

PLANT LOCATION-ALLOCATION PROBLEMS WITH
PRICE SENSITIVE STOCHASTIC DEMANDS

By

RASARATNAM LOGENDRAN
II

Bachelor of Science
University of Sri Lanka
Katubedda, Sri Lanka
1975

Master of Engineering
Asian Institute of Technology
Bangkok, Thailand
1980

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

Thesis
1984D
L832p
cop. 2



PLANT LOCATION-ALLOCATION PROBLEMS WITH
PRICE SENSITIVE STOCHASTIC DEMANDS

Thesis Approved:

M. Palmer Terrell

Thesis Adviser

Carl B. Ister

J. Leroy Fisher

Philip M. Wolfe

C. Patrick Keelbing

Norman D. Murkham

Dean of the Graduate College

PREFACE

This study is concerned with the transportation plant location-allocation problem in the presence of price sensitive stochastic demands. The purpose of this study is to develop a quantitative model for this problem and to establish solution methodologies that can provide decisions on location and allocation. Both capacitated and uncapacitated problems are analyzed for normal and uniform distribution of demands.

I wish to express my profound gratitude to my major adviser and chairman of my Ph.D. committee, Dr. M. Palmer Terrell, for his constant encouragement, guidance, and assistance rendered throughout the course of this study and during my doctoral program. Sincere thanks are due to my committee members, Dr. Carl B. Estes, Dr. Philip M. Wolfe, Dr. C. Patrick Koelling, and Dr. J. Leroy Folks for their interest and assistance.

I am also thankful to the staff of the University Computer Center for their cooperation extended during the development of the computer programs.

Thanks are due to Ms. Velda Davis and her staff for their excellent typing and suggestions concerning form.

Finally, I thank my parents for instilling within me the importance of an education and for their understanding and confidence in me as a student and as a son. To my brother, goes my deepest appreciation for his moral encouragement and understanding.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
The Problem Environment	1
Transportation vs. Transshipment Type Network Location Problems	3
Categorization of Transportation Plant Location Problems	4
The Research Problem	5
Price Sensitivity	7
Stochastic Demand	8
Research Objectives	8
Overview	8
Primary Objectives	9
Secondary Objectives	9
Decisions to be Provided by the Model	10
Summary	11
II. LITERATURE REVIEW	12
Deterministic Transportation Location Problems	12
Stochastic Transportation Location Problems	18
III. MODEL DEVELOPMENT	21
Basic Assumptions	21
On Some Deterministic Model Formulations of Plant Location Problems	23
Appropriateness of an Expected Value/Profit Model as Opposed to a Chance Constrained Model	32
Overview	32
Conclusions	33
Model Formulation for Problem Type 8	34
The Model	34
Proof for Concavity of $L(x_{ij})$	38
Example Problem	39
General Case	40
Optimality Conditions for Subproblems	41
Difficulties in Obtaining the Optimal Solution	43
Model Simplifications	44
Revised Model	45
Model for Problem Type 7	46
Potential Applications of the Model	48

Chapter	Page
IV. SOLUTION ALGORITHMS FOR THE REVISED MODEL	49
Introduction	49
Applicability of a Heuristic Algorithm for the Revised Model	51
Overview	51
Notations	52
Priority Rules	53
Algorithmic Steps	53
Weakness	56
Applicability of the Extreme Point Ranking Technique for the Revised Model	58
Theorem	58
Methodology	59
Notations	60
Algorithmic Steps	61
Conclusions	63
Applicability of the Branch-and-Bound Technique for the Revised Model	63
Summary	64
V. DATA GENERATION, COMPUTATIONAL EXPERIENCE, SENSITIVITY ANALYSIS AND MODEL VALIDATION	65
Introduction	65
Data Generation	65
The Need	65
Data Required	66
Price Sensitivity and Demand	66
Test Data	67
Computational Experience	68
Organization	68
Data for Test Runs	69
Type 1 Normally Distributed Demand	69
Type 2 Normally Distributed Demand	69
Uniformly Distributed Demand	70
Programming Concerns	70
Programming Difficulties	71
Notations	71
Computational Results for Problem Type 8	72
Computational Results for Problem Type 7	80
Sensitivity Analysis	87
Design of Test Runs	87
Type 1 Normally Distributed Demand	87
Type 2 Normally Distributed Demand	88
Uniformly Distributed Demand	89
Results of the Sensitivity Analysis for Problem Type 8	90
Results of the Sensitivity Analysis for Problem Type 7	94
Model Validation	94
Appropriateness of x_{ij}^*	97

