VIII for the proposed algorithm. Computer time ranges from 3.88 seconds to 20.7 seconds while the average number of stations per route varies from 5.25 to 6.25, and the total number of stations from 21 to 50.

The results of test problem 1 are presented in Table IX, for three different Models. It shows that the outcomes of the problem differ, depending upon the DM's preference on the goal priority structure. Since Models I and III attempt to minimize total travel distance first, minimum deterioration and/or maximum fulfillment of service requirements are sacrificed to a certain degree. Thus, there are 9 units of deterioration and no fulfillment in Model I and 32 units of deterioration and 2 requirements fulfillment in Model III. These are the expected outcomes with regard to the 2nd priority goal in each of Models I and III. It is interesting to note that total distance and deterioration derived in Model III exceeds those obtained in Model I by 33 and 23 units, respectively, in order to attain two more fulfillment of service requirements in Model III.

TABLE IX

RESULTS OF TEST PROBLEM 1
FOR THREE MODELS

Model	Rts.	GTT	GTD	GFR	Time[a] (sec)
Model I	4	612	9	0	3.88
Model II	6	761	0	2	0.51
Model III	4	645	32	2	3.81

[a]IBM 3081D

Model II is primarily to minimize the deterioration to zero, while impacting the distance minimization and service requirements fulfillment maximization. This desired deterioration goal is achieved completely by increasing the number of vehicles, which consequently results in an increase of vehicle travel distance. In Table IX, two additional vehicles are required in Model II in order to deliver the commodity to the depot without deterioration, resulting in an increase of more than 100 distance units comparing with the outcomes in Models I and III. Model II with an average of 3.5 stations per route was solved in 0.51 seconds and, on the other hand, Models I and III with an average of 5.25 stations solved in about 3.8 seconds. This result, consistent with the general principle about computation time in Cluster First methods, implies that the proposed algorithm is extremely useful for very large problems that average only a few stations per route.

Summary

The computational experience of the proposed algorithm on three test problems is presented. Its performance is evaluated by comparing the results with those obtained by three savings methods that are for VRPs with a single objective. Based on solution optimality, the algorithm produces the nondominated solution in all cases. On the priority structure of Model I, it turns out the best results with respect to the deterioration and/or fulfillment of service requirements, without a considerable sacrifice to a distance optimality. In particular, due to the shortcomings of the computer code available for integer GP, the SLGP approach is adopted to solve a GP model at each iteration in the routing procedure.

The results of the experiments show that the algorithm is capable of performing a trade-off between the achievement levels of the objectives, based on the DM's preference regarding the goal priority structure and the target value of the goal constraints. This implies that the proposed algorithm can allow the DM to make intelligent trade-off decisions about the different objectives. This idea will be fully described in the next chapter, through an interactive procedure.

The shortcomings of the proposed algorithm lie in the fact that more than one run is necessary to solve SLGP problems during the routing procedure. The resultant computation time and computer memory requirement can therefore be substantial.

CHAPTER VI

USING THE INTERACTIVE COMPUTER PROGRAM

Introduction

Solution of a large scale multicriteria VRP requires the use of a computer. An analyst gathers all the necessary data including the DM's prior preference information on a global level, and the computer does the work. The analyst, however, may not be able to provide all the necessary preference information in advance because of the complexity of the system. Instead, he may be able to afford the information regarding trade-offs or preferences on a local level to a particular solution. An interactive method for multicriteria VRPs was developed because it has the advantage of allowing the DM to not only provide local information but also gain a greater understanding and feeling for the behavior of the system, due to involvement in the solution process.

This chapter discusses the design of the interactive procedure which implements the proposed algorithm for the multicriteria VRP where the three objectives are to be achieved as presented in previous chapters, and the use of its computer program. Test problem 1 in Table VII is used to execute the interactive program. Actual interactive ouput is interspersed with comments and explanation in the chapter, for each of the three goal priority models. The output of the interactive procedure addressed in each text appears in the Figure below it. All computer outputs shown were run on an IBM 3081D computer and generated automatically by the computer, except for the input values which follow a question mark (?). These input values are entered by the user.

Interactive Procedure

The procedure consists of two types of interactions. First, the DM is asked about explicit information, based on the current solution of a route, regarding the trade-off between the attainment levels of objectives by changing the target values or preference on the goal priority structure, in order to reach a new preferred solution of the route. Second, the DM is solicited for explicit information, based on the current complete solution of routes, regarding the trade-off between the routes with respect to the achievement level of the objectives. This may cause some station(s) in a subset to cluster to another subset, building up a new form of subsets. A flow chart of the interactive procedure appears in Figure 13. The dotted-line in the Figure represents a User-Machine dialogue, through which a progressive articulation takes place.

The entire interactive computer program coded in FORTRAN appears in Appendix A. In the program, care was taken to reduce the user's burden in providing the computer with the parameters. For example, the minimal vehicle travel distance on a route is given to help the user in determining the target value of the vehicle travel distance. The computer prompts the user for all necessary inputs. These values are presented to the user for either verification or change. In

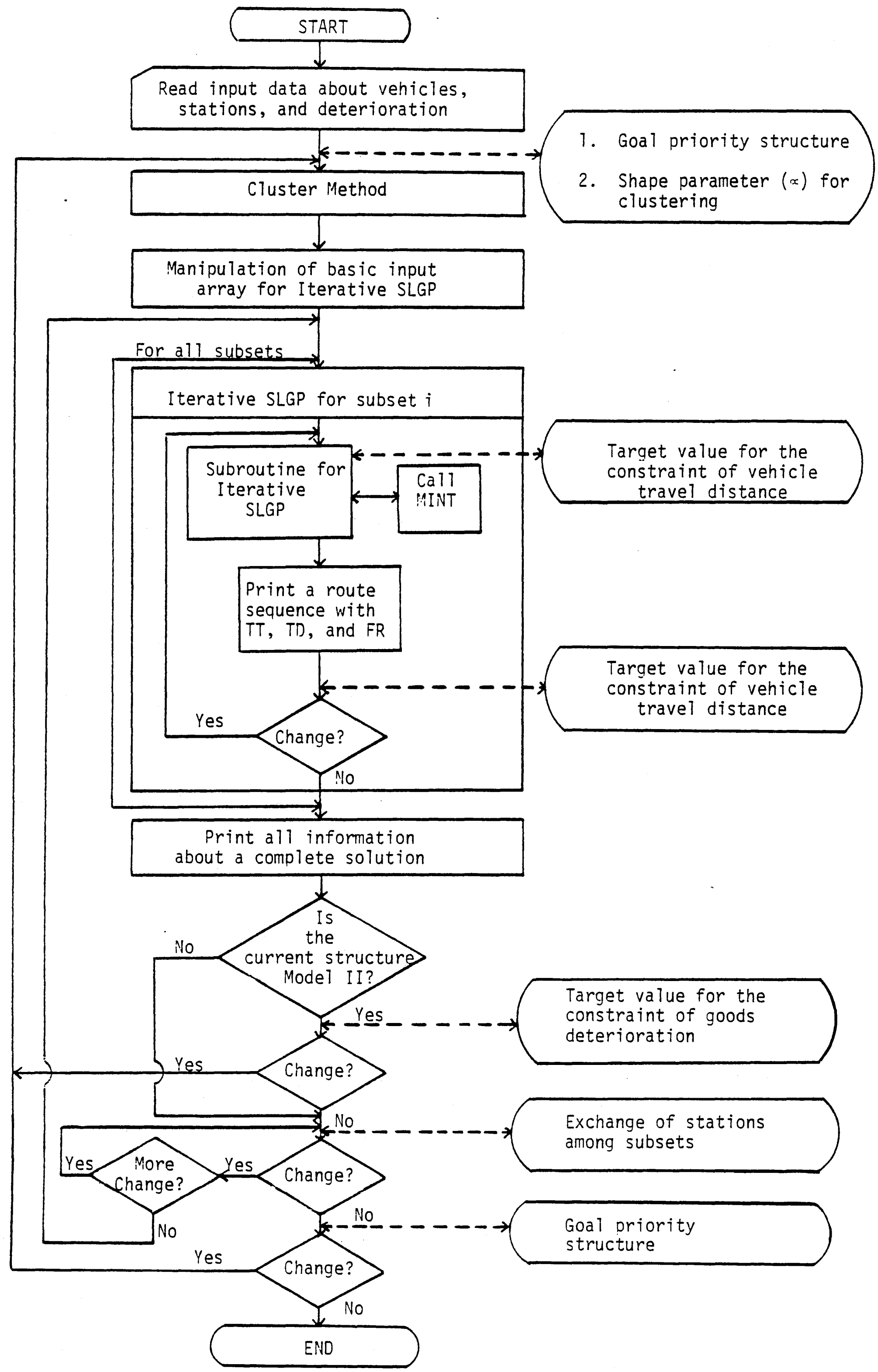

Figure 13. The Logic Flow Chart of the Proposed Interactive Procedure

addition, the user's inputs are checked for their appropriateness and the user is prompted to correct probable errors or inconsistencies. Only when a set of inputs has been checked by the program and verified by the user does the program continue.

When several values are to be entered, they need only be seperated by a space or a comma. The input mechanism is virtually self-explanatory, as long as the user understands the terms being input. Thus, any person, without any previous familiarity with a computer or mathematical programming, can easily use this program to determine the most favorable solution of a multicriteria VRP.

The interactive program reaches the most favorable route sequences through repeatedly changing:

1. the goal priority structure,
2. the target values of the constraints, and
3. the subsets (clusters) formation.

Procedure on the Goal Priority Structure Model I

The program begins by presenting the main options menu. The selection of "1" from this menu indicates that the structure of Model I in Table III is to be employed as the user's goal priority structure. After Model I is selected, the program presents the user a summary of input data and prompts him to enter an α value (shape parameter) for clustering. The output of the distance matrix and of the clustered subsets of stations are presented. The distance matrix is constructed by computing the distances of stations based on the polar coordinates. It is noted that, in all the three

Models, the target value of the deterioration constraint is initially set equal to the predetermined level for goods deterioration, PL.

```
===> GOAL PRI. MENU <===
ENTER OPTION NO.
   1: TRAVEL DIST.=1, DETERIORATION=2, FULFILLMENT OF SERVICE REQ.=3
   2: TRAVEL DIST.=2, DETERIORATION=1, FULFILLMENT OF SERVICE REQ.=3
   3: TRAVEL DIST.=1, DETERIORATION=3, FULFILLMENT OF SERVICE REQ.=2
?
1

THE INPUT DATA GIVEN ARE SUMMARIZED AS FOLLOWS:
   NO. OF STATIONS=   21
   LIMIT OF VEHICLE CAPACITY= 6000
   MAX. ALLOWABLE VEHICLE TRAVEL DISTANCE=  200
   NO. OF TOTAL EMERG. SERV. REQ.=  2
   NO. OF TOTAL COND. DEP. OF STATIONS=  2
   PREDETERMINED LEVEL OF DISTANCE FOR DETERIORATION=  130
   UPPER LEVEL OF DISTANCE FOR THE COMPLETE DETERI.=  200
   STATIONS REQUIRING EMERG. SERV.=  20  11
   CONDITIONALLY DEPEN. STAT.=  ( 2, 9) ( 1,20)

===> ENTER ALPHA VALUE FOR CLUSTERING <===
?
2.0
ALPHA VALUE ENTERED IS: 2.00

** THE DISTANCE MATRIX
```

0	8	23	25	20	18	24	26	30	32	40	47	54	56	57	58	71	72	78	79	82	49
8	0	29	32	14	19	19	27	25	31	43	44	55	54	53	57	69	68	78	76	81	48
23	29	0	2	33	17	33	19	37	28	23	45	40	48	57	49	63	69	65	73	72	41
25	32	2	0	35	18	34	19	38	28	21	44	38	47	56	47	61	69	63	72	70	40
20	14	33	35	0	17	5	22	11	21	38	30	47	42	39	46	56	54	67	62	69	36
18	19	17	18	17	0	15	8	19	14	23	30	36	38	42	40	53	55	59	61	64	31
24	19	33	34	5	15	0	19	6	16	34	25	42	37	34	41	50	49	61	57	63	31
26	27	19	19	22	8	19	0	21	9	16	26	28	31	38	33	46	50	51	55	57	24
30	25	37	38	11	19	6	21	0	15	35	20	40	32	28	37	45	43	58	51	59	27
32	31	28	28	21	14	16	9	15	0	20	17	26	24	28	26	39	42	47	47	50	17
40	43	23	21	38	23	34	16	35	20	0	31	17	29	42	28	42	52	42	53	50	23
47	44	45	44	30	30	25	26	20	17	31	0	27	13	12	18	25	25	38	32	38	11
54	55	40	38	47	36	42	28	40	26	17	27	0	18	35	14	27	40	25	38	33	16
56	54	48	47	42	38	37	31	32	24	29	13	18	0	18	5	15	23	25	24	26	7
57	53	57	56	39	42	34	38	28	28	42	12	35	18	0	23	22	15	39	24	36	20
58	57	49	47	46	40	41	33	37	26	28	18	14	5	23	0	14	26	20	25	24	9
71	69	63	61	56	53	50	46	45	39	42	25	27	15	22	14	0	17	18	11	13	22
72	68	69	69	54	55	49	50	43	42	52	25	40	23	15	26	17	0	35	12	27	29
78	78	65	63	67	59	61	51	58	47	42	38	25	25	39	20	18	35	0	26	12	30
79	76	73	72	62	61	57	55	51	47	53	32	38	24	24	25	11	12	26	0	16	31
82	81	72	70	69	64	63	57	59	50	50	38	33	26	36	24	13	27	12	16	0	33
49	48	41	40	36	31	31	24	27	17	23	11	16	7	20	9	22	29	30	31	33	0

```
** THE CLUSTERED SUBSETS
     1   6   2  10   5   7   8
     3   4  11   9  13
    21  19  16  14
    20  17  18  15  12
```

The Iterative SLGP is applied to all subsets, starting with subset (cluster) 1. The program, initially for subset 1, presents a summary of service requirements with the computed vehicle load. The program computes the minimum travel distance of the route. Based on this, as well as the upper bound for the constraint on vehicle travel distance T, it prompts the user to enter a target value for the vehicle travel distance. Here, the user enters 185 units. The program runs the Iterative SLGP and presents to the user a route sequence with TT, TD, and FR. Based on the information provided, the user is asked if he wants to change the target value of the vehicle travel distance in an effort to obtain a new preferred solution. In this example, the user desires to relax the target value to 200 units. A new solution is then presented with an increased TT and a decreased TD. The user is asked again about he wants to change the target value. A selection of "2" from the menu leads to subset 2.

```
     ** ITERATIVE SLGP APPL. TO CLUSTER  1
      1   6   2  10   5   7   8
   A VEHICLE LOAD:  5800
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  180
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
185
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  185

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  1 IS:
   ROUTE SEQUENCE:   0   7   5   2   1   6   8  10   0
   TOT. DIS.= 180   TOT. DET.=  9   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1
```

```
   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  180
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
200
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  200

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  1 IS:
   ROUTE SEQUENCE:   0   1   2   5   7   6   8  10   0
   TOT. DIS.= 195    TOT. DET.=  6   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2
```

In the next three subsets, the procedure proceeds in a similar manner as subset 1. Here, it is clearly seen that the trade-off between the achievement levels of the objectives are attained by changing the target value of travel distance. Once all subsets are routed on the basis of the user's preference, a complete solution is presented.

```
     ** ITERATIVE SLGP APPL. TO CLUSTER  2
      3   4  11   9  13
   A VEHICLE LOAD:  5200
   STATIONS FOR EMERG. SERV.:   11
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  170
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
190
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  190

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  2 IS:
   ROUTE SEQUENCE:   0   9   3   4  11  13   0
   TOT. DIS.= 170   TOT. DET.=  3   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  3
     21  19  16  14
```

```
   A VEHICLE LOAD:  5600
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  114
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
125
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  125

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  3 IS:
   ROUTE SEQUENCE:   0  21  19  16  14   0
   TOT. DIS.= 117   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  114
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
140
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  140

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  3 IS:
   ROUTE SEQUENCE:   0  21  19  16  14   0
   TOT. DIS.= 117   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.= 0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  4
     20  17  18  15  12
   A VEHICLE LOAD:  5900
   STATIONS FOR EMERG. SERV.:  20
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
140
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  140

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQUENCE:   0  12  15  18  20  17   0
   TOT. DIS.= 133   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1
```

```
   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
150
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:   150

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQUENCE:   0  20  17  18  15  12   0
   TOT. DIS.= 147   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2


   ** ROUTING IS COMPLETED FOR ALL CLUSTERS
        AND A COMPLETE SOLUTION IS OBTAINED AS FOLLOWS:
   TOT. TRAVEL DIST.=  629
   TOT. DETERIORATION=    9
   TOT. FULL. OF SERVICE REQ.=  1
   VEH. LOAD= 5800 TT= 195 TD= 6 FR= 0 RT. SEQ.=  0  1  2  5  7  6  8 10  0
   VEH. LOAD= 5200 TT= 170 TD= 3 FR= 0 RT. SEQ.=  0  9  3  4 11 13  0
   VEH. LOAD= 5600 TT= 117 TD= 0 FR= 0 RT. SEQ.=  0 21 19 16 14  0
   VEH. LOAD= 5900 TT= 147 TD= 0 FR= 1 RT. SEQ.= 0 20 17 18 15 12  0
```

In an effort to obtain a new preferred complete solution, a menu is presented so that any of stations in subsets can be exchanged as long as it does not violate any restrictions, such as the vehicle capacity and travel distance. Note that the program checks the user's input with regard to the vehicle capacity and prompts the user with helpful error messages. The exchanges are continued until the user selects "2" from the menu. Then a new form of subsets based on the exchanges are presented.

```
DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

   ENTER ONE CLUSTER NO., ITS STATION NO. AND THE OTHER CLUSTER NO.,
   ITS STATION NO., FOR EXCHANGE OF STATIONS
?
2  3  4  20

!ERROR! VEH. CAPACITY RESTRICTION IS VIOLATED!! DO IT AGAIN!
```

```
DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

   ENTER ONE CLUSTER NO., ITS STATION NO. AND THE OTHER CLUSTER NO.,
   ITS STATION NO., FOR EXCHANGE OF STATIONS
?
1  13  2  4


EXCHANGED STATIONS ARE:
   STATION NO. 13 IN CLUSTER NO.  1 AND STATION NO.  4 IN CLUSTER NO.  2
!ERROR!, CHECK INPUT DATA!!

DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1


   ENTER ONE CLUSTER NO., ITS STATION NO. AND THE OTHER CLUSTER NO.,
   ITS STATION NO., FOR EXCHANGE OF STATIONS
?
1  2  3  21

EXCHANGED STATIONS ARE:
   STATION NO.  2 IN CLUSTER NO.  1 AND STATION NO. 21 IN CLUSTER NO.  3

DO YOU WANT TO CONTINUE TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

   ENTER ONE CLUSTER NO., ITS STATION NO. AND THE OTHER CLUSTER NO.,
   ITS STATION NO., FOR EXCHANGE OF STATIONS
?
2  9  3  14

EXCHANGED STATIONS ARE:
   STATION NO.  9 IN CLUSTER NO.  2 AND STATION NO. 14 IN CLUSTER NO.  3

DO YOU WANT TO CONTINUE TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2

** THE CLUSTERED SUBSETS
      1   6  21  10   5   7   8
      3   4  11  14  13
      2  19  16   9
     20  17  18  15  12
```