Chapter	Page
Relationship Between p_{ij} and q_{ij}	98
Summary	98
VI. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS	100
BIBLIOGRAPHY	103
APPENDIX - FORTRAN PROGRAMS LISTING	108

LIST OF TABLES

Table	Page
1.1. Characterization of Problem Types	6
5.1. Results for Small Size Problems With Capacitated Plants and Type 1 Normally Distributed Demands.	73
5.2. Results for Small Size Problems With Capacitated Plants and Type 2 Normally Distributed Demands.	74
5.3. Results for Small Size Problems With Capacitated Plants and Uniformly Distributed Demands	75
5.4. Results for Medium and Large Size Problems With Capacitated Plants and Type 1 Normally Distributed Demands	78
5.5. Results for Medium and Large Size Problems With Capacitated Plants and Type 2 Normally Distributed Demands	78
5.6. Results for Medium and Large Size Problems With Capacitated Plants and Uniformly Distributed Demands .	79
5.7. Results for Small Size Problems With Uncapacitated Plants and Type 1 Normally Distributed Demands	81
5.8. Results for Small Size Problems With Uncapacitated Plants and Type 2 Normally Distributed Demands	82
5.9. Results for Small Size Problems With Uncapacitated Plants and Uniformly Distributed Demands	83
5.10. Results for Medium and Large Size Problems With Uncapacitated Plants and Type 1 Normally Distributed Demands	84
5.11. Results for Medium and Large Size Problems With Uncapacitated Plants and Type 2 Normally Distributed Demands	85
5.12. Results for Medium and Large Size Problems With Uncapacitated Plants and Uniformly Distributed Demands	86

Table	Page
5.13. Results of Sensitivity Analysis for Problems With Capacitated Plants and Type 1 Normally Distributed Demands	91
5.14. Results of Sensitivity Analysis for Problems With Capacitated Plants and Type 2 Normally Distributed Demands	91
5.15. Results of Sensitivity Analysis for Problems With Capacitated Plants and Uniformly Distributed Demands .	92
5.16. Results of Sensitivity Analysis for Problems With Uncapacitated Plants and Type 1 Normally Distributed Demands	95
5.17. Results of Sensitivity Analysis for Problems With Uncapacitated Plants and Type 2 Normally Distributed Demands	96
5.18. Results of Sensitivity Analysis for Problems With Uncapacitated Plants and Uniformly Distributed Demands	96

LIST OF FIGURES

Figure	Page
3.1. Plot of Plant Cost vs. Total Produced	24
4.1. Flow Chart for Heuristic Algorithm	57

CHAPTER I

INTRODUCTION

The Problem Environment

In the past two decades or so, investigations have been carried out extensively in the area of private and public sector location problems. A similarity exists between the two in that they both share the objective of maximizing or minimizing some measure of effectiveness to the owners while at the same time satisfying constraints on demands and other conditions. However, the formulations of these objectives and constraints for the private and public differ considerably. The most important difference is that the ownership conditions are different. Although several issues, including some of a non-economic nature, can be considered for making decisions on private sector locations, a reasonably accurate statement of the objective of the location decision is the minimization of cost or maximization of profit to the private owners; ReVelle et al. [48]. On the other hand, the public sector location decisions are made in response to a different set of owners, to a society as a whole, and the objective here is to maximize a benefit or to minimize a cost which may be unquantifiable in dollar terms. The intent of this research is specific to the private sector, and further discussion pertinent to public facility locations will not be addressed.