Again, Iterative SLGP is applied to all subsets, starting with subset 1. At the beginning of each subset, the program computes the minimal travel distance and compares it with the upper bound for the constraint on vehicle travel distance for the feasibility test of the route. Here, in subset 1, the violation of the restriction is discovered and a helpful error message is presented. The program then prompts the user to convert the current subset 1 formation to the previous one. After the conversion, the user is again allowed to exchange stations among subsets if desired. Here, the user does not show the desire by selecting "2" from the menu. In this case a new from of subsets is presented.

```
     ** ITERATIVE SLGP APPL. TO CLUSTER   1
      1    6   21   10    5    7    8
   A VEHICLE LOAD:   5800
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

OPTIMALITY ESTABLISHED
END OF PROBLEM, ITERATION NO.      25
!ERROR! RESTRICTION ON VEH. TRAV. DIST. IS VIOLATED!!
CONVERT TO THE PREVIOUS SUBSETS FORMATION!

   ENTER ONE CLUSTER NO., ITS STATION NO. AND THE OTHER CLUSTER NO.,
   ITS STATION NO., FOR EXCHANGE OF STATIONS
?
1   21   3   2

EXCHANGED STATIONS ARE:
   STATION NO. 21 IN CLUSTER NO.   1 AND STATION NO.   2 IN CLUSTER NO.   3

DO YOU WANT TO CONTINUE TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2

** THE CLUSTERED SUBSETS
      1    6    2   10    5    7    8
      3    4   11   14   13
     21   19   16    9
     20   17   18   15   12
```

Again, the Iterative SLGP is applied to all subsets, starting from subset 1. Basically the same procedure as for the previous form of subsets is followed. It is also seen that the trade-off between the achievement levels of the objectives are attained by changing the target value of the vehicle travel distance. A complete solution is presented.

```
      ** ITERATIVE SLGP APPL. TO CLUSTER  1
       1   6   2  10   5   7   8
    A VEHICLE LOAD:  5800
    NO. OF EMERG. SERV. REQ.= 0
    NO. OF COND. DEP. STA.= 0

    ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  180
    ** RESTRICTION ON VEH. TRAV. DIST. IS  200

    ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
    THE INFORMATION GIVEN ABOVE.
?
200
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  200

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  1 IS:
    ROUTE SEQUENCE:   O   1   2   5   7   6   8  10   O
    TOT. DIS.= 195   TOT. DET.=  6   TOT. FULL. OF EM. SERV. & COND. DEP.=  O

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
    1:YES   2:NO
?
2

      ** ITERATIVE SLGP APPL. TO CLUSTER  2
       3   4  11  14  13
    A VEHICLE LOAD:  5000
    STATIONS FOR EMERG. SERV.:  11
    NO. OF EMERG. SERV. REQ.= 1
    NO. OF COND. DEP. STA.= 0

    ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  161
    ** RESTRICTION ON VEH. TRAV. DIST. IS  200

    ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
    THE INFORMATION GIVEN ABOVE.
?
190
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  190

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  2 IS:
    ROUTE SEQUENCE:   O  11   4   3  13  14   O
    TOT. DIS.= 161   TOT. DET.=  O   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
    1:YES   2:NO
?
2

      ** ITERATIVE SLGP APPL. TO CLUSTER  3
      21  19  16   9
    A VEHICLE LOAD:  5800
    NO. OF EMERG. SERV. REQ.= 0
    NO. OF COND. DEP. STA.= 0

    ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  167
    ** RESTRICTION ON VEH. TRAV. DIST. IS  200

    ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
    THE INFORMATION GIVEN ABOVE.
?
180
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  180
```

```
** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  3 IS:
   ROUTE SEQUENCE:   0  21  19  16   9   0
   TOT. DIS.= 169   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  4
     20  17  18  15  12
   A VEHICLE LOAD:  5900
   STATIONS FOR EMERG. SERV.:  20
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
170
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  170

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQUENCE:   0  20  17  15  18  12  0
   TOT. DIS.= 165   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
150
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  150

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQUENCE:   0  20  17  18  15  12  0
   TOT. DIS.= 147   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
140
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  140
```

```
** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQUENCE:   0  12  15  18  20  17   0
   TOT. DIS.= 133   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEH. TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
150
 TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  150

** THE MOST SATISFACTORY ROUTE SEQUENCE OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQUENCE:   0  20  17  18  15  12   0
   TOT. DIS.= 147   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2


   ** ROUTING IS COMPLETED FOR ALL CLUSTERS
        AND A COMPLETE SOLUTION IS OBTAINED AS FOLLOWS:
   TOT. TRAVEL DIST.=  672
   TOT. DETERIORATION=    6
   TOT. FULL. OF SERVICE REQ.=   2
   VEH. LOAD= 5800 TT= 195 TD= 6 FR= 0 RT. SEQ.=  0  1  2  5  7  6  8 10  0
   VEH. LOAD= 5000 TT= 161 TD= 0 FR= 1 RT. SEQ.=  0 11  4  3 13 14  0
   VEH. LOAD= 5800 TT= 169 TD= 0 FR= 0 RT. SEQ.=  0 21 19 16  9  0
   VEH. LOAD= 5900 TT= 147 TD= 0 FR= 1 RT. SEQ.=  0 20 17 18 15 12  0

DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2
```

Procedure on the Goal Priority Structure Model II

After a new complete solution is obtained, the program prompts the user to enter the option number which represents the change of the goal priority structure. A selection of "2" from this menu leads to the end of the interactive procedure. Here, a change is attempted by selecting "1" from the menu. The major goal priority structure options menu is presented. A selection of "2" from this menu indicates that the structure of Model II is employed. The program then presents to the user a summary of input data and prompts him to enter the ∝ value for clustering. Here the user inputs 2.0. The program then runs the Cluster Method in order to partition a set of stations into subsets and its output is presented. It is noted that the target value of the deterioration constraint is initially set equal to the predetermined level of transportation duration for goods deterioration.

```
DO YOU WANT TO CHANGE THE GOAL PRIORITY STRUCTURE?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

===> GOAL PRI. MENU <===
ENTER OPTION NO.
   1: TRAVEL DIST.=1, DETERIORATION=2, FULFILLMENT OF SERVICE REQ.=3
   2: TRAVEL DIST.=2, DETERIORATION=1, FULFILLMENT OF SERVICE REQ.=3
   3: TRAVEL DIST.=1, DETERIORATION=3, FULFILLMENT OF SERVICE REQ.=2
?
2

THE INPUT DATA GIVEN ARE SUMMARIZED AS FOLLOWS:
   NO. OF STATIONS=   21
   LIMIT OF VEHICLE CAPACITY= 6000
   MAX. ALLOWABLE VEHICLE TRAVEL DISTANCE=  200
   NO. OF TOTAL EMERG. SERV. REQ.=  2
   NO. OF TOTAL COND. DEP. OF STATIONS=  2
   PREDETERMINED LEVEL OF DISTANCE FOR DETERIORATION=  130
   UPPER LEVEL OF DISTANCE FOR THE COMPLETE DETERI.=  200
   STATIONS REQUIRING EMERG. SERV.=  20  11
   CONDITIONALLY DEPEN. STAT.=  ( 2, 9) ( 1,20)

===> ENTER ALPHA VALUE FOR CLUSTERING <===
?
2.0
ALPHA VALUE ENTERED IS: 2.00

** THE CLUSTERED SUBSETS
       1   6  10
       2   5   7
       3   4   8  11
      21  19  16  14
      20  17  18  15
       9  13  12
```

Iterative SLGP is applied to all subsets, starting with subset 1. The most favorable route sequence is presented with TT, TD, and FR, for each subset. In subsets 3 and 5, the program prompts the user with the minimal travel distance of the route computed and he must enter the target value of the vehicle travel distance. This input is required because the third priority goal, OBFR, is to be considered in both routes. It is seen that the trade-off between the achievement levels of the objectives are attained by changing the target value of travel distance. Once all subsets are routed, a complete solution is presented.

```
   ** ITERATIVE SLGP APPL. TO CLUSTER  1
    1   6  10
 A VEHICLE LOAD:  2100
 NO. OF EMERG. SERV. REQ.= 0
 NO. OF COND. DEP. STA.= 0

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  1 IS:
 ROUTE SEQ.:    0    6    1   10    0
 TOT. DIST.= 128    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  0

   ** ITERATIVE SLGP APPL. TO CLUSTER  2
    2   5   7
 A VEHICLE LOAD:  3600
 NO. OF EMERG. SERV. REQ.= 0
 NO. OF COND. DEP. STA.= 0

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  2 IS:
 ROUTE SEQ.:    0    7    5    2    0
 TOT. DIST.= 128    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  0

   ** ITERATIVE SLGP APPL. TO CLUSTER  3
    3   4   8  11
 A VEHICLE LOAD:  3500
 STATIONS FOR EMERG. SERV.:  11
 NO. OF EMERG. SERV. REQ.= 1
 NO. OF COND. DEP. STA.= 0

 ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  129
 ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

 ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
 THE INFORMATION GIVEN ABOVE.
?
145
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  145

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  3 IS:
 ROUTE SEQ.:    0   11    8    3    4    0
 TOT. DIST.= 140    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  1
```

```
DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  129
   ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

   ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
135
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  135

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  3 IS:
   ROUTE SEQ.:   O  11   4   3   8   O
   TOT. DIST.= 129   TOT. DET.=  O   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  4
     21  19  16  14
   A VEHICLE LOAD:  5600
   NO. OF EMERG. SERV. REQ.= O
   NO. OF COND. DEP. STA.= O

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQ.:   O  14  21  19  16   O
   TOT. DIST.= 114   TOT. DET.=  O   TOT. FULL. OF EM. SERV. & COND. DEP.=  O

     ** ITERATIVE SLGP APPL. TO CLUSTER  5
     20  17  18  15
   A VEHICLE LOAD:  4600
   STATIONS FOR EMERG. SERV.:  20
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= O

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  120
   ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

   ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
125
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  125

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  5 IS:
   ROUTE SEQ.:   O  17  20  18  15   O
   TOT. DIST.= 120   TOT. DET.=  O   TOT. FULL. OF EM. SERV. & COND. DEP.=  O

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  120
   ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

   ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
140
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  140
```

```
** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  5 IS:
   ROUTE SEQ.:    0  20  17  18  15   0
   TOT. DIST.= 134    TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  6
     9  13  12
   A VEHICLE LOAD:  3100
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  6 IS:
   ROUTE SEQ.:    0  13   9  12   0
   TOT. DIST.= 117    TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

   ** ROUTING IS COMPLETED FOR ALL CLUSTERS
      AND A COMPLETE SOLUTION IS OBTAINED AS FOLLOWS:
   TOT. TRAVEL DIST.=  750
   TOT. DETERIORATION=    0
   TOT. FULL. OF SERVICE REQ.=  2
   VEH. LOAD= 2100 TT= 128 TD= 0 FR= 0 RT. SEQ.=  0  6  1 10  0
   VEH. LOAD= 3600 TT= 128 TD= 0 FR= 0 RT. SEQ.=  0  7  5  2  0
   VEH. LOAD= 3500 TT= 129 TD= 0 FR= 1 RT. SEQ.=  0 11  4  3  8  0
   VEH. LOAD= 5600 TT= 114 TD= 0 FR= 0 RT. SEQ.=  0 14 21 19 16  0
   VEH. LOAD= 4600 TT= 134 TD= 0 FR= 1 RT. SEQ.=  0 20 17 18 15  0
   VEH. LOAD= 3100 TT= 117 TD= 0 FR= 0 RT. SEQ.=  0 13  9 12  0
```

The user is then asked if he wants to change the target value of the transportation duration for goods deterioration. A selection of "1" from the menu, followed by entering its new target value, leads to the newly clustered subsets. The program then runs Iterative SLGP for each of the subsets. It is clear that a trade-off between the achievement levels of the objectives are attained by changing the target value of the transportation duration for goods deterioration.

```
DO YOU WANT TO CHANGE TARGET VALUE FOR TD?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

   ** PREDETERMINED  LEVEL OF DISTANCE FOR DETERIORATION IS:  130
   ** CURRENT TARGET VALUE FOR THE DETERI. CONSTRAINT IS:  130

   ENTER NEW TARGET VLAUE FOR THE DETERI. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
155
NEW TARGET VALUE FOR TD IS:  155

THE INPUT DATA GIVEN ARE SUMMARIZED AS FOLLOWS:
   NO. OF STATIONS=   21
   LIMIT OF VEHICLE CAPACITY= 6000
   MAX. ALLOWABLE VEHICLE TRAVEL DISTANCE=  200
   NO. OF TOTAL EMERG. SERV. REQ.=  2
   NO. OF TOTAL COND. DEP. OF STATIONS=  2
   PREDETERMINED LEVEL OF DISTANCE FOR DETERIORATION=  130
   UPPER LEVEL OF DISTANCE FOR THE COMPLETE DETERI.=  200
   STATIONS REQUIRING EMERG. SERV.=  20  11
   CONDITIONALLY DEPEN. STAT.=  ( 2, 9) ( 1,20)

===> ENTER ALPHA VALUE FOR CLUSTERING <===
?
2.0
ALPHA VALUE ENTERED IS: 2.00

** THE CLUSTERED SUBSETS
      1   6   2  10
      3   4   8  11  13
      5   7   9  15  12
     21  19  16  14
     20  17  18

     ** ITERATIVE SLGP APPL. TO CLUSTER  1
      1   6   2  10
   A VEHICLE LOAD:  2800
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  1 IS:
   ROUTE SEQ.:   0  10   6   1   2   0
   TOT. DIST.= 145   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

     ** ITERATIVE SLGP APPL. TO CLUSTER  2
      3   4   8  11  13
   A VEHICLE LOAD:  4800
   STATIONS FOR EMERG. SERV.:  11
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  149
   ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

   ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
160
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  160

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  2 IS:
   ROUTE SEQ.:   0  11   4   3   8  13   0
   TOT. DIST.= 159   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1
```

```
DC YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  3
      5   7   9  15  12
   A VEHICLE LOAD:  5600
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  3 IS:
   ROUTE SEQ.:    0   12   15    9    7    5    0
   TOT. DIST.= 148    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  0

     ** ITERATIVE SLGP APPL. TO CLUSTER  4
     21  19  16  14
   A VEHICLE LOAD:  5600
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQ.:    0   14   21   19   16    0
   TOT. DIST.= 114    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  0

     ** ITERATIVE SLGP APPL. TO CLUSTER  5
     20  17  18
   A VEHICLE LOAD:  3700
   STATIONS FOR EMERG. SERV.:  20
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  104
   ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

   ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
110
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  110

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  5 IS:
   ROUTE SEQ.:    0   18   20   17    0
   TOT. DIST.= 104    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
1

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  104
   ** RESTRICTION ON VEH. TRAVEL DIST. IS  200

   ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
140
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  140

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  5 IS:
   ROUTE SEQ.:    0   20   18   17    0
   TOT. DIST.= 112    TOT. DET.=  0    TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES    2:NO
?
2
```

```
   ** ROUTING IS COMPLETED FOR ALL CLUSTERS
      AND A COMPLETE SOLUTION IS OBTAINED AS FOLLOWS:
   TOT. TRAVEL DIST.=  678
   TOT. DETERIORATION=    0
   TOT. FULL. OF SERVICE REQ.=  2
   VEH. LOAD= 2800 TT= 145 TD= 0 FR= 0 RT. SEQ.=  0 10  6  1  2  0
   VEH. LOAD= 4800 TT= 159 TD= 0 FR= 1 RT. SEQ.=  0 11  4  3  8 13  0
   VEH. LOAD= 5600 TT= 148 TD= 0 FR= 0 RT. SEQ.=  0 12 15  9  7  5  0
   VEH. LOAD= 5600 TT= 114 TD= 0 FR= 0 RT. SEQ.=  0 14 21 19 16  0
   VEH. LOAD= 3700 TT= 112 TD= 0 FR= 1 RT. SEQ.=  0 20 18 17  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TD?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2
```

Procedure on the Goal Priority

Structure Model III

The program, again, prompts the user to enter the option number which represents the change of the goal priority structure. Here its change is attempted by selecting "1" from the menu. The major goal priority structure options menu is then presented. A selection of "3" from this menu indicates that the structure of Model III is employed. The interactive procedure and outputs on this Model follow the same basic structure as on Model I. It is seen through the procedure that the trade-off between the achievement levels of the objectives are attained by changing the target value of travel distance. After a complete solution is presented, the program prompts the user to enter the option number which represents the change of the goal priority structure. In the menu, a selection of "2" ends execution of the interactive computer program.

```
DO YOU WANT TO CHANGE THE GOAL PRIORITY STRUCTURE?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
1


===> GOAL PRI. MENU <===
ENTER OPTION NO.
   1: TRAVEL DIST.=1, DETERIORATION=2, FULFILLMENT OF SERVICE REQ.=3
   2: TRAVEL DIST.=2, DETERIORATION=1, FULFILLMENT OF SERVICE REQ.=3
   3: TRAVEL DIST.=1, DETERIORATION=3, FULFILLMENT OF SERVICE REQ.=2
?
3

THE INPUT DATA GIVEN ARE SUMMARIZED AS FOLLOWS:
   NO. OF STATIONS=   21
   LIMIT OF VEHICLE CAPACITY= 6000
   MAX. ALLOWABLE VEHICLE TRAVEL DISTANCE=  200
   NO. OF TOTAL EMERG. SERV. REQ.=  2
   NO. OF TOTAL COND. DEP. OF STATIONS=  2
   PREDETERMINED LEVEL OF DISTANCE FOR DETERIORATION=  130
   UPPER LEVEL OF DISTANCE FOR THE COMPLETE DETERI.=  200
   STATIONS REQUIRING EMERG. SERV.=  20  11
   CONDITIONALLY DEPEN. STAT.=  ( 2, 9) ( 1,20)

===> ENTER ALPHA VALUE FOR CLUSTERING <===
?
2.0
ALPHA VALUE ENTERED IS: 2.00

** THE CLUSTERED SUBSETS
     1   6   2  10   5   7   8
     3   4  11   9  13
    21  19  16  14
    20  17  18  15  12

     ** ITERATIVE SLGP APPL. TO CLUSTER  1
     1   6   2  10   5   7   8
   A VEHICLE LOAD:  5800
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  180
   ** RESTRICTION ON VEHICLE TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TARV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
190
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  190

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  1 IS:
   ROUTE SEQ.:   0   7   5   2   1   6   8  10   0
   TOT. DIST.= 180   TOT. DET.=  9   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  2
     3   4  11   9  13
   A VEHICLE LOAD:  5200
   STATIONS FOR EMERG. SERV.:  11
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  170
```

```
   ** RESTRICTION ON VEHICLE TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TARV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
190
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  190

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  2 IS:
   ROUTE SEQ.:   0  11   4   3   9  13   0
   TOT. DIST.= 189   TOT. DET.= 26   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  3
     21  19  16  14
   A VEHICLE LOAD:  5600
   NO. OF EMERG. SERV. REQ.= 0
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  114
   ** RESTRICTION ON VEHICLE TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TARV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
120
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  120

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  3 IS:
   ROUTE SEQ.:   0  21  19  16  14   0
   TOT. DIST.= 117   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

     ** ITERATIVE SLGP APPL. TO CLUSTER  4
     20  17  18  15  12
   A VEHICLE LOAD:  5900
   STATIONS FOR EMERG. SERV.:  20
   NO. OF EMERG. SERV. REQ.= 1
   NO. OF COND. DEP. STA.= 0

   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEHICLE TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TARV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
145
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  145

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQ.:   0  17  20  18  15  12   0
   TOT. DIST.= 133   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  0

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO

?
1
```

```
   ** MINIMAL TRAVEL DIST. OF THE ROUTE IS  133
   ** RESTRICTION ON VEHICLE TRAV. DIST. IS  200

   ENTER UPPER LIMIT OF TARV. DIST. CONSTRAINT BASED ON
   THE INFORMATION GIVEN ABOVE.
?
150
TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:  150

** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR CLUSTER  4 IS:
   ROUTE SEQ.:   0  20  17  18  15  12  0
   TOT. DIST.= 147   TOT. DET.=  0   TOT. FULL. OF EM. SERV. & COND. DEP.=  1

DO YOU WANT TO CHANGE TARGET VALUE FOR TT?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2


   ** ROUTING IS COMPLETED FOR ALL CLUSTERS
        AND A COMPLETE SOLUTION IS OBTAINED AS FOLLOWS:
   TOT. TRAVEL DIST.=  633
   TOT. DETERIORATION=   35
   TOT. FULL. OF SERVICE REQ.=  2
   VEH. LOAD= 5800 TT= 180 TD= 9 FR= 0 RT. SEQ.=  0  7  5  2  1  6  8 10  0
   VEH. LOAD= 5200 TT= 189 TD=26 FR= 1 RT. SEQ.=  0 11  4  3  9 13  0
   VEH. LOAD= 5600 TT= 117 TD= 0 FR= 0 RT. SEQ.=  0 21 19 16 14  0
   VEH. LOAD= 5900 TT= 147 TD= 0 FR= 1 RT. SEQ.=  0 20 17 18 15 12  0

DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2

DO YOU WANT TO CHANGE THE GOAL PRIORITY STRUCTURE?
===> ENTER OPTION NUMBER <===
   1:YES   2:NO
?
2


*** THE MOST FAVORABLE VEHICLE ROUTE SEQUENCES ARE DETERMINED
   WITH RESPECT TO THE DECISION MAKER'S PREFERENCE
```

Summary

Almost all the features of the interactive computer program are illustrated in this chapter. Several examples are given which describe the capabilities of this computer program. In particular, through the change of the target values and the DM's goal priority structure, it is shown that the proposed algorithm successfully performs the trade-off between the achievement levels of the objectives in a reasonable way.