Problems in location analysis can be broadly categorized into three major structural groups [48]. They are:

1. Location on a plane,
2. Location on a network, and
3. Location only on central points of a network

Each group can be categorized as follows:

1. Location on a plane: A continuous or an infinite solution space is considered in this case. In other words, central facilities may be located anywhere on the plane and are confined to neither nodes of the network nor to the points on the links between the nodes. The criterion for evaluation is usually to minimize a distance measure. Two distance measures commonly used are rectilinear and Euclidean distances; Francis and White [24].
2. Location on a network: The solution space for this group consists of points on the network. It includes both nodes and points on the arcs which join the nodes. Commonly used criterion for evaluation is to minimize either a distance measure or a time measure along the network.
3. Location only on central points of a network: The plant/source and warehouse/distribution center location problems are typical of this characterization. A number of potential locations where the warehouses or plants can be built are predetermined. The facilities,¹ if built, will satisfy the known demands of a

¹The term facilities is broadly used for plants/sources or warehouses/distribution centers.

number of demand centers/markets for a certain product or multiple of products.

Transportation vs. Transshipment Type Network Location Problems

This research focuses on facilities location and, therefore, belongs to the third group as detailed above. Interestingly enough, the problem of locating plants on a network can be classified further as a transportation type network location problem or a transshipment type network location problem. The former pertains to the location of plants or sources in a subset or in the entire set of predetermined potential locations. These plants, when located, supply product(s) to existing demand centers that are also located on the network. In this situation, the demand in the area surrounding each center is assumed to be concentrated at a point. The latter situation pertains to the location of warehouses or distribution centers in a subset or in the entire set of predetermined potential locations that serve as intermediary between plants/sources and the demand centers/markets. Under this set up, product(s) are shipped from the plants to those warehouses whose locations are to be determined and are eventually distributed from these warehouses to the demand centers.

The key point here is that both problems adequately represent two different situations commonly encountered in facilities location depending upon the type of management involved with the distribution network. On the one hand, if the production and distribution are handled by the same management, the transportation type location problem is an adequate representation. On the other hand, if the production and distribution

are associated with different managements, then the transshipment type location is a good representation. In the latter situation, the term distribution includes the tasks of locating warehouses/distribution centers and eventually shipping (allocating) the product(s) in appropriate quantities to the demand centers.

The scope of this study is limited to a situation where both production and distribution of the product(s) are controlled by the same management. Thus, a transportation type location problem will be investigated. Consequently, the discussion to follow will refer to specific terminologies such as location of plants and eventual distribution of product(s) from the plants to the demand centers.

Categorization of Transportation Plant

Location Problems

The state of the art specific to transportation plant location problems evidences the fact that all investigations in this area can be categorized into different types of problems with the aid of two sets of parameters that identify the characteristics of each problem. Let these sets be defined as S_1 and S_2 . The parameters contained in set S_1 are:

$$S_1 = \begin{cases} \text{demand} & - \text{either } \underline{\text{deterministic}} \text{ or } \underline{\text{stochastic}} \\ \text{price} & - \text{either } \underline{\text{price sensitive}} \text{ or } \underline{\text{price insensitive}} \\ \text{facility} & - \text{either } \underline{\text{capacitated}} \text{ or } \underline{\text{uncapacitated}} \end{cases}$$

While those contained in set S_2 are:

$$S_2 = \begin{cases} \text{product} & - \text{either } \underline{\text{a single product}} \text{ or } \underline{\text{multi-product}} \\ \text{period} & - \text{either } \underline{\text{a single period}} \text{ or } \underline{\text{multi-period}} \end{cases}$$

There are 32 (2^5) different types of transportation plant location problems that can be perceived using the suggested categorization. Table 1.1 presents the 32 different types of problems along with the names of researchers/authors who have investigated each specific type. Problem types 1 through 6, 9, 10, and 17 are the only ones that have been attempted and reported in the literature. A close look at the potential types of problems not investigated led to the development of the research problem outlined below.