The interactive and user-oriented features of this program make it a flexible and convenient tool in reaching the most favorable vehicle routes for a multicriteria VRP, with respect to a DM's preference. It allows any person, without previous familiarity with a computer or mathematical programming, to practically use and benefit from the results of this research. Furthermore, it allows a DM to not only provide local preference information but also gain understanding and feeling for the behavior of the system. As such it will help the implementation of the proposed algorithm for multicriteria VRPs in practice.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

This chapter includes a summary of how the research objectives set forth in Chapter I were accomplished, a summary of the results, and suggestions for future research.

Conclusions

VRP is a generic name given to a whole class of problems involving the visiting of "stations" by "vehicles". In recent years, many researchers have been concerned with developing solution methods for VRPs with a single objective. However, the collection or delivery problems inherent in VRPs may not lend themselves to a model construction concerning only one objective and may involve relevant multiple objectives, creating mulitcritieria VRP. In this research, three objectives were considered: the minimization of total travel distance of vehicles, the minimization of total deterioration of goods during transportation, and the maximization of total fulfillment of emergent services and conditional dependencies of stations.

The literature of VRP solving techniques, particularly for single-depot, multiple-vehicle and multiple-depot, multiple-vehicle cases, was surveyed extensively and described in Chapter II of this dissertation. Chapter III discussed the multiple objective optimization analysis that consisted of the nondominated solutions set, Goal

Programming, and interactive methods for multiple objective decision making. The research work was done in two phases. Phase I research work concentrated on the development of an algorithm, to determine the most satisfactory vehicle routes of multicriteria VRPs where the three objectives are to be achieved. Phase II focused on the development of an interactive procedure that implemented the algorithm proposed in Phase I and relied on the progressive definition of DM's preferences along with the exploration of the criterion space, in order to reach the most favorable vehicle routes of multicriteria VRPs.

The research work of Phase I consisted of three sub-objectives. The first sub-objective was to construct a mathematical model of the multicriteria VRP in a GP framework and develop an algorithm to apply it to the VRPs in a multiple objective environment. Chapter IV described the development of a heuristic algorithm that consisted of the Cluster Method to partition a set of stations into subsets and the Iterative GP Procedure to sequence the stations in each subset. The algorithm was illustrated by a simple example. The proposed algorithm has the capability of treating the conflicting multiple objectives simultaneously.

The second sub-objective of Phase I was to develop a computer program of the proposed algorithm. Its programming was described in Chapter V. In particular, due to the shortcomings of the computer code available for integer GP, a Sequential Linear Goal Programming approach was adopted to solve a GP model at each interation in the routing procedure. The proposed algorithm was coded in FORTRAN. A list of the source program is included in Appendix A.

The third sub-objective of Phase I was to perform computational experiments of the proposed algorithm on three test problems incorporating multiple objectives, and evaluate its performance by comparing the results with those obtained by savings algorithms for VRPs with a single objective, with respect to some criteria corresponding to the multiple objectives. Chapter V presented the computational experience of the algorithm developed in this research. Three savings methods, Clarke and Wright's savings, multiple and sequential approaches, and Gaskell's savings, multiple (λ) approach, were selected for the comparsion. Based on solution optimality, the proposed algorithm produced the nondominated solution in all cases. The experiments showed that the outcomes of a test problem differed, depending upon the DM's preference regarding the goal priority structure. The computer times were difficult to contrast since the algorithms were programmed on different computers.

The research work of Phase II consisted of two sub-objectives. The first and second sub-objectives were to develop an interactive procedure and its computer program, respectively. Chapter VI discussed the design of the interactive procedure that implemented the algorithm proposed in Phase I and the use of its computer program. A test problem was used to execute the interactive program. In particular, through the change of target values and the DM's goal priority structure, it was shown that the proposed algorithm successfully performs the trade-off between the achievement levels of the objectives in a reasonable way.

The research results in this dissertation can be summarized as follows:

1. A heuristic algorithm was developed to determine the most

satisfactory vehicle routes of multicriteria VRPs where three objectives are to be achieved. The algorithm consists of a Cluster Method and an Iterative GP Procedure. It has the important capability of taking into account the DM's preference regarding the goal priority structure and the target values of the goal constraints. Therefore, it can provide the DM with the ability to make intelligent trade-off decisions about the different objectives.

2. Computational experiments showed that the proposed algorithm is capable of performing a trade-off between the achievement levels of the objectives, based on the DM's preference regarding the goal priority structure and the target values of the goal constraints. However, the shortcomings of the algorithm lie in the fact that more than one run is necessary to solve SLGP problems in the routing procedure. The resultant computation time and computer memory requirement can therefore be substantial.

3. An interactive procedure was developed to reach the most favorable vehicle routes of multicriteria VRPs where three objectives are to be achieved. It successfully performed the trade-off between the achievement levels of the objectives. The interactive procedure allows a DM not only to provide local preference information but also gain understanding and feeling for the behavior of the system. As such it will help the implementation of the proposed algorithm for multicriteria VRPs in practice.

Recommendations

The general procedure establised in this research provides a foundation on which more refined procedures could be developed. Some possible areas for future study are recommended below:

1. Extend the present model of multicriteria VRPs to include more possible objectives, such as the minimization of the violation of the specified service time (or day) requirements at stations, the minimization of number of visits to the customer when more than one visit to the customer is allowed to collect or deliver the commodity, the minimization of the sum of fixed and variable costs, etc.
2. Develop an algorithm for multicriteria VRP where demands or supplys at stations are probabilistic, the distance between stations are nonsymmetric, and/or the capacity of vehicles are different.
3. Develop an algorithm for multicriteria VRPs that is capable of searching for all of the nondominated solutions.
4. Implement IBM MIP (Mixed Integer Programming)/370 in solving the SLGP problems in the routing procedure of the proposed algorithm, which will make it possible to handle large-scale multicriteria VRPs.
5. Apply a computer graphic system to the interactive procedure developed, and help a DM to perceive visually the vehicles routes generated.

The recommendations listed above constitutes a new direction of research that may prove to have a great impact on the future use of vehicle routing models.

BIBLIOGRAPHY

(1) Aneja, Y. P., and K. P. K. Nair. "Bicriteria Transportation Problem." Management Science, Vol. 6, No. 1 (1979), pp.73-78.

(2) Baker, K. R. Introduction to Sequencing and Scheduling. John Wiley and Sons, Inc., New York, 1974, pp. 104-106.

(3) Balinski, M. L., and R. E. Quandt. "On an Integer Program for a Delivery Problem." Operations Research, Vol. 12 (1964), pp. 300-304.

(4) Bellman, R. "Dynamic Programming Treatment of the Traveling Salesman Problem." Journal of the Association for Computing Machinery, Vol. 9 (1962), pp. 61-63.

(5) Beltrami, E. J., and L. D. Bodin. "Networks and Vehicle Routing for Municipal Waste Collection." Networks, Vol. 4 (1974), pp. 65-94.

(6) Benson, R. G. "Interactive Multiple Criteria Optimization Using Satisfactory Goals." (Unpublished PH.D. Dissertation, University of Iowa, 1975.)

(7) Bodin, L. D., B. L. Golden, A. A. Assad, and M. O. Ball. "Special Issue: Routing and Scheduling of Vehicles and Crews." Computer and Operations Research, Vol. 10, No. 2 (1982), pp. 62-211.

(8) Bussey, L. E. The Economic Analysis of Industrial Projects. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

(9) Buxey, G. M. "The Vehicle Scheduling Problen and Monte Carlo Simulation." Journal of Operation Research Society, Vol. 30 (1979), pp. 563-573.

(10) Charnes, A., and W. W. Cooper. Management Models and Industrial Applications of Linear Programming. Vol. I, Wiley, New York, 1961. (Chap. 6, Appendix B, Basic Existence Theorems and Goal Programming.)

(11) Cheshire, I. M., A. M. Malleson, and P. F. Naccache. " A Dual Heuristic For Vehicle Scheduling." Journal of Operational Research Society, Vol. 33 (1982), pp. 51-61.

(12) Christofides, N. Mover Users Guide: Memo. Imperial College, London, 1979.

(13) Christofides, N., and S. Eilon. "An Algorithm for the Vehicle Dispatching Problem." Operational Research Quarterly, Vol. 20 (1969), pp. 309-318.

(14) Christofides, N., A. Mingozzi, and P. Toth. "The Vehicle Routing Problem." Combinatorial Optimisation, Wiley, 1979.

(15) Christofides, N., A. Mingozzi, and P. Toth. "Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations." Mathematical Programming, Vol. 20 (1981), pp. 255-282.

(16) Chvatal, V. Linear Programming. Freeman and Co., New York, 1983.

(17) Clarke, G., and J. Wright. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." Operations Research, Vol. 12 (1964), pp. 568-581.

(18) Cook. T. M., and R. A. Russell. "A Simulation and Statistical Analysis of Stochastic Vehicle Routing with Timing Constraints." Decision Sciences, Vol. 9 (1978), pp. 673-687.

(19) Dantzig, G. P., and J. H. Ramser. "The Truck Dispatching Problem." Management Science, Vol. 6 (1959), pp. 80-91.

(20) Doll, C. L. "Quick and Dirty Vehicle Routing Procedure." Interfaces, Vol. 10 (1980), pp. 84-85.

(21) Fisher, M. L., A. J. Greenfield, R. Jaikumar, and J. T. Lester III. "A Computerized Vehicle Routing Application." Interfaces, Vol. 12 (1982), pp. 42-52.

(22) Foster, B. A., D. M. Ryan. "An Integer Programming Approach to the Vehicle Scheduling Problem." Operational Research Quarterly, Vol. 27 (1976), pp. 367-384.

(23) Gaskell, T. J. "Bases for Vehicle Fleet Scheduling." Operational Research Quarterly, Vol. 18 (1967), pp. 281-295.

(24) Gillett, B. E., and L. R. Miller. "A Heuristic Algorithm for the Vehicle Dispatch Problem." Operations Research, Vol. 22 (1974), pp. 340-349.

(25) Goicoechea, A., D. R. Hansen., and L. Duckstein. Multi-objective Decision Analysis with Engineering and Business Applications. John Wiley and Sons, Inc., Toronto, 1982.

(26) Golden, B. L., T. L. Magnanti, and H. Q. Nguyen. "Implementing Vehicle Routing Algorithms." Networks, Vol. 7 (1977), pp. 113-148.

(27) Golden, B. L., and W. Stewart. "Vehicle Routing with Probabilistic Demands." Proceedings of the Tenth Annual Symp. on the Interface of Computer Science and Statistics, Gaithersburg, Maryland, April, 1977.

(28) Golden, B. L., and J. R. Yee. "A Framework for Probabilistic Vehicle Routing." AIIE Transactions, Vol. 11 (1979), pp. 109-112.

(29) Homes, R. A., and P. G. Parker. "A Vehicle Scheduling Procedure Based upon Savings and a Solution Perturbation Scheme." Operational Research Quarterly, Vol. 27 (1976), pp. 83-92.

(30) Hwang, C., and A. S. M. Masud. Multiple Objective Decision Making-Methods and Applications. Springer-Verlag, New York, 1979.

(31) Ignizio, J. P. Linear Programming in Single and Multiple Objective Systems. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

(32) Ignizio, J. P. Goal Programming and Extensions. Lexington Books, Massachusetts, 1976.

(33) Ignizio, J. P., and J. H. Perlis. "Sequential Linear Goal Programming: Implementation via MPSX." Computer and Operations Research, Vol. 6 (1979), pp. 141-145.

(34) Ijiri, Y. Management Goals and Accounting for Control. North-Holland, Amsterdam, 1965.

(35) Kirby, R. F., and J. J. McDonald. "Viewpoints: The Savings Method for Vehicle Scheduling." Operational Research Quarterly, Vol. 24, No. 2 (1973), pp. 305-307.

(36) Krolek, P., W. Flets, and J. Nelson. "A Man-Machine Approach Towards Solving the Generalized Truck Dispatching Problem." Transportation Science, Vol. 6, (1972), pp. 149-169.

(37) Kuester, J. L., and J. H. Mize. Optimization Techniques with FORTRAN. McGraw-Hill Book Co., 1973.

(38) McDonald, J. J. "Vehicle Scheduling-A Case Study." Operational Research Quarterly, Vol. 23 (1972), pp. 434-444.

(39) Miller, C., A. W. Tucker, and R. A. Zemlin. "Interger Programming Formulation of the Travelling Salesman Problem." Journal of the Association for Computing Machinery, Vol. 7 (1960), pp. 326-329.

(40) Mole, R. H. "A Survey of Local Delivery Vehicle Routing Methodology." Journal of Operational Research Society, Vol. 30 (1979), pp. 245-252.

(41) Mole, R. H., and S. R. Jameson. "A Sequential Route Building Algorithm Employing a Generalized Savings Criterion." Operational Research Quarterly, Vol. 27 (1976), pp. 503-511.

(42) Lee, S. M. Goal Programming for Decision Analysis. Auerbach Publishers, Philadelphia, Pennsylvania, 1972.

(43) Lee, S. M., and L. J. Moore. "Multi-Criteria School Busing Models." Management Science, Vol. 23, No. 7 (1977), pp. 703-715.

(44) Lin, S., and B. Kernighan. "An Effective Heuristic Algorithm for the Traveling Salesman Problem." Operations Research, Vol. 21 (1973), pp. 498-516.

(45) Little, J. D. C., K. G. Murty, D. W. Sweeney, and C. Karel. "An Algorithm for the Traveling Salesman Problem." Operations Research, Vol. 11 (1963), pp. 979-989.

(46) Park, Y. B., C. D. Pegden, and E. E. Enscore. "A Survey and Evaluation of Static Flowshop Scheduling Heuristics." International Journal of Production Research, Vol. 22, No. 1 (1984), pp. 127-141.

(47) Parkhideh, B. "The Economic Design of Dynamic $\overline{X}$-Control Charts." (Unpublished PH. D. Dissertation, Oklahoma State University, 1984.)

(48) Pierce, J. "Direct Search Algorithms for Truck Dispatching Problems. Part I." Transportation Research, Vol. 3 (1969), pp. 1-42.

(49) Robbins, J. A. "A Program for Solution of Large Scale Vehicle Routing Problems." (Unpublished M. S. Thesis, Oklahoma State University, 1976.)

(50) Robbins, J., J. Shamblin, W. Turner, and D. Byrd. "Development and Computational Experience with a Combination Tour Construction-Tour Improvement Algorithm for Vehicle Routing Problem." (Presented at the ORSA/TIMS meeting, Las Vegas, Nevada, Nov., 1975.)

(51) Robertson, W. C. "Route and Van Scheduling in the Newspaper Industry." (Presented to the O. R. Society Annual Conference, Edinburgh.)

(52) Russell, R. A. "An Effective Heuristic for the M-Tour Traveling Salesman Problem with Some Side Conditions." Operations Research, Vol. 25 (1977), pp. 517-524.

(53) Schilling, D. A., A. McGarity, and C. Revelle. "Hidden Attributes and The Display of Information in Multiobjective Analysis." Management Science, Vol. 28, No. 3 (1982), pp. 236-252.

(54) Sharif, M. N., and S. Agarwalla. "Sequential Approach to Multicriteria Optimization." Unknown.

(55) Taha, H. A. Integer Programming: Theory, Applications, and Computations. Academic Press, Inc., London, 1975.

(56) Tillman, F., and T. Cain. "An Upper Bounding Algorithm for the Single and Multiple Terminal Delivery Problem." Management Science, Vol. 18 (1972), pp. 664-682.

(57) Tillman, F. A., and R. D. Hering. "A Study of a Look Ahead Procedure for Solving the Multiple Delivery Problem." Transportation Research, Vol. 5 (1971), pp. 225-229.

(58) User's Manual, University Computer Center, Oklahoma State University, 1983.

(59) Watson, C. D. T., and L. R. Foulds. "The Vehicle Scheduling Problem." New Zealand Operational Research, Vol. 9 (1981), pp. 73-92.

(60) Webb, M. H. J. "Relative Performance of Some Sequential Method of Planning Multiple Delivery Journeys." Operational Research Quarterly, Vol. 23 (1972), pp. 361-372.

(61) Williams, B. W. "Vehicle Scheduling: Proximity Priority Searching." Journal of Operational Research Society, Vol. 33 (1982), pp. 961-966.

(62) Wren, A., and A. Holliday, "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Depots. Operational Research Quarterly, Vol. 23 (1972), pp. 333-344.

(63) Yellow, P. C. "A Computational Modification to the Savings Method of Vehicle Scheduling." Operational Research Quarterly, Vol. 21 (1970), pp. 281-283.

(64) Zeleny, M. Multiple Criteria Decision Making, McGraw-Hill, Inc., New York, 1982.