The Research Problem

Almost all studies presented in Table 1.1 are in the area of transportation plant location problems with a few exceptions of transshipment plant location problems. In the latter case, those cited had a reduced model formulation using a transportation type solution methodology. Among all of these types, investigations have been heavily concentrated on types 1 and 2. Although types 3, 4, 5, and 6 have been investigated, not many additional attempts have been made to either discover an alternate solution methodology or to improve upon the existing one. This is particularly true when the dimension of stochastic demand is incorporated into the problem as compared to incorporating price sensitivity. The reason is that when stochastic parameters are incorporated into a problem, one looks forward to possibly transforming the stochastic problem into an equivalent deterministic problem. Even if this is accomplished, the degree of complexity introduced into the transformed formulation makes it difficult in most instances, if not impossible, to seek an exact technique. It is apparent that the key to the problems of types 7 and 8 are the dimensions of price sensitivity and stochastic

TABLE 1.1

CHARACTERIZATION OF PROBLEM TYPES

	Price insensitive, deterministic demand, and uncapacitated facilities	Price insensitive, deterministic demand, and capacitated facilities	Price sensitive, deterministic demand, and uncapacitated facilities	Price sensitive, deterministic demand, and capacitated facilities	Price insensitive, stochastic demand, and uncapacitated facilities	Price insensitive, stochastic demand, and capacitated facilities	Price sensitive, stochastic demand, and uncapacitated facilities	Price sensitive, stochastic demand, and capacitated facilities
	1	2*	3	4	5	6	7	8
Single product- Single period	Brown [5] Cornuejols [9] Drysdale and Sandiford [14] Efroymsan and Ray [15] Erlenkotter [20] Feldman, Lehrer, and Ray [22] Khumawala [40] Kuehn and Hamburger [41] Manne [43] McGinnis [45] Spielberg [53] Spielberg [54] Warszawski [59]	Davis and Ray [13] Ellwein and Gray [17] Marks [44] Sa [49] Yagiz [65]	Erlenkotter [19] Hansen and Thisse [32]	Truscott [57] Truscott [58]	Gonzalez-Valenzuela [29] Jucker and Carlson [36]	Gonzalez-Valenzuela [29]	†	†
Single product- Multi period	Bhalla [4] Kelly [39] Warszawski [60]	Cho [7] Eschenbach [21] Karanicolas [37]						
Multi product- Single period	Karkazis and Boffey [38] Neebe and Khumawala [46] Warszawski [60] Warszawski and Peer [61]							
Multi product- Multi period								

*Refers to problem types

†Possible problem types for investigation

demand. Thus the need for having to simultaneously incorporate these two dimensions in the location problem should be recognized.

Price Sensitivity

Models belonging to types 1 and 2 for plant location have typically assumed that plants specified by location are to be established to meet fixed demands at minimum cost. These are static models in the sense that they exclude the possibility that demands may be influenced by pricing; Erlenkotter [19]. The per unit variable cost of a product is composed of the per unit plant cost, production cost, and the transportation cost.²

The per unit plant cost is dependent upon the operational and maintenance characteristics of the plant, while the per unit production cost is dictated by the material and labor costs. The per unit transportation cost, however, is dependent upon service capabilities such as delivery times, etc. of a plant. As a consequence, the total variable cost per unit of product received at a demand center varies depending upon the plant from which the shipments are made. Thus, the prices established for a product received at a demand center will vary considerably depending from which plant the shipments are made. This very fact reinforces the need for having to consider the revenue generation effects or a profit maximization approach to facility location as opposed to the minimization of total cost approach to facility location exhibited by static models. "Elasticity of demand" is an important economic concept descriptive of a drop in demand with increase in

²See page 23.

price. The question is 'can deterministic estimates of demand for a product be made at different price levels.' And the answer is possibly 'no' in most realistic situations. Implicit in this is the assumption that the product is not produced to order and supplies are made before the actual demand becomes known.

Stochastic Demand

When the demand for a certain commodity is not known with certainty, the solution is to treat it as a random variable. For the purpose of analysis, this can be assumed to follow a known distribution with a given mean and variance. At a given demand center, it will not be unrealistic to assume that the demand generated exhibits the same distribution, yet independent, with varying mean and/or variance at different price levels. For example, it is logical to have a drop in mean, but the same variance with the increase in price. Since prices to be established are determined by the management and are within its control, it will be assumed that demand is the only source of uncertainty introduced in this research.