APPENDICES

APPENDIX A

FORTRAN PROGRAM LISTING

**** TSO FOREGROUND HARDCOPY ****
DSNAME=U14387A.INTER2.DATA

```
C*****************************************************************
C  THIS INTERACTIVE PROGRAM DETERMINES THE MOST FAVORABLE VEHICLE
C  ROUTES OF MULTICRITERIA VEHICLE ROUTING PROBLEM (VRP) , WITH
C  RESPECT TO THE DECISION MAKER'S PREFERENCE.
C
C  BY YANG BYUNG PARK, SCHOOL OF INDUSTRIAL ENGINEERING AND
C                     MANAGEMENT
C  DISSERTATION ADVISER: DR. C. PATRICK KOELLING
C
C*****************************************************************
C
C  FUNCTION OF SUBROUTINES
C
C    SUBROUTINE                  FUNCTION
C    ----------                  -------
C      SORT1          SORTS STATIONS ABOUT A STATION IN INCREASING
C                     ORDER
C      SORT2          SORTS STATIONS ABOUT THE DEPOT IN DECREASING
C                     ORDER
C      LONG           SEARCHES FOR THE FURTHEST UNASSIGNED STATION
C                     FROM THE DEPOT
C      SFEA1          SEARCHES FOR THE CLOSEST FEASIBLE STATION TO
C                     AN END
C      SFEA2          SEARCHES FOR THE CLOSEST FEASIBLE STATION TO
C                     OTHER END
C      CRT            DETERMINES THE STATION TO BE ASSIGNED TO A LINK
C      PCASE1         SLGP SUBROUTINE BASED ON THE GOAL PRIORITY
C                     STRUCTURE MODEL I
C      PCASE2         SLGP SUBROUTINE BASED ON THE GOAL PRIORITY
C                     STRUCTURE MODEL II
C      PCASE3         SLGP SUBROUTINE BASED ON THE GOAL PRIORITY
C                     STRUCTURE MODEL III
C      SMINT          SUBROUTINE FOR MIXED INTEGER LINEAR PROGRAMMING
C      COMPT          COMPUTES THE VALUE OF EACH OBJECTIVE FOR THE
C                     ROUTE SEQUENCE GENERATED
C*******************************************************************
C
C  DEFINITION OF VARIABLES
C
C    MSTOP:    # OF STATIONS TO SERVE IN MULTICRITERIA VRP
C    MSTA:     # OF STATIONS TO SERVE INCLUDING A DEPOT IN MULTICRITERIA
C              VRP
C    MSTOPG:   # OF STATIONS IN A ROUTE, EXCLUDING A DEPOT
C    MSTAG:    # OF STATIONS IN A ROUTE, INCLUDING A DEPOT
C    NOEM:     # OF EMERGENT SERVICES REQUIRED
C    NOCON:    # OF CONDITIONAL DEPENDENCIES REQUIRED
C    NEMCI:    # OF EMERGENT SERVICES REQUIRED IN SUBSET(CLUSTER) I
C    NCOCI:    # OF CONDITIONAL DEPENDENCIES REQUIRED IN SUBSET I
C    MDISL:    MAX. ALLOWABLE TRAVEL DISTANCE OF VEHICLES
C    JPSL:     PREDETERMINED LEVEL OF DURATION FOR GOODS DETERIORATION
C    ALPHA:    SHAPE PARAMETER IN CLUSTERING
C    DAVG:     AVERAGE DISTANCE FROM A DEPOT TO STATION
C    IEND1:    AN END OF A LINK
C    IEND2:    OTHER END OF A LINK
C    JROW:     # OF SUBSETS CLUSTERED
C    JPSLG:    TARGET VALUE FOR GOODS DETERIORATION
C    MDISLG:   TARGET VALUE FOR VEHICLE TRAVEL DISTANCE
C    NMAX:     # OF DECISION VARIABLES IN SLGP
C    MMAX:     # OF CONSTRAINTS INCLUDING AN OBJECTIVE FUNCTION IN SLGP
C    NZR1VR:   # OF INTEGER DECISION VARIABLES IN SLGP
C    NGPS:     GOAL PRIORITY STRUCTURE OPTION NO.
C    TT:       VEHICLE TRAVEL DISTANCE ON A ROUTE
```

```
C     TD:       TOTAL DETERIORATION ON A ROUTE
C     FR:       TOTAL FULFILLMENT OF SERVICE REQUIREMENTS ON A ROUTE
C     ISUMTT:   GRAND TOTAL TT
C     ISUMTD:   GRAND TOTAL TD
C     ISUMFR:   GRAND TOTAL FR
C     SOLMIN:   UPPER LIMIT OF OBJECTIVE FUNCTION
C     NCSM:     # OF CALLS FOR SUBROUTINE MINT
C     JHANG:    1 IF AN INFINITE LOOP IS GENERATED IN MINT ALGORITHM
C               0 OTHERWISE
C     MZOPT:    OPTIMAL VALUE OF AN OBJECTIVE FUNCTION
C     INEXT:    NEXT STATION TO STOP DETERMINED IN THE ROUTING PROCEDURE
C     IRTR:     1 IF A VIOLATION OF A RESTRICTION IS DISCOVERED
C               0 OTHERWISE
C
C   DEFINITION OF ARRAYS
C
C     MX(I):    X COORDINATE OF STATION I
C     MY(I):    Y COORDINATE OF STATION I
C     MP(I):    1 IF STATION I IS CLUSTERED
C               0 OTHERWISE
C     MSUP(I): QUANTITY OF SUPPLY AT STATION I
C     MATX(I,J): DISTANCE MATRIX OF STATIONS
C     MDIS(I,J): DISTANCE BETWEEN STATIONS I AND J
C     MCL(I):   PSEUDO-ASSIGNED STATIONS TO BOTH ENDS IN THE CLUSTERING
C               PROCEDURE
C     ICLUST(I): STATIONS CLUSTERED INTO SUBSET I
C     LOAD(I): VEHICLE LOAD ON SUBSET I
C     MEX(I):   STATIONS REQUIRING EMERGENT SERVICE ON SUBSET I
C     MEY(I):   STATIONS REQUIRING CONDITIONAL DEPENDENCY ON SUBSET I
C     TTAB(I,J): ARRAY TABLEAU FOR SLGP
C     ATAB(I,J): COPIED ARRAY TABLEAU FOR SLGP
C     UPBND(I): UPPER BOUND OF DECISION VARIABLE I
C     BAS(I):   FUNCTION CRT VALUE OF STATION I
C     IROW(I): VECTOR OF CONSTRAINT TYPE I
C     T(I):     VALUE OF DECISION VARIABLE I IN AN OPTIMAL SOLUTION
C     MEND(I): ENDS OF A LINK IN THE CLUSTERING PROCEDURE
C     AEMEG(I): STATIONS REQUIRING EMERGENT SERVICE
C     ACOND(I): STATIONS REQUIRING CONDITIONAL DEPENDENCY
C     IBB(I):   ARRAY FOR TT, TD, FR, AND A ROUTE SEQUENCE
C     IBB(I,J): ARRAY FOR TT, TD, FR, AND A ROUTE SEQUENCE OF SUBSET I
C
C*****************************************************************
C
C
C*****************************************************************
C   MAIN PROGRAM
C     IT CONSTRUCTS AN INITIAL INPUT DATA ARRAY OF SYSTEM CONSTRAINTS
C     FOR ITERATIVE SLGP PROCEDURE, CALL AN APPROPRIATE SLGP SUBROUTINE
C     BASED ON THE DM'S GOAL PRIORITY STRUCTURE, AND DETERMINES THE
C     MOST FAVORABLE VEHICLE ROUTES THROUGH CHANGING TARGET VALUES OF
C     CONSTRAINTS AND/OR EXCHANGING STATIONS IN SUBSETS.
C*****************************************************************
      DOUBLE PRECISION TTAB(65,70),UPBND(70)
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER2/ MCL(4),MEND(4),MSUP(101),MQ,ILOD,IDIS
      COMMON/USER3/ MATX(99,99)
      COMMON/USER4/ NDEP(100)
      COMMON/USER5/ ANGLE(100),ALPHA,DAVG
      COMMON/USER6/ MSTOPG,MSTAG,MDISL,JPSLG,NEMCI,NCOCI,IROWG,JPSLGG
      COMMON/USER7/ NMAX,MMAX,MSCO,IBB(20)
      COMMON/USER8/ NZR1VR,ISIZE,IOUT1,IOUT2,IOUT3,M,N,IROW(65),KKNG
      COMMON/USER10/ UPBND
      DIMENSION MX(101),MY(101),AEMEG(10),ACOND(10,2),LOADI(20)
     *,IBBALL(20,20),NUMST(10)
      INTEGER ZFIN,AEMEG,ACOND
```

```
C  READ INPUT DATA
      READ(9,10) MSTOP,MCAPL,MDISL,JPSL
   10 FORMAT(4I10)
      MSTA=MSTOP+1
      READ(9,18) NOEM,NOCON
   18 FORMAT(2I10)
      IF(NOEM.EQ.0) GO TO 7
      READ(9,17) (AEMEG(I),I=1,NOEM)
   17 FORMAT(10I5)
   7  IF(NOCON.EQ.0) GO TO 8
      DO 11 I=1,NOCON
      READ(9,12) ACOND(I,1),ACOND(I,2)
   12 FORMAT(2I5)
   11 CONTINUE
   8  DO 20 I=1,MSTA
      READ(9,25) MX(I),MY(I),MSUP(I)
   25 FORMAT(3I10)
   20 CONTINUE
C  TARGET VALUE FOR TD IS INITIALLY SET EQUAL TO THE PREDETERMINED
C  LEVEL FOR GOODS DETERIORATION
      JPSLGG=JPSL
  303 ISUMTT=0
      ISUMTD=0
      ISUMFR=0
C  DETERMINE GOAL PRIORITY STRUCTURE
      WRITE(6,4)
   4  FORMAT(//,T2,'===> GOAL PRI. MENU <===',/,T2,'ENTER OPTION NO.',
     */,T5,'1: TRAVEL DIST.=1, DETERIORATION=2, FULFILLMENT',
     *' OF SERVICE REQ.=3',
     */,T5,'2: TRAVEL DIST.=2, DETERIORATION=1, FULFILLMENT',
     *' OF SERVICE REQ.=3',
     */,T5,'3: TRAVEL DIST.=1, DETERIORATION=3, FULFILLMENT',
     *' OF SERVICE REQ.=2')
      READ(5,*) NGPS
  509 IF(NGPS.EQ.2) MDISL4=JPSLGG
      IF(NGPS.NE.2) MDISL4=MDISL
      WRITE(6,30) MSTOP,MCAPL,MDISL,NOEM,NOCON,JPSL
   30 FORMAT(/,T2,'THE INPUT DATA GIVEN ARE SUMMARIZED AS',
     *' FOLLOWS:',/,T5,'NO. OF STATIONS=',I5,/,T5,'LIMIT OF VEHICLE',
     *' CAPACITY=',I5,/,T5,'MAX. ALLOWABLE VEHICLE TRAVEL',
     *' DISTANCE =',I5,/,T5,
     *'NO. OF TOTAL EMERG. SERV. REQ.=',I3,/,T5,'NO. OF TOTAL COND. DEP'
     *,' OF STATIONS=',I3,/,T5,'PREDETERMINED LEVEL OF DISTANCE',
     *' FOR DETERIORATION=',I5,/,T5,'UPPER LEVEL OF DISTANCE',
     *' FOR THE COMPLETE DETERI.=',I5)
      WRITE(6,92) (AEMEG(I),I=1,NOEM)
   92 FORMAT(T5,'STATIONS REQUIRING EMERG. SERV.=',10I4)
      IF(NOCON.EQ.0) GO TO 93
      WRITE(6,94) ((ACOND(I,J),J=1,2),I=1,NOCON)
   94 FORMAT(T5,'CONDITIONALLY DEPEN. STAT.=',2X,10('(',I2,',',I2,')'
     *,1X))
C  DETERMINE THE ALPHA VALUE IN FUNCTION CRT(I)
   93 WRITE(6,5)
   5  FORMAT(/,T2,'===> ENTER ALPHA VALUE FOR CLUSTERING <===')
      READ(5,*) ALPHA
      WRITE(6,6) ALPHA
   6  FORMAT(T2,'ALPHA VALUE ENTERED IS:',F5.2)
      JROW=0
C  COMPUTE A DISTANCE MATRIX
      DO 35 I=1,MSTA
      DO 35 J=1,MSTA
      IF(I.EQ.J) MDIS(I,J)=0
      IF(I.GE.J) GO TO 35
      WOO=FLOAT((MX(I)-MX(J))**2+(MY(I)-MY(J))**2)
      MDIS(I,J)=SQRT(WOO)
      MDIS(J,I)=MDIS(I,J)
```

```
   35  CONTINUE
C   SORT STATIONS ABOUT A STATION IN INCREASING ORDER
       CALL SORT1
C   SORT STATIONS ABOUT THE DEPOT IN DECREASING ORDER
       CALL SORT2
       DO 31 I=1,MSTOP
       MP(I)=0
   31  CONTINUE
       DO 32 I=1,20
       DO 32 J=1,10
       ICLUST(I,J)=0
   32  CONTINUE
C   COMPUTE THE AVERAGE DISTANCE FROM A DEPOT TO STATION
       ITOT=0
       DO 33 I=1,MSTOP
       ITOT=ITOT+MDIS(MSTA,I)
   33  CONTINUE
       DAVG=FLOAT(ITOT)/FLOAT(MSTA)
       SOS=1.
       IF(SOS.EQ.O.) GO TO 61
       WRITE(6,40)
   40  FORMAT(//,T2,'** THE DISTANCE MATRIX')
       DO 60 I=1,MSTA
       WRITE(6,65) (MDIS(I,J),J=1,MSTA)
   65  FORMAT(1X,26I4)
   60  CONTINUE
   61  DO 62 I=1,MSTA
       DO 62 J=1,MSTOP
       IF(I.EQ.J) GO TO 62
       MDIS(I,J)=MDIS(I,J)+10
   62  CONTINUE
C   COMPUTE ANGLES OF STATIONS
       DO 70 I=1,MSTOP
       GAMES=FLOAT(MX(I)-MX(MSTA))
       IF(GAMES.EQ.O.) GAMES=0.0001
       CBS=(FLOAT(MY(I)-MY(MSTA)))/GAMES
       ANGLE(I)=ATAN(CBS)
   70  CONTINUE
C   SEARCH FOR THE FURTHEST UNASSIGNED STATION FROM THE DEPOT
  100  CALL LONG(IFUS)
       IF(IFUS.EQ.0) GO TO 115
       MP(IFUS)=1
       ILOD=MSUP(IFUS)
       IDIS=MDIS(MSTA,IFUS)+MDIS(IFUS,MSTA)
       JROW=JROW+1
       JCOL=1
C   ASSIGN STATION IFUS TO SUBSET JROW
       ICLUST(JROW,JCOL)=IFUS
       IEND1=IFUS
       IEND2=IEND1
C   SEARCH FOR THE CLOSEST FEASIBLE STATIONS TO AN END
   90  CALL SFEA1(IEND1,MDISL4,MCAPL,ZFIN)
       IF(IEND2.EQ.IEND1) GO TO 75
C   SEARCH FOR THE CLOSEST FEASIBLE STATIONS TO ANOTHER END
       CALL SFEA2(IEND2,MDISL4,MCAPL,ZFIN)
   75  IF(MQ.EQ.0) GO TO 95
C   DETERMINE THE STATION TO BE ASSIGNED TO A ROUTE(SUBSET)
       CALL CRT(LINK)
       LAST=MEND(LINK)
       MEW=MCL(LINK)
       ILOD=ILOD+MSUP(MEW)
       IDIS=IDIS-MDIS(LAST,MSTA)+MDIS(LAST,MEW)+MDIS(MEW,MSTA)
       JCOL=JCOL+1
       ICLUST(JROW,JCOL)=MEW
       MP(MEW)=1
       IF(IEND1.EQ.LAST) GO TO 80
```

```
      IEND2=MEW
      GO TO 90
 80   IEND1=MEW
      GO TO 90
 95   IF(ZFIN.EQ.O) GO TO 115
      LOADI(JROW)=ILOD
      GO TO 100
 115  LOADI(JROW)=ILOD
 401  ISUMTT=O
      ISUMTD=O
      ISUMFR=O
      WRITE(6,105)
 105  FORMAT(/,T2,'** THE CLUSTERED SUBSETS')
      DO 110 I=1,JROW
      WRITE(6,120) (ICLUST(I,J),J=1,10)
 120  FORMAT(T5,10I4)
 110  CONTINUE
C  APPLICATION OF ITERATIVE SLGP HEURISTIC ALGO. TO EACH CLUSTER
      DO 99 IROWG=1,JROW
C  DETERMINE # OF STATIONS IN SUBSET IROWG
      ICOLG=O
      DO 149 J=1,10
      IF(ICLUST(IROWG,J).EQ.O) GO TO 152
      ICOLG=ICOLG+1
 149  CONTINUE
 152  MSTOPG=ICOLG
      MSTAG=MSTOPG+1
      NUMST(IROWG)=MSTOPG
      JPSLG=JPSL
      WRITE(6,43) IROWG,(ICLUST(IROWG,J),J=1,MSTOPG)
 43   FORMAT(//,T7,'** ITERATIVE SLGP APPL. TO CLUSTER',I3,/,T5,10I4)
      WRITE(6,44) LOADI(IROWG)
 44   FORMAT(T5,'A VEHICLE LOAD:',I6)
C  DETERMINATION OF EMER. SERV. AT CLUSTER IROWG
      NEMCI=O
      DO 200 I=1,MSTOPG
      KP=ICLUST(IROWG,I)
      DO 205 J=1,NOEM
      KQ=AEMEG(J)
      IF(KP.NE.KQ) GO TO 205
      NEMCI=NEMCI+1
      MEX(NEMCI)=KQ
      MXX(NEMCI)=MSTOPG*MSTOPG+I
      GO TO 200
 205  CONTINUE
 200  CONTINUE
      IF(NEMCI.GE.1) WRITE(6,210) (MEX(I),I=1,NEMCI)
      WRITE(6,201) NEMCI
 201  FORMAT(T5,'NO. OF EMERG. SERV. REQ.=',I2)
 210  FORMAT(T5,'STATIONS FOR EMERG. SERV.:',10I4)
C  DETERMINATION OF CON. DEP. STATIONS
      NCOCI=O
      DO 211 I=1,NOCON
      KP=ACOND(I,1)
      DO 212 J=1,MSTOPG
      KQ=ICLUST(IROWG,J)
      JJ=J
      IF(KP.EQ.KQ) GO TO 213
 212  CONTINUE
      GO TO 211
 213  KR=ACOND(I,2)
      DO 214 L=1,MSTOPG
      KQ=ICLUST(IROWG,L)
      LL=L
      IF(L.GT.J) LL=LL-1
      IF(KR.EQ.KQ) GO TO 216
```

```
  214 CONTINUE
      GO TO 211
  216 NCOCI=NCOCI+1
      MEY(NCOCI,1)=ACOND(I,1)
      MEY(NCOCI,2)=ACOND(I,2)
      MYY(NCOCI)=MSTOPG*(JJ-1)+LL
  211 CONTINUE
      IF(NCOCI.GE.1) WRITE(6,202) ((MEY(I,J),J=1,2),I=1,NCOCI)
      WRITE(6,203) NCOCI
  202 FORMAT(T5,'COND. DEP. STA.:',10('(',I3,',',I3,')',1X))
  203 FORMAT(T5,'NO. OF COND. DEP. STA.=',I2)
C  CONSTRUCT AN INITIAL INPUT DATA ARRAY OF SYSTEMS CONST. FOR
C  ITERATIVE SLGP ALGORITHM
C  DETERMINE # OF DECISION VARIABLES AND THE MAX. # OF CONSTRAINTS
C  INCLUDING AN OBJECTIVE FUNCTION IN SLGP TO BE RUN
  355 NMAX=MSTAG*MSTOPG+MSTOPG+1
      MMAX=2*MSTAG+MSTOPG*(MSTOPG-1)+3
      MSCO=MMAX-2
C  DETERMINE THE ALL CONSTANT INPUT DATA
      NZR1VR=MSTAG*MSTOPG
      ISIZE=NZR1VR*(2*NMAX-NZR1VR+1)/2+200
      IOUT1=0
      IOUT2=0
      IOUT3=0
C  UPPER BOUNDS OF ALL VARIABLES
      KA=NZR1VR+MSTOPG
      DO 22 I=1,NZR1VR
   22 UPBND(I)=1.0
      KG=NZR1VR+1
      DO 23 I=KG,KA
   23 UPBND(I)=20.0
      DO 220 I=1,MMAX
      DO 220 J=1,NMAX
  220 TTAB(I,J)=0.0
C  RIGHT HAND SIDE(RHS) OF EQ. (1)-(3)
      KA=2*MSTAG+1
      DO 225 I=2,KA
  225 TTAB(I,1)=1.0
      KA=KA+1
      DO 230 I=KA,MSCO
  230 TTAB(I,1)=FLOAT(MSTOPG)
      LQR=MSTAG+1
      JP=1
C  COEFF. OF EQ. (1)
      DO 235 I=2,LQR
      DO 235 J=1,MSTOPG
      JP=JP+1
      TTAB(I,JP)=1.0
  235 CONTINUE
C  COEFF. OF EQ. (2)
      MM=MSTOPG-1
      DO 240 I=1,MM
      KA=I+MSTAG+1
      ITI=I+1
      TTAB(KA,ITI)=1.0
      DO 245 J=2,MSTOPG
      IF(I.EQ.(J-1)) ITI=ITI+MSTAG
      ITI=ITI+MSTOPG
      TTAB(KA,ITI)=1.0
  245 CONTINUE
  240 CONTINUE
      DO 250 I=MSTOPG,MSTAG
      KA=KA+1
      ITI=I+1
      DO 255 J=1,MSTOPG
      TTAB(KA,ITI)=1.0
```

```
      ITI=ITI+MSTOPG
  255 CONTINUE
  250 CONTINUE
C  COEFF. OF EQ. (3)
      JAL=MSTAG*MSTOPG+2
      KAL=JAL
      NAL=JAL
      MM=MSTOPG-1
      IX=1
      DO 260 I=1,MSTOPG
      DO 265 J=1,MM
      KA=KA+1
      IX=IX+1
      TTAB(KA,IX)=FLOAT(MSTAG)
      TTAB(KA,JAL)=1.0
      IF(JAL.EQ.KAL) KAL=KAL+1
      TTAB(KA,KAL)=-1.0
      KAL=KAL+1
  265 CONTINUE
      IX=IX+1
      KAL=NAL
      JAL=JAL+1
  260 CONTINUE
C  COEFF. OF EQ. (4)
      KA=KA+1
      ICLUST(IROWG,MSTAG)=MSTA
      IX=1
      DO 268 NP=1,MSTAG
      KF=ICLUST(IROWG,NP)
      DO 270 NQ=1,MSTAG
      IF(NQ.EQ.NP) GO TO 270
      KG=ICLUST(IROWG,NQ)
      IX=IX+1
      TTAB(KA,IX)=FLOAT(MDIS(KF,KG))
  270 CONTINUE
  268 CONTINUE
      ICLUST(IROWG,MSTAG)=0