Research Objectives

Overview

The objectives delineated below are divided into two sections, the primary and the secondary objectives. The primary objectives focus on a suitable model development for the capacitated problem and on the determination of an appropriate solution algorithm. The secondary objectives are to develop a computer routine based on the solution algorithm and to carry out a sensitivity analysis by varying the appropriate parameters.

Additionally, a solution methodology for the uncapacitated problem will be deduced from the solution algorithm determined for the capacitated problem. More specifically, the primary and secondary objectives can be documented as follows:

Primary Objectives

The primary objectives of this research are:

1. To develop and validate a suitable model for a single product--single period capacitated plant location-allocation problem that includes the dimensions of price sensitivity and stochastic demand.³
2. To establish an appropriate solution algorithm for this model.

In order to accomplish the primary objectives, several secondary objectives have been included.

Secondary Objectives

The secondary objectives of this research are:

1. To develop a computer routine based on the algorithm determined above. This routine is intended to solve a problem of reasonable size and to provide the management with decisions on both location and allocation.⁴
2. To carry out a sensitivity analysis by performing test runs with different values of demands and/or prices in order to determine the changes in model responses measured by the optimal objective function values. The demand can be modified by

³Type 8 in Table 1.1.

⁴See page 10.

varying either the mean and/or variance parameters of the demand distribution. Both normal and uniform distributions have been found to be good representations of variability in demand; Gonzalez-Valenzuela [29]. Thus, a sensitivity analysis will also be performed by changing the demand distributions.

3. To deduce a solution methodology for an uncapacitated plant location-allocation problem described by type 7 in Table 1.1 from the algorithm developed for problem type 8.

Decisions to be Provided by the Model

The model for this research problem should have the capability of providing the management with a sequence of decisions related to the following four criteria. They are:

1. Decision on location - where to build the plant
2. Decision on capacity - how big the plant should be
3. Decision on production - how much to produce in each plant
4. Decision on allocation - The amount of production supplied to the demand centers from the plants.

Capacity decisions are frequently dictated by resource limitations on material, labor, etc. faced by the management in each of the potential locations where the plant is to be built. In this connection, it is possible for the management to make a reasonably accurate estimate of the maximum available capacity given a potential location. Thus, capacity decisions can be made a priori rather than have the model provide them as output.

The problem is specific to a single product-single period situation. Being single period it will not be realistic to consider

maintaining inventories at the plant. In effect, the total amount produced in a given plant is allocated to the demand centers as appropriate. As a consequence, the model is intended to explore the possibilities of arriving at decisions on location and allocation in the presence of price sensitive stochastic demands.

Summary

The material presented in this chapter has strongly supported the fact that both price sensitivity and stochastic demand are two important dimensions to be taken into consideration when analyzing a complete single product-single period plant location problem. Each of these dimensions have, however, been included separately in problem types 3 and 4 and types 5 and 6. The intent of this research is, therefore, to incorporate both dimensions and investigate a more comprehensive problem evidenced by types 7 and 8 in Table 1.1.

CHAPTER II

LITERATURE REVIEW

The investigation of this research is limited to a single product-single period situation. Therefore, the review documented in this chapter will explore studies related to problem types 1 through 6. In particular, the solution methodologies attempted to solve these problems types are documented. As such, those related to deterministic transportation location problems and those related to stochastic transportation location problems are broadly described.

Deterministic Transportation Location Problems

Simulation is one of the approaches attempted to solve the simple plant location problem described by type 1. It has been successfully used by Gerson and Maffei [28] and Shycon and Maffei [52]. The underlying basis for this approach is that knowing the number of plants, one could randomly generate locations for the plants and evaluate them in order to find the best solution for the assumed number of plants. The major drawback of this approach is that an optimum solution is not guaranteed.

Baumol and Wolfe [1] first formulated Problem 1A¹ as a transportation problem with some fixed charges. The solution technique is based on an iterative methodology. At each stage, it required setting up and

¹See page 25.

solving a transportation problem in order to obtain an improvement on the objective function value of the original problem. The major shortcomings of this approach are twofold; it does not ensure optimality, and even on medium sized problems the computation time is considerably high.

The inherent difficulty in solving the simple plant location problem is essentially due to its combinatorial structure. From m potential locations for plants, there are $2^m - 1$ feasible locations to be considered, each corresponding to a subset or the entire set. In an actual problem, the number of potential locations can range anywhere between 10 to 40 with more demand centers than the locations under consideration. Evidently, in such cases determining an optimum subset of plants in the system can be overly difficult, if not impossible. Thus, the characterization of solution methodologies are based on the degree of computing efficiency attained. This very fact has led to the development of several heuristics to produce good solutions; Kuehn and Hamburger [41], Manne [43], Feldman, Lehrer and Ray [22], and Drysdale and Sandiford [14].

The mechanism behind the operation of these heuristics is basically the same. Each of the 2^m subsets of solutions are contained on the lattice points of a unit hypercube. The approach proceeds by starting at an arbitrary lattice point of this unit hypercube, and then moving in one dimension to another lattice point. Such movement can either be an addition of a plant to or a dropping of a plant from the subset under consideration. The former is commonly referred to as "add heuristics" while the latter as "drop heuristics".

²This includes the infeasible solution with all plants being closed.

Kuehn and Hamburger [41] proposed a solution methodology that uses the following three principal add heuristics in the main routine.

1. Locations with promise will be at or near concentrations of demand.
2. Near optimum plant systems can be developed by locating plants one at a time, adding at each stage of the analysis that plant which produces the greatest cost savings for the entire system.
3. Only a small subset of all possible plant locations needs to be evaluated in detail at each stage of the analysis to determine the next plant to be added.

If another plant cannot be added without increasing the total cost, a solution is reached in the main routine. A "bump and shift" routine is then used to improve the above solution by eliminating (bumping) any plant which is no longer economical because some of the demand centers originally assigned to it are now serviced by plants located subsequently. Thereafter, economics of shifting each plant from its currently assigned location to the other potential sites within its territory are also considered.

Manne [43] suggests both an add and a drop be effected one dimensionally on the starting solution. For each subset of locations encountered, a demand center is simply assigned to that plant for which the sum of variable cost and fixed cost is a minimum. If an improvement is found, the new lattice point is used as a starting point to continue further iteration. The algorithm terminates when no further improvement can be realized.

A more general form for the plant cost, that of a continuous concave function is assumed by Feldman, Lehrer, and Ray [22]. Their approach is

based on drop heuristics in that it starts with all plants existing and drops out plants. The algorithm terminates when no further cost reductions can be obtained by eliminating any other plants.

Efroymsen and Ray [15] were the first to propose an exact algorithm for solving Problem 1C³. They used an implicit enumeration technique widely known as branch-and-bound. The technique incorporated a selective enumeration, guided at each stage by a bound on the value of the objective function obtained at that stage.

For the same problem, Spielberg [53] investigated the improvements on computational efficiency attained by intelligent choice of starting points. A study was later reported by Spielberg [54] on the same problem including configuration constraints.

Results reported by Khumawala [40] show that the use of efficient branching decision rules lead to the attainment of greater computational efficiency in solving Problem 1C.

Warszawski [59] has proposed an exact solution technique for Problem 1B⁴.

Brown [5] investigated a generalization of the simple plant location problem. The zero-one decision variables are replaced by general integer variables in this formulation. Two enumerative algorithms have been proposed. Each uses a heuristic procedure to obtain a good initial solution.

The same problem was formulated as a mixed integer programming problem by McGinnis [45]. Both exact and approximate solution procedures are proposed. The former employs implicit enumeration, ruling out

³See page 27.

⁴See page 26.

non-optimal solutions with the aid of Bender's constraints. The latter is based on maintaining feasibility and upper and lower bounds while trying to improve the solution.

Erlenkotter [20] presents a methodology for the simple plant location problem that is based on a linear programming dual formulation. The optimal dual solutions are produced by a simple ascent and adjustment procedure. These dual solutions often corresponded directly to optimal integer primal solutions. If integral values are not met, then the solution process is completed by a branch-and-bound procedure.

Another interesting formulation of a class of location problems that is closely related to the simple plant location problem is due to Cornuejols [9]. It is formulated as a problem requiring the sum of the profits received in all centers be maximized, subject to an additional configuration constraint. Heuristics such as "greedy" heuristic and "interchange" heuristic are proposed as solution techniques and appropriate error or deviation from the optimal value is given.