C  COEFF. OF EQ. (5)
      KA=KA+1
      LQR=MSTAG*MSTOPG
      DO 275 I=1,LQR
      II=I+1
      TTAB(KA,II)=TTAB(KA-1,II)
  275 CONTINUE
C  CHECK THE GOAL PRIORITY STRUCTURE AND CALL AN APPRORIATE SUBROUTINE
      IF(NGPS.EQ.1) CALL PCASE1(TTAB,JRTR,NPASS)
      IF(NGPS.EQ.2) CALL PCASE2(TTAB,JRTR,NPASS)
      IF(NGPS.EQ.3) CALL PCASE3(TTAB,JRTR,NPASS)
      IF(JRTR.EQ.1) GO TO 390
      IF(NPASS.EQ.1) GO TO 606
  304 WRITE(6,309)
  309 FORMAT(/,T2,'DO YOU WANT TO CHANGE TARGET VALUE FOR TT?',/,
     *T2,'===> ENTER OPTION NUMBER <===',/,T5,'1:YES   2:NO')
      READ(5,*) IOPT
      IF(IOPT.EQ.1) GO TO 355
  606 DO 315 I=1,20
      IBBALL(IROWG,I)=IBB(I)
  315 CONTINUE
C  COMPUTE THE SUM FOR EACH OBJ. FN.
      ISUMTT=ISUMTT+IBB(MSTAG+2)
      ISUMTD=ISUMTD+IBB(MSTAG+3)
      ISUMFR=ISUMFR+IBB(MSTAG+4)
   99 CONTINUE
      WRITE(6,351)
  351 FORMAT(///,T5,'** ROUTING IS COMPLETED FOR ALL CLUSTERS',/,T9,
     *' AND A COMPLETE SOLUTION IS OBTAINED AS FOLLOWS:')
```

```
      WRITE(6,314) ISUMTT,ISUMTD,ISUMFR
  314 FORMAT(T5,'TOT. TRAVEL DIST.=',I5,/,T5,'TOT. DETERIORATION=',
     *I5,/,T5,'TOT. FULL. OF SERVICE REQ.=',I3)
      DO 353 I=1,JROW
      IHH=NUMST(I)+2
      WRITE(6,399) LOADI(I),IBBALL(I,IHH+1),IBBALL(I,IHH+2),
     *IBBALL(I,IHH+3),(IBBALL(I,J),J=1,IHH)
  399 FORMAT(T5,'VEH. LOAD=',I5,' TT=',I5,' TD=',I3,' FR=',I2,
     *' ROUTE SEQ.=',20I3)
  353 CONTINUE
      IF(NGPS.NE.2) GO TO 376
      WRITE(6,504)
  504 FORMAT(/,T2,'DO YOU WANT TO CHANGE TARGET VALUE FOR TD?',/,
     *T2,'===> ENTER OPTION NUMBER <===',/,T5,'1:YES   2:NO')
      READ(5,*) IOPT
      IF(IOPT.EQ.2) GO TO 376
      WRITE(6,507) JPSL,JPSLGG
  507 FORMAT(/,T5,'** PREDETERMINED LEVEL OF DISTANCE FOR',
     *' DETERIORATION IS:',I5,/,
     *T5,'** CURRENT TARGET VALUE FOR THE DETERI. CONSTRAINT IS:',
     *I5,//,T5,
     *'ENTER NEW TARGET VALUE FOR THE DETERI. CONSTRAINT',
     *' BASED ON THE INFORMATION GIVEN ABOVE.')
      READ(5,*) JPSLGG
      WRITE(6,511) JPSLGG
  511 FORMAT(/,T2,'NEW TARGET VALUE FOR TD IS:',I5)
      GO TO 509
  376 WRITE(6,357)
  357 FORMAT(/,T2,'DO YOU WANT TO EXCHANGE STATIONS AMONG CLUSTERS?',/,
     *T2,'===> ENTER OPTION NUMBER <===',/,T5,'1:YES   2:NO')
      READ(5,*) IOPT
      IF(IOPT.EQ.2) GO TO 381
  390 WRITE(6,363)
  363 FORMAT(T5,'ENTER ONE CLUSTER NO., ITS STATION NO. AND THE OTHER',
     *' CLUSTER NO., ITS STATION NO., FOR EXCHANGE OF STATIONS')
      READ(5,*) JCLN1,JSTN1,JCLN2,JSTN2
      LOADT1=LOADI(JCLN1)-MSUP(JSTN1)+MSUP(JSTN2)
      LOADT2=LOADI(JCLN2)-MSUP(JSTN2)+MSUP(JSTN1)
      IF(LOADT1.GT.MCAPL.OR.LOADT2.GT.MCAPL) GO TO 412
      WRITE(6,365) JSTN1,JCLN1,JSTN2,JCLN2
  365 FORMAT(/,T2,'EXCHANGED STATIONS ARE:',/,T5,'STATION NO.',I3,
     *' IN CLUSTER NO.',I3,' AND STATION NO.',I3,' IN CLUSTER NO.',I3)
C  EXCHANGE THE STATIONS IN TWO CLUSTERS
      DO 367 I=1,10
      KP=ICLUST(JCLN1,I)
      IF(KP.EQ.0) GO TO 373
      IF(KP.EQ.JSTN1) GO TO 369
  367 CONTINUE
  369 ICLUST(JCLN1,I)=JSTN2
      LOADI(JCLN1)=LOADI(JCLN1)-MSUP(JSTN1)+MSUP(JSTN2)
      DO 371 I=1,10
      KP=ICLUST(JCLN2,I)
      IF(KP.EQ.0) GO TO 373
      IF(KP.EQ.JSTN2) GO TO 375
  371 CONTINUE
  373 WRITE(6,374)
  374 FORMAT(T2,'!ERROR!, CHECK INPUT DATA!!')
      GO TO 376
  375 ICLUST(JCLN2,I)=JSTN1
      LOADI(JCLN2)=LOADI(JCLN2)-MSUP(JSTN2)+MSUP(JSTN1)
      WRITE(6,387)
  387 FORMAT(T2,'DO YOU WANT TO CONTINUE TO EXCHANGE STATIONS',
     *' AMONG CLUSTERS?',/,T2,'===> ENTER OPTION NUMBER <===',/,T5,
     *'1:YES   2:NO')
      READ(5,*) IOPT
      IF(IOPT.EQ.1) GO TO 390
```

```
      GO TO 401
  412 WRITE(6,414)
  414 FORMAT(T2,'!ERROR! VEH. CAPACITY RESTRICTION IS VIOLATED!!',
     *' DO IT AGAIN!')
      GO TO 376
C  INQUIRY REGARDING GOAL PRIORITY STRUCTURE CHANGE
  381 WRITE(6,403)
  403 FORMAT(//,T2,'DO YOU WANT TO CHANGE THE GOAL PRIORITY STRUCTURE?',
     */,T2,'===> ENTER OPTION NUMBER <===',/,T5,'1:YES    2:NO')
      READ(5,*) IOPT
      IF(IOPT.EQ.1) GO TO 303
C  THE END OF THE INTERACTIVE PROCEDURE
      WRITE(6,407)
  407 FORMAT(T2,'*** THE MOST FAVORABLE VEHICLE ROUTE SEQUENCES ARE',
     *' DETERMINED',/,T5,'WITH RESPECT TO THE DECISION MAKERS',
     *' PREFERENCE')
      STOP
      END
C
C
C
      SUBROUTINE SORT1
C*****************************************************************
C     IT SORTS STATIONS ABOUT A STATION IN INCREASING ORDER.
C*****************************************************************
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER3/ MATX(99,99)
      DIMENSION NDIS(101,101)
      INTEGER FRONT,BIG,AMIN
C  COPY THE DISTANCE MATRIX TO NDIS(I,J)
      DO 10 I=1,MSTA
      DO 10 J=1,MSTA
      NDIS(I,J)=MDIS(I,J)
   10 CONTINUE
      BIG=9999999
      DO 20 I=1,MSTOP
      FRONT=1
   30 AMIN=BIG
      DO 40 J=1,MSTOP
      IF(J.EQ.I) GO TO 40
      IF(NDIS(I,J).GE.AMIN) GO TO 40
      AMIN=NDIS(I,J)
      LL=J
   40 CONTINUE
      NDIS(I,LL)=BIG
      MATX(I,FRONT)=LL
      FRONT=FRONT+1
      IF(FRONT.LT.MSTOP) GO TO 30
   20 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SORT2
C*****************************************************************
C     IT SORTS STATIONS ABOUT A DEPOT IN DECREASING ORDER.
C*****************************************************************
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER4/ NDEP(100)
      DIMENSION LDIS(100)
      INTEGER FRONT,SMALL,AMAX
      DO 10 I=1,MSTOP
      LDIS(I)=MDIS(MSTA,I)
```

```
   10 CONTINUE
      FRONT=1
      SMALL=-99
   30 AMAX=SMALL
      DO 20 I=1,MSTOP
      IF(LDIS(I).LE.AMAX) GO TO 20
      AMAX=LDIS(I)
      LL=I
   20 CONTINUE
      LDIS(LL)=SMALL
      NDEP(FRONT)=LL
      FRONT=FRONT+1
      IF(FRONT.LE.MSTOP) GO TO 30
      RETURN
      END
C
C
C
      SUBROUTINE LONG(JFUS)
C******************************************************************
C    IT SEARCHES FOR THE FURTHEST UNASSIGNED STATION FROM THE DEPOT.
C******************************************************************
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER4/ NDEP(100)
      JFUS=0
      DO 10 I=1,MSTOP
      IW=NDEP(I)
C  IF STATION IW HAS BEEN ALREADY ASSIGNED, GO TO 10
      IF(MP(IW).EQ.1) GO TO 10
      JFUS=IW
      GO TO 20
   10 CONTINUE
   20 RETURN
      END
C
C
C
      SUBROUTINE SFEA1(JEND1,NDISL,NCAPL,FIN)
C*****************************************************************
C    IT SEARCHES FOR THE CLOSEST FEASIBLE STATION(S) TO AN END.
C*****************************************************************
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER2/ MCL(4),MEND(4),MSUP(101),MQ,ILOD,IDIS
      COMMON/USER3/ MATX(99,99)
      INTEGER FIN
      MQ=0
      FIN=0
      NN=MSTOP-1
      JDIS=IDIS
      JLOD=ILOD
      DO 10 I=1,NN
      KG=MATX(JEND1,I)
C  IF STATION KG HAS BEEN ALREADY ASSIGNED, GO TO 10
      IF(MP(KG).EQ.1) GO TO 10
      FIN=1
C  PERFORM A FEASIBILITY TEST REGARDING DISTANCE AND CAPACITY
      JDIS=JDIS-MDIS(JEND1,MSTA)+MDIS(JEND1,KG)+MDIS(KG,MSTA)
      IF(JDIS.GT.NDISL) GO TO 20
      JLOD=JLOD+MSUP(KG)
      IF(JLOD.GT.NCAPL) GO TO 10
      MQ=MQ+1
      MCL(MQ)=KG
      MEND(MQ)=JEND1
      IF(MQ.EQ.2) GO TO 20
```

```
      JDIS=IDIS
      JLOD=ILOD
   10 CONTINUE
   20 RETURN
      END
C
C
C
      SUBROUTINE SFEA2(JEND2,NDISL,NCAPL,FIN)
C***************************************************************
C     IT SEARCHES FOR THE CLOSEST FEASIBLE STATION(S) TO OTHER END.
C***************************************************************
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER2/ MCL(4),MEND(4),MSUP(101),MQ,ILOD,IDIS
      COMMON/USER3/ MATX(99,99)
      INTEGER FIN
      MQL=MQ+2
      NN=MSTOP-1
      JDIS=IDIS
      JLOD=ILOD
      DO 10 I=1,NN
      KG=MATX(JEND2,I)
C  IF STATION KG HAS BEEN ALREADY ASSIGNED, GO TO 10
      IF(MP(KG).EQ.1) GO TO 10
      FIN=1
C  PERFORM A FEASIBILITY TEST REGARDING DISTANCE AND CAPACITY
      JDIS=JDIS-MDIS(JEND2,MSTA)+MDIS(JEND2,KG)+MDIS(KG,MSTA)
      IF(JDIS.GT.NDISL) GO TO 20
      JLOD=JLOD+MSUP(KG)
      IF(JLOD.GT.NCAPL) GO TO 10
      MQ=MQ+1
      MCL(MQ)=KG
      MEND(MQ)=JEND2
      IF(MQ.EQ.MQL) GO TO 20
      JDIS=IDIS
      JLOD=ILOD
   10 CONTINUE
   20 RETURN
      END
C
C
C
      SUBROUTINE CRT(NINK)
C***************************************************************
C     IT DETERMINES THE STATION TO BE ASSIGNED TO A LINK.
C***************************************************************
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER2/ MCL(4),MEND(4),MSUP(101),MQ,ILOD,IDIS
      COMMON/USER5/ ANGLE(100),ALPHA,DAVG
      DIMENSION BAS(4)
      INTEGER BEND
      SMALL=-99.0
C  COMPUTE THE VALUE OF CRT FUNCTION OF STATION I
      DO 10 I=1,MQ
      KG=MCL(I)
      BEND=MEND(I)
      DIF=ANGLE(KG)-ANGLE(BEND)
      IF(DIF.EQ.0.0) DIF=0.01
      BAS(I)=MDIS(MSTA,KG)+DAVG/(ABS(DIF)*ALPHA)
      IF(BAS(I).LE.SMALL) GO TO 10
      SMALL=BAS(I)
      NINK=I
   10 CONTINUE
      RETURN
```

```
      END
C
C
C
      SUBROUTINE PCASE1(TTAB,IRTR,NPASS)
C***************************************************************
C    IT IS FOR SLGP BASED ON THE GOAL PRIORITY STRUCTURE MODEL I.
C***************************************************************
      DOUBLE PRECISION DABS
      DOUBLE PRECISION TTAB(65,70),ATAB(65,70),T(70),UPBND(70)
      DOUBLE PRECISION ZOPT,PCTTOL,SOLMIN
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER6/ MSTOPG,MSTAG,MDISL,JPSLG,NEMCI,NCOCI,IROWG,JPSLGG
      COMMON/USER7/ NMAX,MMAX,MSCO,IBB(20)
      COMMON/USER8/ NZR1VR,ISIZE,IOUT1,IOUT2,IOUT3,M,N,IROW(65),KKNG
      COMMON/USER10/ UPBND
      COMMON/USER9/ ATAB,T,ZOPT,PCTTOL,SOLMIN
      IRTR=0
      NPASS=0
      DO 5 I=1,MSCO
      DO 5 J=1,NMAX
      ATAB(I,J)=TTAB(I,J)
  5   CONTINUE
C  ADD 1ST OBJ. FN. TO ATAB(I,J)
      LQR=MSTAG*MSTOPG
      DO 20 I=1,LQR
      II=I+1
      ATAB(1,II)=TTAB(MSCO+1,II)
  20  CONTINUE
C  DEFINE THE VARIANT INPUT DATA: IROW(I)-VECTOR OF CONST. TYPE
C                                 NCSM-# OF CALLS OF SUBROUT MINT
      SOLMIN=FLOAT(MDISL)
      PCTTOL=0.0
      M=MSCO
      N=NMAX
      KA=2*MSTAG+1
      DO 30 I=2,KA
  30  IROW(I)=0
      KA=KA+1
      DO 35 I=KA,M
  35  IROW(I)=-1
      NCSM=0
      LOVE=0
      KKNG=0
C  RUN THE SUBROUTINE MINT
      CALL SMINT(JHANG)
      IF(JHANG.EQ.1) GO TO 801
C  COMPUTE DEGREES OF ACCOMPLISHMENT FN.
      CALL COMPT(TTAB,T)
      KPOINT=MSTAG
      JPOINT=MSTAG
C  DETERMINE MDISLG
      MZOPT=ZOPT+0.001
      IF(MZOPT.GT.MDISL) GO TO 919
      WRITE(6,33) MZOPT,MDISL
  33  FORMAT(/,T5,'** MINIMAL TRAVEL DIST. OF THE ROUTE IS',I5,/,
     *T5,'** RESTRICTION ON VEH. TRAV. DIST. IS',I5,//,T5,
     *'ENTER UPPER LIMIT OF TRAVEL DIST. CONSTRAINT BASED ON THE'
     *,' INFORMATION GIVEN ABOVE.')
      READ(5,*) MDISLG
      WRITE(6,34) MDISLG
  34  FORMAT(T3,'TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:',I5)
C  RENEW INPUT DATA ARRAY,RHS, AND ADD 2ND OBJ. FN
  80  DO 40 I=1,MMAX
      DO 40 J=1,NMAX
```

```
      ATAB(I,J)=TTAB(I,J)
   40 CONTINUE
      ATAB(MSCO+1,1)=FLOAT(MDISLG)
      DO 45 I=1,MSTOPG
      KA=(KPOINT-1)*MSTOPG+I+1
      TTAB(MMAX,KA)=0.0
   45 CONTINUE
      DO 41 I=1,NMAX
      ATAB(1,I)=TTAB(MMAX,I)
   41 CONTINUE
C  FIX A LINK DETERMINED AND SO MODIFY CONST. (1)
      IF(NCSM.EQ.0) GO TO 48
      KX=(JPOINT-1)*MSTOPG+KPOINT
      IF(KPOINT.GE.JPOINT) KX=KX-1
      DO 44 I=1,NMAX
      II=I+1
      ATAB(JPOINT+1,II)=0.0
      IF(I.EQ.KX) ATAB(JPOINT+1,II)=1.0
      TTAB(JPOINT+1,II)=ATAB(JPOINT+1,II)
   44 CONTINUE
      JPOINT=KPOINT
C  DEFINE THE VARIANT INPUT DATA
   48 SOLMIN=FLOAT(MDISLG)
      PCTTOL=0.0
      M=MMAX-1
      N=NMAX
      KA=2*MSTAG+1
      DO 50 I=2,KA
      IROW(I)=0
   50 CONTINUE
      KA=KA+1
      DO 55 I=KA,MSCO
      IROW(I)=-1
   55 CONTINUE
      IROW(MSCO+1)=-1
C  RUN THE SUBROUTINE MINT
      IOUT1=0
      CALL SMINT(JHANG)
      IF(JHANG.EQ.1) GO TO 801
C  COMPUTE THE DEGREES OF ACCOMPLISHMENT FN.
      CALL COMPT(TTAB,T)
      NCSM=NCSM+1
      LOPT=ZOPT+0.001
      KBB=LOPT-JPSLG
      IF(KBB.LE.0) KBB=0
      IF(KBB.LE.0) GO TO 500
      IF(NCSM.GE.(MSTOPG-1)) GO TO 700
C  NEXT STATION TO VISIT IS DETERMINED
      DO 60 I=1,MSTOPG
      LQR=I
      IF(I.GE.KPOINT) LQR=LQR+1
      KA=(KPOINT-1)*MSTOPG+I
      BB=DABS(T(KA)-1.0)
      IF(BB.LE.0.001) GO TO 65
   60 CONTINUE
   65 KPOINT=LQR
      INEXT=ICLUST(IROWG,KPOINT)
      GO TO 80
  500 IF((NEMCI+NCOCI).EQ.0) GO TO 700
      IF(NCSM.GE.2.AND.NCOCI.EQ.0) GO TO 700
      KKNG=1
      LOVE=1
C  RENEW ATAB(I,J),ADD 3RD OBJ. FN. AND RHS
      DO 505 I=1,MMAX
      DO 505 J=1,NMAX
      ATAB(I,J)=TTAB(I,J)
```

```
  505 CONTINUE
      DO 507 I=1,NZR1VR
  507 ATAB(1,I+1)=1.0
      IF(NEMCI.EQ.0) GO TO 518
      DO 510 I=1,NEMCI
      KA=MXX(I)+1
      ATAB(1,KA)=0.0
  510 CONTINUE
  518 IF(NCOCI.EQ.0) GO TO 519
      DO 511 I=1,NCOCI
      KA=MYY(I)+1
      ATAB(1,KA)=0.0
  511 CONTINUE
  519 ATAB(MSCO+1,1)=FLOAT(MDISLG)
      ATAB(MMAX,1)=FLOAT(JPSLG)
C  DEFINE THE VARIANT INPUT DATA
      SOLMIN=FLOAT(MSTAG)
      PCTTOL=0.0
      M=MMAX
      N=NMAX
      KA=2*MSTAG+1
      DO 515 I=2,KA
  515 IROW(I)=0
      KA=KA+1
      DO 520 I=KA,MSCO
  520 IROW(I)=-1
      IROW(MSCO+1)=-1
      IROW(M)=-1
C  RUN THE SUBROUTINE MINT
      CALL SMINT(JHANG)
      IF(JHANG.EQ.1) GO TO 801
C  COMPUTE THE DEGREES OF ACCOMPLISHMENT FN.
      CALL COMPT(TTAB,T)
  700 WRITE(6,718) IROWG
  718 FORMAT(T2,'** THE MOST SATISFACTORY ROUTE SEQUENCE'
     *' OBTAINED FOR CLUSTER',I3,' IS:')
      KOR=MSTAG+1
      WRITE(6,901) (IBB(I),I=1,KOR)
  901 FORMAT(/,T5,'ROUTE SEQUENCE:',12I4)
      WRITE(6,902) IBB(MSTAG+2),IBB(MSTAG+3),IBB(MSTAG+4)
  902 FORMAT(T5,'TOT. DIS.=',I5,5X,'TOT. DET.=',I5,5X,
     *'TOT. FULL. OF EM. SERV. & COND. DEP.=',I5)
  801 RETURN
C  INFORM THE VIOLATION OF RESTRICTION ON VEH. TRAV. DIST.
  919 IRTR=1
      WRITE(6,929)
  929 FORMAT(T2,'!ERROR! RESTRICTION ON VEH. TRAV. DIST. IS',
     *' VIOLATED!!',/,T2,'CONVERT TO THE PREVIOUS SUBSETS FORMATION!')
      RETURN
      END
C
C
C
      SUBROUTINE SMINT(IHANG)
C*****************************************************************
C    IT IS FOR MIXED INTEGER LINEAR PROGRAMMING.
C*****************************************************************
      DOUBLE PRECISION DABS
      DOUBLE PRECISION ATAB(65,70), UPBND(70), TPVAL(60), BTMVL(60),
     1VAL(100), TBSAV(65,70), SAVTAB(65,2200), T(70)
      DOUBLE PRECISION SOLMIN, PCTTOL, TLRNCE, YVECT, ATAB11, AMAX,
     1RTIO, ALFA, ARTIO, ADELT, ZOPT, ATAB12, X1, AMAX2, AMAX3, ALW,
     2AUP, RTIO2, DIFF1, DIFF2, DIFF, SVALW, ANDCT4