Davis and Ray [13] used a branch-and-bound algorithm for solving the type 2 problem. The algorithm employed a decomposition technique to solve the dual of the associated continuous problem in each branch-and-bound iteration. Moreover, the dual of the "sub-problem" at any decomposition iteration is conveniently solved using an out-of-kilter algorithm identifying it as a capacitated transportation problem.

Sa [49] investigated the capacitated plant location problem both by an exact approach and an approximate approach. The exact approach is based on a branch-and-bound technique somewhat similar to Efraymson and Ray's [15] but adds two pruning rules before solving the subproblems. One is a dominance test on the subtrees and the other is an upper bound

on the fixed cost for the optimal location based on Gray's [30] cut. The approximate approach embodies a routine that is based on the add heuristics of Kuehn and Hamburger [41] and the drop heuristics of Feldman, Lehrer, and Ray [22].

Marks [44] investigated a more general formulation of the problem, namely a fixed-charge transshipment plant location problem with capacity constraints. Precisely, the investigation is directed toward the location of warehouses intermediate to the plants and demand centers. The solution technique is based on a network algorithm. A capacity constraint and a linear cost function are ascribed to each plant location by adding a capacitated node for each plant in the network. Though the solution procedure proposed is more general, a reduced version of it is applicable to the type 2 problem.

Ellwein and Gray [17] have presented a solution procedure for problem type 2 including configuration constraints. It uses an enumerative search scheme in conjunction with adaptive bounds on the fixed costs, and constraints based on the dual variables to reduce the feasible solution set to a manageable size.

Yağiz [65] employed an efficient heuristic method, which yielded "good" solutions very rapidly. These had been in most instances optimal, and if not a Balas-type single-branch search method is used to attain optimality.

Hansen and Thisse [32] were the first to investigate the type 3 problem that included the dimension of price sensitivity. Their approach is to reformulate this problem in the form of Problem 1B and to use any of the available solution techniques for the latter to solve the reformulated problem. The same methodology was later addressed by Erlenkotter

[19], though his work was mainly focused on public sector models.

A solution algorithm for solving a transshipment plant location problem with price sensitive demand is documented by Truscott [57]. A similar algorithm applicable for a type 4 problem is later addressed in Truscott [58]. All possible zero-one solutions, either explicitly or implicitly, are taken into account in this algorithm. Application of infeasibility and non-optimality tests enabled eliminating groups of complete solutions on the basis of the results from partial solutions.

Stochastic Transportation Location Problems

Problem type 5 was first investigated by Gonzalez-Valenzuela [29]. He considered both a chance constrained formulation and a two-stage stochastic programming formulation of this problem. However, the models did not include the effect of price sensitivity on demand. This led to the formulation of models that only accounted for cost effects and not the revenue generation aspects.

A transformation of the chance constrained problem provided an equivalent deterministic problem which is then solved by one of the existing algorithms [40] for a simple plant location problem. Similarly, the stochastic programming problem for the uncapacitated case is also transformed to an equivalent deterministic problem. An existing method [40] is used to solve this deterministic problem.

Jucker and Carlson [36] studied the same problem in a different context. Although they used a profit maximization approach, there is no dependency between the price and quantity demanded. Thus, it was possible to decompose the entire problem into two simpler problems. The latter uses values determined from the first subproblem, which is then solved as

a regular simple plant location problem with the aid of one of the many efficient methods available [15], [40].

Problem type 6 investigated by Gonzalez-Valenzuela [29] is similar to that of type 5 except that they are now analyzed in the presence of capacitated plants. As in type 5, both a chance constrained problem and a two-stage stochastic programming problem are investigated. The chance constrained problem is transformed into an equivalent deterministic problem, and an existing solution technique [16] is used. However, for the stochastic programming problem, the transformation employed led to the establishment of only an approximate problem. It is then solved using one of the existing methods [16].

Given a location vector $y = (y_i)$, any plant location problem with stochastic demand can be reduced to a transportation problem with stochastic demand. A review of work undertaken in this area is documented below.