      DIMENSION ITBROW(65),ICOL(70),ITBCOL(70),IVAR(70)
      DIMENSION ISVROW(65,60),ISVRCL(60),ICORR(60),ISVN(60),KSVN(60)
      COMMON/USER8/ NZR1VR,ISIZE,IOUT1,IOUT2,IOUT3,M,N,IROW(65),KKNG
```

```
      COMMON/USER10/ UPBND
      COMMON/USER9/ ATAB,T,ZOPT,PCTTOL,SOLMIN
      X1 = 1.0
   10 FORMAT (1HO, (7D10.3))
C     UNPACKED FORMAT NO. 11
   12 FORMAT ( 1X, 8D13.7)
   14 FORMAT (1HO,30HUPPER BOUND ON VARIABLE 1 TO N)
  15  FORMAT(20I4)
   18 FORMAT (4HOI =, I4, 6I10)
   19 FORMAT (27HOSTRUCTURAL VARIABLES: X(I))
   21 FORMAT (30HOCONSTRAINT TYPES IN ROW ORDER)
   22 FORMAT (52HOINPUT TABLEAU ECHO, CONSTRAINT VALUE LEFT.  BY ROW.)
   23 FORMAT (1HO,10D13.3/(1H , 10D13.3))
   24 FORMAT (1HO,13HITERATION NO.,I6)
   25 FORMAT ( 1HO,8D13.5/(1H , 8D13.5))
   26 FORMAT ( 1H , I6, 7I13)
   27 FORMAT(1H+,114X,I5)
   29 FORMAT (18HOTOLERANCE SET AT ,E15.7,14H  AT ITERATION,I6)
   30 FORMAT(21H PROBLEM NOT FEASIBLE)
   35 FORMAT (21HOOBJECTIVE FUNCTION =, F15.7,14H  AT ITERATION,I6)
   40 FORMAT (29HOCONTINUOUS SOLUTION COMPLETE)
   42 FORMAT (38HOFINAL TABLEAU FOR CONTINUOUS SOLUTION)
   45 FORMAT(40HOCONTINUOUS SOLUTION IS INTEGER SOLUTION)
   46 FORMAT (1HO,30HNO INTEGER VARIABLES REQUESTED)
   50 FORMAT (23HOOPTIMALITY ESTABLISHED)
   55 FORMAT(33HOPROBLEM TOO BIG FOR MACHINE SIZE)
   65 FORMAT (30HOEND OF PROBLEM, ITERATION NO., I6)
   70 FORMAT(26HOBRANCH POINT INCREASED TO,I4)
   75 FORMAT(26HOBRANCH POINT DECREASED TO,I4)
   78 FORMAT (24HOINITIAL WORKING TABLEAU)
      NI = 5
      NO = 6
C     INITIALIZATION
      IHANG=0
   68 CONTINUE
      INDCT7=1
      KSVN(1)=1
      INDCTR=1
      ICNTR=0
      I1ROW=1000
      IROW(1)=0
      ADELT = 5.0E-7
   73 DO 72 I=1,N
   72 T(I)=0.
      NM1=N-1
   74 IF(SOLMIN)786,787,786
C     INPUT UPPER BOUND ON OBJECTIVE FUNCTION
  786 TLRNCE=SOLMIN
      PCTTOL=-1.
      GO TO 90
  787 ITOL=1
      SOLMIN = 1E35
      IF(PCTTOL)90,788,90
  788 PCTTOL=.1
  90  ICHAMP=0
      IF(ICHAMP.EQ.0) GO TO 91
      WRITE(NO,14)
      WRITE(NO,10) (UPBND(I), I = 1,NM1)
C     CONSTRAINT TYPES: ( +1, = 0, ' -1 .
      WRITE (NO, 21)
      WRITE (NO, 15) (IROW(I), I = 2, M)
  91  ICHAMP=0
      IF(ICHAMP.EQ.0) GO TO 9520
C     PRINT INPUT TABLEAU FOR ERROR CHECK
      WRITE(NO,22)
      DO 80 I = 1, M
```

```
      WRITE (NO, 23) (ATAB(I,J), J = 1, N)
   80 CONTINUE
 9520 DO 954 I=2,M
      IF(IROW(I))953,9521,9521
 9521 DO 9523 J=2,N
 9523 ATAB(I,J)=-ATAB(I,J)
      GO TO 954
  953 ATAB(I,1)=-ATAB(I,1)
  954 CONTINUE
  450 CONTINUE
  955 DO 98 I=2,N
      IF(UPBND(I-1))96,96,98
   96 UPBND (I-1) = 1E3
   98 CONTINUE
C     COMPUTE NO. OF Y VECTORS
  981 YVECT=UPBND(1)+1.
      IF ( NZR1VR .LT. 2) GO TO 322
      DO 982 I=2,NZR1VR
  982 YVECT=YVECT*(UPBND(I)+1.)
  322 CONTINUE
C     SET SOLUTION VECTOR OF VARIABLES EQUAL TO ZERO
C     AND SAVE ORIGINAL UPPER BOUNDS
  985 DO 99 I=2,N
   99 IVAR(I-1)=0
C     INITIALIZE ROW AND COLUMN IDENTIFIERS,+K=VARIABLE NO. K,
C     ZERO = ZERO SLACK, -K = POSITIVE SLACK
      IF ( M .LT. 2) GO TO 451
      DO 102 I=2,M
      IF(IROW(I))100,102,100
  100 IROW(I)=1-I
  102 CONTINUE
  451 CONTINUE
      ATAB11=ATAB(1,1)
      ICOL(1) = 0
      DO 103 J=2,N
      IF(ATAB(1,J))1022,1025,1025
 1022 DO 1023 I=1,M
      ATAB(I,1)=ATAB(I,1)+ATAB(I,J)*UPBND(J-1)
 1023 ATAB(I,J)=-ATAB(I,J)
      ICOL(J)=1000+J-1
      GO TO 103
 1025 ICOL(J)=J-1
  103 CONTINUE
C     OUTPUT INITIAL TABLEAU
      IF(IOUT2)104,254,104
  104 WRITE(NO,78)
      WRITE(NO,26)(ICOL(J),J=1,N)
      DO 110 I=1,M
      WRITE(NO,25)(ATAB(I,J),J=1,N)
  110 WRITE(NO,27)IROW(I)
      GO TO 254
C     START DUAL LP
C     CHOOSE PIVOT ROW, MAXIMUM POSITIVE VALUE IN CONSTANT COLUMN
  112 AMAX = 0.0
      IF ( M .LT. 2) GO TO 452
      DO 120 I=2,M
      IF(ATAB(I,1))120,120,115
  115 IF(ATAB(I,1)-AMAX)120,120,117
  117 AMAX=ATAB(I,1)
      IPVR=I
  120 CONTINUE
  452 CONTINUE
C     IF NO POSITIVE VALUE, LP FINISHED (PRIMAL FEASIBLE)
      IF(AMAX)265,265,130
C     CHOOSE PIVOT COLUMN, ALGEBRAICALLY MAXIMUM RATIO A(1,J)/A(PIVOTROW
C     FOR A (PIVOTROW,J) NEGATIVE. IF NO NEGATIVE A(PIVOTROW,J) PROBLEM
```

```
C       INFEASIBLE
  130 AMAX = -1E35
      IF(N-2)143,132,132
  132 IPVC=0
      DO 140 J=2,N
      IF(ATAB(IPVR,J))133,140,140
  133 RTIO=ATAB(1,J)/ATAB(IPVR,J)
      IF(RTIO-AMAX)140,137,135
  135 AMAX=RTIO
  136 IPVC=J
      GO TO 140
  137 IF(ATAB(IPVR,J)-ATAB(IPVR,IPVC))136,140,140
  140 CONTINUE
      IF(IPVC)150,143,150
  143 GO TO (145,435,542,610,665),INDCTR
  145 WRITE(NO,30)
      GO TO 1001
C     CARRY OUT PIVOT STEP
  150 ALFA=ATAB(IPVR,IPVC)
C       UPDATE TABLEAU
      DO 180 J=1,N
      IF(ATAB(IPVR,J))152,180,152
  152 IF(J-IPVC)153,180,153
  153 ARTIO=ATAB(IPVR,J)/ALFA
      DO 175 I=1,M
      IF(ATAB(I,IPVC))157,175,157
  157 IF(I-IPVR)160,175,160
  160 ATAB(I,J)=ATAB(I,J)-ARTIO*ATAB(I,IPVC)
      IF(DABS(ATAB(I,J))-ADELT) 165, 165, 175
  165 ATAB(I,J) = 0.0
  175 CONTINUE
  180 CONTINUE
      DO 190 J=1,N
  190 ATAB(IPVR,J)=ATAB(IPVR,J)/ALFA
C     EXCHANGE ROW AND COLUMN IDENTIFIERS
      ISV=IROW(IPVR)
      IROW(IPVR)=ICOL(IPVC)
      IF(ISV)197,195,197
C     IF PIVOT ROW WAS ZERO SLACK, SET MODIFIED PIVOT COLUMN ZERO.
  195 DO 196 I=1,M
  196 ATAB(I,IPVC)=ATAB(I,N)
      ICOL(IPVC)=ICOL(N)
      N=N-1
      GO TO 200
  197 DO 198 I=1,M
  198 ATAB(I,IPVC)=-ATAB(I,IPVC)/ALFA
      ICOL(IPVC)=ISV
      ATAB(IPVR,IPVC)=1./ALFA
C     COUNT PIVOTS
  200 ICNTR=ICNTR+1
      IF(ICNTR.GT.600) GO TO 3447
      IF(IROW(IPVR)+1000)210,205,210
  205 DO 207 J=1,N
  207 ATAB(IPVR,J)=ATAB(M,J)
      IROW(IPVR)=IROW(M)
      M=M-1
  210 IF(IOUT1)240,2505,240
C     OUTPUT CURRENT TABLEAU
  240 WRITE (NO,24) ICNTR
      WRITE(NO,26)(ICOL(J),J=1,N)
      DO 250 K=1,M
      WRITE(NO,25)(ATAB(K,L),L=1,N)
  250 WRITE(NO,27)IROW(K)
 2505 GO TO (254,251,252,253,2535),INDCTR
C     IF SEEKING INTEGER SOLUTION, TEST OBJECTIVE FUNCTION AGAINST CURRE
  251 IF(ATAB(1,1)-TLRNCE)254,435,435
```

```
  252 IF(ATAB(1,1)-TLRNCE)254,542,542
  253 IF(ATAB(1,1)-TLRNCE)254,610,610
 2535 IF(ATAB(1,1)-TLRNCE)254,665,665
C     IF CONSTANT COLUMN OF ZERO SLACK ROW IS NEG., REVERSE SIGNS OF ENT
  254 IF ( M .LT. 2) GO TO 453
      DO 260 K = 2, M
      IF(IROW(K))260,255,260
  255 IF(ATAB(K,1))256,260,260
  256 DO 258 L=1,N
  258 ATAB(K,L)=-ATAB(K,L)
  260 CONTINUE
  453 CONTINUE
C     GO TO NEXT PIVOT STEP
      GO TO 112
  265 CONTINUE
C     IF ANY BASIS VARIABLE EXCEEDS ITS UPPER BOUND, COMPLEMENT IT, AND
C     PIVOT ON CORRESPONDING ROW
      IF ( M .LT. 2) GO TO 454
      DO 275 I=2,M
      IF(IROW(I))275,275,266
  266 J=IROW(I)
      IF(J-1000)268,268,267
  267 J=J-1000
  268 IF(UPBND(J)+ATAB(I,1))269,275,275
  269 IF(ADELT+UPBND(J)+ATAB(I,1))270,274,274
  270 ATAB(I,1)=-ATAB(I,1)-UPBND(J)
      DO 271 K=2,N
  271 ATAB(I,K)=-ATAB(I,K)
      IPVR=I
      IF(J-IROW(I))272,273,272
  272 IROW(I)=J
      GO TO 130
  273 IROW(I)=IROW(I)+1000
      GO TO 130
  274 ATAB(I,1)=-UPBND(J)
  275 CONTINUE
  454 CONTINUE
C     TRUE END OF LINEAR PROGRAMMING
C     SET SOLUTION VECTOR VALUES FOR BASIC VARIABLES
      IF ( M .LT. 2) GO TO 455
      DO 280 I=2,M
      IF(IROW(I))280,280,277
  277 IF(IROW(I)-1000)279,279,278
  278 J=IROW(I)-1000
      T(J)=UPBND(J)+ATAB(I,1)
      GO TO 280
  279 J=IROW(I)
      T(J)=-ATAB(I,1)
  280 CONTINUE
  455 CONTINUE
C     SET SOLUTION VECTOR VALUES FOR NON-BASIC VARIABLES IN COMPLEMENTED
      DO 285 I=2,N
      IF(ICOL(I))285,285,282
  282 IF(ICOL(I)-1000)284,284,283
  283 J=ICOL(I)-1000
      T(J)=UPBND(J)
      GO TO 285
  284 J=ICOL(I)
      T(J)=0.
  285 CONTINUE
      GO TO (286,437,548,615,670),INDCTR
  286 NXXYY=0
      IF(NXXYY.EQ.0) GO TO 291
C     FIRST TIME,WRITE CONTINUOUS SOLUTION TABLEAU
      WRITE(NO,40)
      IF(IOUT3)287,291,287
```

```
  287 WRITE(NO,42)
      WRITE(NO,26)(ICOL(J),J=1,N)
  288 DO 290 I=1,M
      WRITE(NO,25)(ATAB(I,J),J=1,N)
  290 WRITE(NO,27)IROW(I)
  291 ZOPT =DABS( ATAB(1,1))
      IF(NXXYY.EQ.0) GO TO 1004
      WRITE (NO, 35) ZOPT, ICNTR
      WRITE (NO, 19)
      WRITE (NO,18) (I, I = 1, NM1)
      WRITE (NO, 10) (T(I), I = 1, NM1)
C     COMPUTE ABSOLUTE TOLERANCE
 1004 ATAB12=ATAB(1,1)
      ATAB11 =DABS (ATAB11 - ATAB(1,1))
      IF(PCTTOL)294,293,292
  292 TLRNCE=PCTTOL*ATAB11+ATAB12
      GO TO 294
  293 TLRNCE = 1E35
  294 CONTINUE
C     DETERMINE WHETHER CONTINUOUS SOLUTION IS MIXED INTEGER SOLUTION
      IF ( M .LT. 2) GO TO 456
  301 DO 310 I=2,M
      IF(IROW(I))310,310,302
  302 IF(IROW(I)-1000)303,303,304
  303 IF(IROW(I)-NZR1VR)305,305,310
  304 IF(IROW(I)-1000-NZR1VR)305,305,310
  305 AJ01 = ATAB(I,1)
      AJ02 = ADELT
      AJ03 = X1
      IF(AMOD(-AJ01,AJ03)-AJ02) 310,310,306
  306 IF(1.0-AMOD(-AJ01,AJ03)-AJ02) 310,310,295
  310 CONTINUE
  456 CONTINUE
      IF ( NZR1VR) 307, 308, 307
  307 WRITE (NO,45)
      GO TO 998
  308 WRITE (NO,46)
      GO TO 998
C DETERMINE WHETHER PROBLEM FITS IN MEMORY , AND IF SO  WHETHER TO SAVE
C     ALL INTERMEDIATE TABLEAUS OR ONLY SOME
  295 IF(N-NZR1VR)297,297,298
  297 ISVLOC=(N*(N+1))/2
      GO TO 299
  298 ISVLOC=(NZR1VR*(2*N-NZR1VR+1))/2
  299 IF(ISIZE-ISVLOC)3001,3001,300
  300 I1ROW=0
      GO TO 315
 3001 NONBSC=0
      DO 3006 J=2,N
      IF(ICOL(J))3006,3006,3002
 3002 IF(ICOL(J)-1000)3003,3004,3004
 3003 IF(ICOL(J)-NZR1VR)3005,3005,3006
 3004 IF(ICOL(J)-1000-NZR1VR)3005,3005,3006
 3005 NONBSC=NONBSC+1
 3006 CONTINUE
      IF(N-NZR1VR)3007,3007,3008
 3007 ISVLOC=N+((N-NONBSC)*(N-NONBSC+1))/2
      GO TO 3009
 3008 ISVLOC=N+((NZR1VR-NONBSC)*(N-NONBSC+N-NZR1VR+1))/2
 3009 IF(ISIZE-ISVLOC)3010,3010,315
 3010 WRITE(NO,55)
      GO TO 998
  315 CONTINUE
C     BEGIN INTEGER PROGRAMMING
  400 I1=1
  402 AMAX = -X1
```

```
      KSVN(I1+1)=KSVN(I1)
C     CHOOSE NEXT INTEGER VARIABLE TO BE CONSTRAINED
C     TRY NONBASIC VARIABLES FIRST, CHOOSING ONE WITH LARGEST SHAD PRICE
      DO 4085 I=2,N
      IF(ICOL(I))4085,4085,405
  405 IF(ICOL(I)-1000)406,407,407
  406 IF(ICOL(I)-NZR1VR)408,408,4085
  407 IF(ICOL(I)-1000-NZR1VR)408,408,4085
  408 IF(AMAX-ATAB(1,I))4082,4085,4085
 4082 ISVI=I
      AMAX=ATAB(1,I)
 4085 CONTINUE
C     IF NONE LEFT, TRY BASIC VARIABLES
      IF ( AMAX + X1) 4087, 420, 4087
C     VARIABLE CHOSEN
 4087 IVAR(I1)=ICOL(ISVI)
      BTMVL(I1)=-1.
      ISVRCL(I1)=ISVI
      ICORR(I1)=0
      VAL (I1) = 0.0
C     IF OBJECTIVE FUNCTION VALUE + SHADOW PRICE EXCEEDS TOLERANCE,
C     INDICATE UPWARD DIRECTION INFEASIBLE
      IF(ATAB(1,1)+ATAB(1,ISVI)-TLRNCE)410,409,409
  409 TPVAL(I1)=1000.
      IF(I1-1)4101,4101,4095
 4095 ISVN(I1)=0
      GO TO 4132
  410 TPVAL(I1)=1.
      IF(I1-1)4100,4101,4100
C     SAVE ENTIRE TABLEAU OR ONLY COLUMN CORRESPONDING TO CURRENT
C     NONBASIC VARIABLE, DEPENDING ON SIZE OF PROB AND 2ND DIM OF SAVTAB
 4100 IF(I1-I1ROW)4132,4101,4101
 4101 L=KSVN(I1)
      DO 412 J=1,M
      ISVROW(J,I1)=IROW(J)
      DO 411 K=1,N
      I=L+K-1
      IF(J-1)4105,4105,411
 4105 SAVTAB(M+1,I)=ICOL(K)
  411 SAVTAB(J,I)=ATAB(J,K)
  412 CONTINUE
      ISVN(I1)=N
      KSVN(I1+1)=L+N
 4132 ICOL(ISVI)=ICOL(N)
      DO 4135 J=1,M
 4135 ATAB(J,ISVI)=ATAB(J,N)
      N=N-1
      GO TO 5000
C     CHOOSE NEXT INTEGER VARIABLE TO BE CONSTRAINED FROM
C     AMONG BASIC VARIABLES IN CURRENT TABLEAU
  420 CONTINUE
      IF(I1-I1ROW)4204,600,4205
 4204 I1ROW=I1
 4205 INDCT7=1
  421 AMAX = -X1
      IF ( M .LT. 2) GO TO 457
      DO 425 I2=2,M
      IF(IROW(I2))425,425,422
  422 IF(IROW(I2)-1000)423,424,424
  423 IF(IROW(I2)-NZR1VR)4241,4241,425
  424 IF(IROW(I2)-1000-NZR1VR)4241,4241,425
 4241 AMAX2 = 1.0E35
       AMAX3 = -1.0E35
      AJ0 = -ATAB(I2,1) + ADELT
      ALW = AINT(AJ0)
      AUP=ALW+1.
```

```
      IF(N-1)426,426,4240
 4240 DO 4246 I3=2,N
      IF(ATAB(I2,I3))4244,4246,4242
 4242 RTIO=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO-AMAX2)4243,4246,4246
 4243 AMAX2=RTIO
      GO TO 4246
 4244 RTIO2=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO2-AMAX3)4246,4246,4245
 4245 AMAX3=RTIO2
 4246 CONTINUE
      IF ( AMAX3 + 1E34) 430, 430, 4247
 4247 IF (AMAX2 - 1E34) 4248, 429, 429
 4248 DIFF1 =DABS (AMAX2 * (ATAB(I2,1) + ALW))
      DIFF2 =DABS (AMAX3 * (ATAB(I2,1) + AUP))
      DIFF =DABS (DIFF1 - DIFF2)
      IF(DIFF-AMAX)425,425,4249
 4249 AMAX=DIFF
      SVALW=ALW
      ISVI2=I2
      IF(DIFF1-DIFF2)4251,4251,4252
 4251 ANDCT4=0.
      GO TO 425
 4252 ANDCT4=1.
  425 CONTINUE
  457 CONTINUE
      ALW=SVALW
      I2=ISVI2
      VAL(I1)=ALW+ANDCT4
      BTMVL(I1)=VAL(I1)-1.
 4255 TPVAL(I1)=VAL(I1)+1.
      GO TO 432
C     IF NO. OF COLS=1 AND RIGHT HAND SIDE=0, DONT GO TO LP
  426 IF (DABS( ATAB(I2,1) + ALW) - ADELT) 427, 427, 5100
  427 BTMVL(I1)=-1.
      TPVAL(I1)=1000.
      VAL(I1)=ALW
      IVAR(I1)=IROW(I2)
      IROW(I2)=0
      GO TO 5000
C     CONSTRAINING VARIABLE IN LOWER DIRECTION INFEASIBLE
  429 BTMVL(I1)=-1.
      IF (DABS ( ATAB(I2,1) + ALW) - ADELT ) 4295, 4295, 4296
 4295 ANDCT4=0.
      VAL(I1)=ALW+ANDCT4
      GO TO 4255
 4296 TPVAL(I1)=ALW+2.
      ANDCT4=1.
      GO TO 431
C     CONSTRAINING VARIABLE IN UPPER DIRECTION INFEASIBLE
  430 TPVAL(I1)=1000.
      BTMVL(I1)=ALW-1.
      ANDCT4=0.
  431 VAL(I1)=ALW+ANDCT4
C     SAVE ENTIRE TABLEAU
  432 JSVN=N
      L=KSVN(I1)
  438 DO 439 I3=1,M
      ISVROW(I3,I1)=IROW(I3)
      DO 439 I4=1,N
      I6=L+I4-1
      IF(I3-1)4385,4385,439
 4385 SAVTAB(M+1,I6)=ICOL(I4)
  439 SAVTAB(I3,I6)=ATAB(I3,I4)
      ISVN(I1)=N
      KSVN(I1+1)=L+N
```

```
      ATAB(I2,1)=ATAB(I2,1)+VAL(I1)
      ISVRCL(I1)=I2
      IVAR(I1)=IROW(I2)
      ICORR(I1)=1
      IROW(I2)=0
      IF (DABS ( ATAB(I2,1)) - ADELT) 433, 433, 434
  433 ATAB (I2,1) = 0.0
  434 INDCTR=2
C     RETURN TO CARRY OUT LP
      IF(IOUT1)240,254,240
C     INFINITE RETURN
  435 IF(ANDCT4)4355,4352,4355
 4352 BTMVL(I1)=-1.
      GO TO 5120
 4355 TPVAL(I1)=1000.
      GO TO 5120
C     FINITE RETURN
  437 GO TO 5000
C     TEST FOR ANY INTEGER VARIABLES LEFT TO BE CONSTRAINED
 5000 IF(I1-NZR1VR)5050,550,550
C     INCREMENT POINTER AND RETURN TO CONSTRAIN NEXT INTEGER VARIABLE
 5050 I1=I1+1
      IF(IOUT1)5051,402,5051
 5051 WRITE(NO,70)I1
      GO TO 402
C     DECREMENT POINTER AND CONSTRAIN CURRENT VARIABLE TO
C     CURRENT VALUE + OR - 1
 5100 I1=I1-1
      IF(IOUT1)5110,5115,5110
 5110 WRITE(NO,75)I1
 5115 IF(I1)995,995,5120
 5120 IF(IVAR(I1)-1000)5151,5151,5152
 5151 K=IVAR(I1)
      GO TO 5153
 5152 K=IVAR(I1)-1000
 5153 I2=ISVRCL(I1)
 5155 IF(BTMVL(I1))516,517,517
  516 IF(TPVAL(I1)-UPBND(K))518,518,5100
  517 IF(TPVAL(I1)-UPBND(K))530,530,525
C     TOP END FEASIBLE
  518 INDCT5=1
 5181 IF(ICORR(I1))5198,5182,5198
 5182 IF(I1-I1ROW)5183,5198,5198
 5183 INDCT8=1
      IF(I1-1)5185,5198,5185
 5185 INDCT5=4
      ISVI1=I1-1
      I1=1
      GO TO 5198
 5190 DO 5194 I3=1,ISVI1
      I4=ISVRCL(I3)
      ICOL(I4)=ICOL(N)
      DO 5193 J=1,M
      IF(VAL(I3)-1.)5193,5191,5192
 5191 ATAB(J,1)=ATAB(J,1)+ATAB(J,I4)
      GO TO 5196
 5192 ATAB(J,1)=ATAB(J,1)+VAL(I3)*ATAB(J,I4)
 5196 INDCT8=2
 5193 ATAB(J,I4)=ATAB(J,N)
      N=N-1
 5194 CONTINUE
 5195 I1=ISVI1+1
      INDCT5=1
      GO TO 521
C     RETRIEVE SAVED TABLEAU
 5198 N=ISVN(I1)
```

```
      L=KSVN(I1)
      DO 5199 I3=1,M
      IROW(I3)=ISVROW(I3,I1)
      DO 5199 I4=1,N
      I6=L+I4-1
      IF(I3-1)5197,5197,5199
 5197 ICOL(I4)=SAVTAB(M+1,I6)
 5199 ATAB(I3,I4)=SAVTAB(I3,I6)
 5205 GO TO (521,526,531,5190),INDCT5
  521 VAL(I1)=TPVAL(I1)
      TPVAL(I1)=TPVAL(I1)+1.
      IF(ICORR(I1))541,522,541
  522 DO 523 I3=1,M
      ATAB(I3,1)=ATAB(I3,1)+(VAL(I1)*ATAB(I3,I2))
      IF (DABS ( ATAB(I3,1)) - ADELT) 5225, 5225, 523
 5225 ATAB(I3,1)=0.
  523 ATAB(I3,I2)=ATAB(I3,N)
      ICOL(I2)=ICOL(N)
      N=N-1
      IF(ATAB(1,1)-TLRNCE)5235,5100,5100
 5235 IF(I1-I1ROW)650,5415,5415
C     BOTTOM END FEASIBLE
  525 INDCT5=2
      GO TO 5198
  526 VAL(I1)=BTMVL(I1)
      BTMVL(I1)=BTMVL(I1)-1.
      GO TO 541
C     BOTH ENDS FEASIBLE
  530 INDCT5=3
      GO TO 5198
  531 AMAX2 = 1.0E35
      AMAX3 = -1.0E35
      DO 536 I3=2,N
      IF(ATAB(I2,I3))534,536,532
  532 RTIO=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO-AMAX2)533,536,536
  533 AMAX2=RTIO
      GO TO 536
  534 RTIO2=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO2-AMAX3)536,536,535
  535 AMAX3=RTIO2
  536 CONTINUE
      IF(AMAX2-1.E35)538,537,537
C     BOTTOM END INFEASIBLE
  537 BTMVL(I1)=-1.
      GO TO 521
  538 IF(AMAX3+1.E35)539,539,540
C     TOP END INFEASIBLE
  539 TPVAL(I1)=1000.
      GO TO 526
  540 DIFF1 =DABS ( AMAX2 * (ATAB(I2,1) + BTMVL (I1)))
      DIFF2 =DABS ( AMAX3 * (ATAB(I2,1) + TPVAL (I1)))
      IF(DIFF1-DIFF2)526,526,521
  541 ATAB(I2,1)=ATAB(I2,1)+VAL(I1)
      IROW(I2)=0
      IF (DABS ( ATAB(I2,1)) - ADELT) 5412, 5412, 5415
 5412 ATAB(I2,1)=0.
 5415 INDCTR=3
      IF(IOUT1)240,2505,240
C     INFINITE RETURN
  542 GO TO (544,547,543),INDCT5
  543 IF(TPVAL(I1)-VAL(I1)-1.)545,544,545
  544 TPVAL(I1)=1000.
      GO TO 5120
  545 IF(VAL(I1)-BTMVL(I1)-1.)546,547,546
  546 CONTINUE
```

```
  547 BTMVL(I1)=-1.
      GO TO 5120
C     FINITE RETURN
  548 GO TO 5000
C     FEASIBLE INTEGER SOLUTION OBTAINED
  550 TLRNCE=ATAB(1,1)
      SOLMIN=1.
C     WRITE CURRENT BEST MIXED INTEGER SOLUTION
      ZOPT =DABS( ATAB( 1,1))
      NXXYY=0
      IF(NXXYY.EQ.0) GO TO 553
      WRITE (NO, 35) ZOPT, ICNTR
  553 DO 560 I = 1, NZR1VR
      IF(IVAR(I))554,560,554
  554 IF(IVAR(I)-1000)555,555,557
  555 J=IVAR(I)
      T(J)=VAL(I)
      GO TO 560
  557 J=IVAR(I)-1000
      T(J)=UPBND(J)-VAL(I)
  560 CONTINUE
      IF(NXXYY.EQ.0) GO TO 1002
      WRITE (NO, 19)
  565 WRITE (NO, 18) (I, I = 1, NM1)
      WRITE (NO, 10) (T(I), I = 1, NM1)
      BOBO=0.0
      IF(BOBO.EQ.0.) GO TO 9976
      GO TO 5115
  600 GO TO (605,4205),INDCT7
  605 INDCTR=4
      IF(IOUT1)240,254,240
C     INFINITE RETURN
  610 GO TO 5100
C     FINITE RETURN
  615 INDCT7=2
      GO TO 402
C     IF USING SECOND SOLUTION METHOD, SAVE TABLEAU MODIFIED
C     FOR NONZERO VALUE OF NONBASIC VARIABLE IN TBSAV
  650 DO 655 I=1,M
      ITBROW(I)=IROW(I)
      DO 655 J=1,N
  655 TBSAV(I,J)=ATAB(I,J)
      DO 660 J=1,N
  660 ITBCOL(J)=ICOL(J)
      JSVN=N
      INDCTR=5
      IF(IOUT1)240,254,240
C     INFINITE RETURN
  665 GO TO (544,5120),INDCT8
C     FINITE RETURN
C     IF USING SECOND SOLUTION METHOD, RETRIEVE MODIFIED TABLEAU FROM
C     TBSAV, AS THIS CORRESPONDS TO SAVED COLUMNS FOR I1 LESS THAN I1ROW
  670 N=JSVN
      DO 675 I=1,M
      IROW(I)=ITBROW(I)
      DO 675 J=1,N
  675 ATAB(I,J)=TBSAV(I,J)
      DO 680 J=1,N
  680 ICOL(J)=ITBCOL(J)
      GO TO 5000
C     OUTPUT FINAL SOLUTION.
  995 IF(ITOL)996,9976,996
  996 IF(SOLMIN-1.E35)9976,997,997
  997 ITOL=ITOL+1
      TLRNCE=FLOAT(ITOL)*PCTTOL*ATAB11+ATAB12
      N=ISVN(1)
```

```
      DO 9972 I=1,M
      IROW(I)=ISVROW(I,1)
      DO 9972 J=1,N
 9972 ATAB(I,J)=SAVTAB(I,J)
      DO 9973 K=1,N
 9973 ICOL(K)=SAVTAB(M+1,K)
      GO TO 400
  998 CONTINUE
 9976 WRITE (NO, 50)
 1001 WRITE (NO, 65) ICNTR
 1002 RETURN
 3447 WRITE(6,3448)
 3448 FORMAT(/,10X,'* ALGORITHM IS TERMINATED DUE TO AN INF. LOOP*')
      IHANG=1
      RETURN
      END
C
C
C
      SUBROUTINE COMPT(TTAB,T)
C**************************************************************
C    IT COMPUTES THE VALUE OF EACH OBJECTIVE FOR THE ROUTE SEQUENCE
C    GENERATED BY RUNNING SLGP.
C**************************************************************
      DOUBLE PRECISION TTAB(65,70),T(70),UPBND(70)
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER6/ MSTOPG,MSTAG.MDISL,JPSLG,NEMCI,NCOCI,IROWG,JPSLGG
      COMMON/USER7/ NMAX,MMAX,MSCO,IBB(20)
      COMMON/USER8/ NZR1VR,ISIZE,IOUT1,IOUT2,IOUT3,M,N,IROW(65),KKNG
      COMMON/USER10/ UPBND
      JJTT=0
      JJTD=0
      JJFR=0
      DO 3 I=1,20
    3 IBB(I)=0
C  DETERMINE TOTAL DISTANCE
      DO 5 I=1,NZR1VR
      II=I+1
      KG=T(I)+0.001
      SSS=SNGL(TTAB(MSCO+1,II))
      JJTT=JJTT+IFIX(SSS)*KG
    5 CONTINUE
      IBB(MSTAG+2)=JJTT
      IBB(1)=0
      KPOINT=MSTAG
      IBEG=MSTA
C  DETERMINE ROUTE SEQ.,TOT. DETE. AND FULL. OF EMER.SERV.&COND.DEP.
      DO 20 K=2,MSTAG
      DO 10 I=1,MSTOPG
      LQR=I
      IF(I.GE.KPOINT) LQR=LQR+1
      KA=(KPOINT-1)*MSTOPG+I
C  CHECK THE VALUE OF DECISION VARIABLES IF IT IS 0 OR 1
      IF(T(KA).GT.0.001) GO TO 15
   10 CONTINUE
   15 KPOINT=LQR
      IDEST=ICLUST(IROWG,KPOINT)
      JJTT=JJTT-MDIS(IBEG,IDEST)
      IGA=JJTT-JPSLG
      IF(IGA.LE.0) IGA=0
      JJTD=JJTD+IGA
      IBB(K)=IDEST
      IF(NEMCI.EQ.0) GO TO 22
      DO 25 J=1,NEMCI
      IF(MXX(J).EQ.KA) GO TO 40
```

```
   25  CONTINUE
   22  IF(NCOCI.EQ.0) GO TO 20
       DO 30 L=1,NCOCI
       IF(MYY(L).EQ.KA) GO TO 40
   30  CONTINUE
       GO TO 20
   40  JJFR=JJFR+1
   20  CONTINUE
C  STORE TD AND FR
       IBB(MSTAG+1)=0
       IBB(MSTAG+3)=JJTD
       IBB(MSTAG+4)=JJFR
       RETURN
       END
C
C
C
       SUBROUTINE PCASE2(TTAB,IRTR,NPASS)
C*******************************************************************
C     IT IS FOR SLGP BASED ON THE GOAL PRIORITY STRUCTURE MODEL II.
C*******************************************************************
       DOUBLE PRECISION DABS
       DOUBLE PRECISION TTAB(65,70),ATAB(65,70),T(70),UPBND(70)
       DOUBLE PRECISION ZOPT,PCTTOL,SOLMIN
       COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
      *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
       COMMON/USER6/ MSTOPG,MSTAG,MDISL,JPSLG,NEMCI,NCOCI,IROWG,JPSLGG
       COMMON/USER7/ NMAX,MMAX,MSCO,IBB(20)
       COMMON/USER8/ NZR1VR,ISIZE,IOUT1,IOUT2,IOUT3,M,N,IROW(65),KKNG
       COMMON/USER10/ UPBND
       COMMON/USER9/ ATAB,T,ZOPT,PCTTOL,SOLMIN
       IRTR=0
       NPASS=1
C  COPY THE INPUT ARRAY TO ATAB(I,J)
       DO 5 I=1,MSCO
       DO 5 J=1,NMAX
       ATAB(I,J)=TTAB(I,J)
    5  CONTINUE
C  ADD 2ND OBJ. FN. TO ATAB(I,J)-- MIN. OF TT
       DO 20 I=1,NMAX
       ATAB(1,I)=TTAB(MMAX,I)
   20  CONTINUE
C  DEFINE THE VARIANT INPUT DATA
       SOLMIN=FLOAT(JPSLGG)
       PCTTOL=0.
       M=MSCO
       N=NMAX
       KA=2*MSTAG+1
       DO 30 I=2,KA
   30  IROW(I)=0
       KA=KA+1
       DO 35 I=KA,M
   35  IROW(I)=-1
C  RUN THE SUBROUTINE MINT
       CALL SMINT(JHANG)
       IF(JHANG.EQ.1) GO TO 801
C  COMPUTE DEGREES OF ACCOMPLISHMENT FN.
       CALL COMPT(TTAB,T)
       IF((NEMCI+NCOCI).EQ.0) GO TO 720
       NPASS=0
C  DETERMINE MDISLG
       MZOPT=ZOPT+0.001
       IF(MZOPT.GT.MDISL) GO TO 919
       WRITE(6,33) MZOPT,MDISL
   33  FORMAT(/,T5,'** MINIMAL TRAVEL DIST. OF THE ROUTE IS',I5,/,T5
      *,'** RESTRICTION ON VEH. TRAVEL DIST. IS',I5,//,T5,
```

```
     *'ENTER UPPER LIMIT OF TRAV. DIST. CONSTRAINT BASED ON THE',
     *' INFORMATION GIVEN ABOVE.')
      READ(5,*) MDISLG
      WRITE(6,34) MDISLG
   34 FORMAT(T2,'TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:',I5)
C  RENEW INPUT DATA ARRAY,RHS, AND ADD 3RD OBJ. FN.--MAX. OF FR
      DO 505 I=1,MMAX
      DO 505 J=1,NMAX
  505 ATAB(I,J)=TTAB(I,J)
      DO 45 I=1,MSTOPG
      KA=(MSTAG-1)*MSTOPG+I+1
      ATAB(MMAX,KA)=0.0
   45 CONTINUE
      ATAB(MSCO+1,1)=FLOAT(MDISLG)
      ATAB(MMAX,1)=FLOAT(JPSLGG)
      DO 507 I=1,NZR1VR
  507 ATAB(1,I+1)=1.0
      IF(NEMCI.EQ.0) GO TO 518
      DO 510 I=1,NEMCI
      KA=MXX(I)+1
      ATAB(1,KA)=0.0
  510 CONTINUE
  518 IF(NCOCI.EQ.0) GO TO 519
      DO 511 I=1,NCOCI
      KA=MYY(I)+1
      ATAB(1,KA)=0.0
  511 CONTINUE
C  DEFINE THE VARIANT INPUT DATA
  519 SOLMIN=FLOAT(MSTAG)
      PCTTOL=0.
      M=MMAX
      N=NMAX
      KA=2*MSTAG+1
      DO 515 I=2,KA
  515 IROW(I)=0
      KA=KA+1
      DO 520 I=KA,MMAX
  520 IROW(I)=-1
C  RUN THE SUBROUTINE
      CALL SMINT(JHANG)
      IF(JHANG.EQ.1) GO TO 801
C  COMPUTE THE DEGREES OF ACCOM. FN.
      CALL COMPT(TTAB,T)
  720 WRITE(6,718) IROWG
  718 FORMAT(T2,'** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR',
     *' CLUSTER',I3,' IS:')
      KOR=MSTAG+1
      WRITE(6,719) (IBB(I),I=1,KOR)
  719 FORMAT(/,T5,'ROUTE SEQ.:',12I4)
      WRITE(6,722) IBB(MSTAG+2),IBB(MSTAG+3),IBB(MSTAG+4)
  722 FORMAT(T5,'TOT. DIST.=',I5,5X,'TOT. DET.=',I5,5X,
     *'TOT. FULL. OF EM. SERV. & COND. DEP.=',I5)
  801 RETURN
C  INFORM THE VIOLATION OF RESTRICTION ON VEH. TRAV. DIST.
  919 IRTR=1
      WRITE(6,929)
  929 FORMAT(T2,'!ERROR! RESTRICTION ON VEH. TRAV. DIST. IS',
     *' VIOLATED!!',/,T2,'CONVERT TO THE PREVIOUS SUBSETS FORMATION!')
      RETURN
      END
C
C
C
      SUBROUTINE PCASE3(TTAB,IRTR,NPASS)
C*****************************************************************
C    IT IS FOR SLGP BASED ON THE GOAL PRIORITY STRUCTURE MODEL III.
```

```
C*********************************************************************
      DOUBLE PRECISION DABS
      DOUBLE PRECISION TTAB(65,70),ATAB(65,70),T(70),UPBND(70)
      DOUBLE PRECISION ZOPT,PCTTOL,SOLMIN
      COMMON/USER1/ MDIS(101,101),MP(100),MSTOP,MSTA
     *,ICLUST(20,10),MEX(10),MXX(10),MEY(10,2),MYY(10)
      COMMON/USER6/ MSTOPG,MSTAG,MDISL,JPSLG,NEMCI,NCOCI,IROWG,JPSLGG
      COMMON/USER7/ NMAX,MMAX,MSCO,IBB(20)
      COMMON/USER8/ NZR1VR,ISIZE,IOUT1,IOUT2,IOUT3,M,N,IROW(65),KKNG
      COMMON/USER10/ UPBND
      COMMON/USER9/ ATAB,T,ZOPT,PCTTOL,SOLMIN
      IRTR=0
      NPASS=0
C  COPY THE INPUT ARRAY TO ATAB(I,J)
      DO 5 I=1,MSCO
      DO 5 J=1,NMAX
      ATAB(I,J)=TTAB(I,J)
    5 CONTINUE
C  ADD 1ST OBJ. FN. TO ATAB(I,J)---MIN. OF TT
      DO 20 I=1,NMAX
      ATAB(1,I)=TTAB(MSCO+1,I)
   20 CONTINUE
C  DEFINE THE VARIANT INPUT DATA
      SOLMIN=FLOAT(MDISL)
      PCTTOL=0.
      M=MSCO
      N=NMAX
      KA=2*MSTAG+1
      DO 30 I=2,KA
   30 IROW(I)=0
      KA=KA+1
      DO 35 I=KA,M
   35 IROW(I)=-1
      NCSM=0
C  RUN THE SUBROUTINE MINT
      CALL SMINT(JHANG)
      IF(JHANG.EQ.1) GO TO 500
C  COMPUTE THE DEGREES OF ACCOM. FN.
      CALL COMPT(TTAB,T)
C  DETERMINE MDISLG
      MZOPT=ZOPT+0.001
      IF(MZOPT.GT.MDISL) GO TO 919
      WRITE(6,33) MZOPT,MDISL
   33 FORMAT(/,T5,'** MINIMAL TRAVEL DIST. OF THE ROUTE IS',I5,/,T5,
     *'** RESTRICTION ON VEHICLE TRAV. DIST. IS',I5,//,T5,
     *'ENTER UPPER LIMIT OF TARV. DIST. CONSTRAINT BASED ON THE',
     *' INFORMATION GIVEN ABOVE.')
      READ(5,*) MDISLG
      WRITE(6,34) MDISLG
   34 FORMAT(T2,'TARGET VALUE FOR VEHICLE TRAVEL DIST. IS:',I5)
      IF((NEMCI+NCOCI).NE.0) GO TO 1000
C  MOVE TO MIN. OF 3RD OBJ. FN.,TD
      KPOINT=MSTAG
      JPOINT=MSTAG
C  RENEW INPUT DATA ARRAY,RHS, AND ADD 3RD OBJ. FN.---MIN. OF TD
   80 DO 40 I=1,MMAX
      DO 40 J=1,NMAX
      ATAB(I,J)=TTAB(I,J)
   40 CONTINUE
      ATAB(MSCO+1,1)=FLOAT(MDISLG)
      DO 45 I=1,MSTOPG
      KA=(KPOINT-1)*MSTOPG+I+1
      TTAB(MMAX,KA)=0.
   45 CONTINUE
      DO 41 I=1,NMAX
      ATAB(1,I)=TTAB(MMAX,I)
```

```
   41  CONTINUE
C  FIX A LINK DETERMINED AND SO MODIFY CONS. (1)
       IF(NCSM.EQ.O) GO TO 48
       KX=(JPOINT-1)*MSTOPG+KPOINT
       IF(KPOINT.GE.JPOINT) KX=KX-1
       DO 44 I=2,NMAX
       ATAB(JPOINT+1,I)=O.O
       IF(I.EQ.(KX+1)) ATAB(JPOINT+1,I)=1.O
       TTAB(JPOINT+1,I)=ATAB(JPOINT+1,I)
   44  CONTINUE
       JPOINT=KPOINT
C  DEFINE THE VARIANT INPUT DATA
   48  SOLMIN=FLOAT(MDISLG)
       PCTTOL=O.
       M=MMAX-1
       N=NMAX
       KA=2*MSTAG+1
       DO 50 I=2,KA
   50  IROW(I)=O
       KA=KA+1
       DO 55 I=KA,MSCO
   55  IROW(I)=-1
       IROW(MSCO+1)=-1
C  RUN THE SUBROUTINE MINT
       IOUT1=O
       CALL SMINT(JHANG)
C  COMPUTE THE DEGREES OF ACCOM. FN.
       CALL COMPT(TTAB,T)
       NCSM=NCSM+1
       LOPT=ZOPT+O.OO1
       KBB=LOPT-JPSLG
       IF(KBB.LE.O) GO TO 499
       IF(NCSM.GE.(MSTOPG-1)) GO TO 499
C  NEXT STATION TO VISIT IS DETERMINED
       DO 60 I=1,MSTOPG
       LQR=I
       IF(I.GE.KPOINT) LQR=LQR+1
       KA=(KPOINT-1)*MSTOPG+I
       BB=DABS(T(KA)-1.O)
       IF(BB.LE.O.OO1) GO TO 65
   60  CONTINUE
   65  KPOINT=LQR
       INEXT=ICLUST(IROWG,KPOINT)
       GO TO 80
C  MOVE TO MAX. OF 2ND OBJ. FN., FR
C  RENEW INPUT DATA ARRAY,RHS, AND 2ND OBJ. FN.---MAX. OF FR
 1000 DO 505 I=1,MMAX
       DO 505 J=1,NMAX
  505 ATAB(I,J)=TTAB(I,J)
       ATAB(MSCO+1,1)=FLOAT(MDISLG)
       DO 507 I=1,NZR1VR
  507 ATAB(1,I+1)=1.O
       IF(NEMCI.EQ.O) GO TO 518
       DO 51O I=1,NEMCI
       KA=MXX(I)+1
  51O ATAB(1,KA)=O.O
  518 IF(NCOCI.EQ.O) GO TO 519
       DO 511 I=1,NCOCI
       KA=MYY(I)+1
  511 ATAB(1,KA)=O.O
C  DEFINE THE VARIANT INPUT DATA
  519 SOLMIN=FLOAT(MSTAG)
       PCTTOL=O.
       M=MMAX-1
       N=NMAX
       KA=2*MSTAG+1
```

```
      DO 515 I=2,KA
  515 IROW(I)=0
      KA=KA+1
      DO 520 I=KA,MSCO
  520 IROW(I)=-1
      IROW(MSCO+1)=-1
C  RUN THE SUBROUTINE MINT
      CALL SMINT(JHANG)
      IF(JHANG.EQ.1) GO TO 500
C  COMPUTE THE DEGREES OF ACCOM. FN.
      CALL COMPT(TTAB,T)
  499 WRITE(6,450) IROWG
  450 FORMAT(T2,'** THE MOST SATISFACTORY ROUTE SEQ. OBTAINED FOR',
     *' CLUSTER',I3,' IS:')
      KOR=MSTAG+1
      WRITE(6,454) (IBB(I),I=1,KOR)
  454 FORMAT(/,T5,'ROUTE SEQ.:',12I4)
      WRITE(6,459) IBB(MSTAG+2),IBB(MSTAG+3),IBB(MSTAG+4)
  459 FORMAT(T5,'TOT. DIST.=',I5,5X,'TOT. DET.=',I5,5X,
     *'TOT. FULL. OF EM. SERV. & COND. DEP.=',I5)
  500 RETURN
C  INFORM THE VIOLATION OF RESTRICTION ON VEH. TRAV. DIST.
  919 IRTR=1
      WRITE(6,929)
  929 FORMAT(T2,'!ERROR! RESTRICTION ON VEH. TRAV. DIST. IS',
     *' VIOLATED!!',/,T2,'CONVERT TO THE PREVIOUS SUBSETS FORMATION!')
      RETURN
      END
```

APPENDIX B

DATA INPUTS FOR THREE TEST PROBLEMS

TABLE X

TEST PROBLEM 1

Station	x	y	Supply
1	151	264	1100
2	159	261	700
3	130	254	800
4	128	252	1400
5	163	247	2100
6	146	246	400
7	161	242	800
8	142	239	100
9	163	236	500
10	148	232	600
11	128	231	1200
12	156	217	1300
13	129	214	1300
14	146	208	300
15	164	208	900
16	141	206	2100
17	147	193	1000
18	164	193	900
19	129	189	2500
20	155	185	1800
21	139	182	700

Depot Coordinates (145, 215)

TABLE XI

TEST PROBLEM 2

Station	x	y	Supply	Station	x	y	Supply
1	218	382	300	16	119	357	150
2	218	358	3100	17	115	341	100
3	201	370	125	18	153	351	150
4	214	371	100	19	175	363	400
5	224	370	200	20	180	360	300
6	210	382	150	21	159	331	1500
7	104	354	150	22	188	357	100
8	126	338	450	23	152	349	300
9	119	340	300	24	215	389	500
10	129	349	100	25	212	394	800
11	126	347	950	26	188	393	300
12	125	346	125	27	207	406	100
13	116	355	150	28	184	410	150
14	126	355	150	29	207	392	1000
15	125	355	550				

Depot Coordinates (162, 354)

TABLE XII

TEST PROBLEM 3

Station	x	y	Supply	Station	x	y	Supply
1	37	52	7	26	27	68	7
2	49	49	30	27	30	48	15
3	52	64	16	28	43	67	14
4	20	26	9	29	58	48	6
5	40	30	21	30	58	27	19
6	21	47	15	31	37	69	11
7	17	63	19	32	38	46	12
8	31	62	23	33	46	10	23
9	52	33	11	34	61	33	26
10	51	21	5	35	62	63	17
11	42	41	19	36	63	69	6
12	31	32	29	37	32	22	9
13	5	25	23	38	45	35	15
14	12	42	21	39	59	15	14
15	36	16	10	40	5	6	7
16	52	41	15	41	10	17	27
17	27	23	3	42	21	10	13
18	17	33	41	43	5	64	11
19	13	13	9	44	30	15	16
20	57	58	28	45	39	10	10
21	62	42	8	46	32	39	5
22	42	57	8	47	25	32	25
23	16	57	16	48	25	55	17
24	8	52	10	49	48	28	18
25	7	38	28	50	56	37	10

Depot Coordinates (30,40)

VITA 2

YANG BYUNG PARK

Candidate for the Degree of

Doctor of Philosophy

Thesis: THE SOLUTION OF VEHICLE ROUTING PROBLEMS IN A MULTIPLE OBJECTIVE ENVIRONMENT

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Kwangju, Korea, November 13, 1952, the son of Dongsik Park and Geum Soon Bae. Married to Giebong Shin on January 7, 1980.

Education: Graduated from Kwangju Jai-Il High School, Kwangju, Korea, in February, 1972; received Backelor of Science degree in Industrial Engineering from Han Yang University, Seoul, Korea, in February 1978; received Master of Science degree in Industrial Engineering from Seoul National University, Seoul, Korea, in February, 1980; received Master of Science degree in Industrial and Management Systems Engineering from Pennsylvania State University in November, 1981; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in December, 1984.

Professional Experience: Research Assistant, Department of Industrial Engineering, Seoul National University, March 1979, to August, 1979; Teaching Assistant, School of Industrial Engineering and Management, Oklahoma State University, September, 1981, to May, 1982; Research Assistant, School of Industrial Engineering and Management, Oklahoma State University, September, 1982, to June, 1984.