Ferguson and Dantzig [23] developed a technique for solving a stochastic transportation problem when the probability distribution of demand is discrete. Their approach relied on transforming the stochastic problem into an equivalent deterministic problem, and then employing decomposition for solving the latter.

Elmaghraby [18] established an excellent iterative technique for solving a transportation problem with continuous demand distribution. The technique yields a global minimum. The necessary and sufficient conditions for attaining it are accomplished by the use of Lagrange multipliers.

Williams [62] proposed a solution methodology for solving a stochastic transportation problem. His approach first determines the

total expected costs of several proposed solutions on a weighted average basis. These are then used to solve a linear programming problem employing a decomposition algorithm. The inherent difficulty with this treatment is that the joint cumulative distribution function of the random demand in each of the demand centers must be known.

Szwarc [55] studied a transportation problem with stochastic demand that focused on minimizing total transportation costs plus expected penalty costs due to both over and undersupplies. The solution technique used a controlled linear approximation in place of the original cost function.

The technique that uses bounds established on the optimal supply quantities is documented by Wilson [63], [64]. This led to reducing a stochastic transportation problem to a deterministic linear approximating problem which is then solved as a regular capacitated transportation problem.

CHAPTER III

MODEL DEVELOPMENT

The material presented in this chapter elucidates the sequence of steps taken toward accomplishing the development of a suitable model for the research problem. First, the basic assumptions made in regard to the model formulation are documented. Thereafter, some deterministic model formulations of plant location problems are discussed. The notations followed in this section are conveniently used later in the formulation of the model for the research problem. Then the appropriateness of an expected value/profit model as opposed to a chance constrained model for the research problem is discussed. The model formulation for problem type 8 is described in detail. This includes the model, its characteristics, and the need for simplifications. A suitable model for problem type 7 is then deduced from the model developed for problem type 8. Finally, the chapter is concluded with a description of potential applications of the model.

Basic Assumptions

The assumptions described below are comprehensive except for those on price sensitivity. The assumptions related to price sensitivity are described in Chapter V.

1. Investigation is limited to a single-product, single-period situation.
2. The potential locations for the plants are known. Under

optimality, the model will determine either a subset or the entire set from these predetermined locations as potential sites for building the plants.

3. The potential demand centers are known. The demand surrounding a center is assumed to be concentrated at one point.
4. The maximum capacity that can be produced by each plant is assumed to be known. For an uncapacitated plant an infinite capacity is assumed. More specifically, in the worst case, each uncapacitated plant should have the capability of supplying all of the demand centers if necessary.
5. There is a fixed cost for each plant built at a potential site which is independent of the amount produced. For a single-period situation, this can be interpreted as the amortized construction cost over a specified life of the plant.
6. There is a per unit plant cost or per unit variable cost associated with each plant built. This cost is assumed to exhibit a linear variation with the amount produced at each plant.
7. Additionally, there is a per unit variable cost, namely the unit production cost which is assumed linear on the amount of product produced at each plant. In particular, this cost is contributed by both material and labor inputs for making the product.
8. There is also a per unit variable cost or unit transportation cost due to allocation which is assumed linear on the shipments made from a given plant to the demand center.
9. The demand is the only source of uncertainty introduced in this research.

10. The supplies to a given demand center from a plant are made before the actual demand becomes known.

On Some Deterministic Model Formulations of Plant Location Problems

The simplest among all types of problems is problem type 1, which is commonly called the "simple plant location problem", [15], [40], [53], [59]. This problem considers the production of a single product at a given set of m plants with unlimited capacity and its distribution to n demand centers. A given demand center j ; $j = 1, \dots, n$ has a demand for d_j units of a product, and a particular plant i ; $i = 1, \dots, m$ may or may not be opened. If it is closed, then the cost associated with plant i is assumed to be 0. Otherwise the plant cost $F_i(x_j)$ is composed of the following components.

1. The fixed cost, $f_i \geq 0$,
2. The per unit plant cost, λ_i ,
3. The per unit production cost, v_i , and
4. The per unit transportation cost, t_{ij} .

An illustration is given in Figure 3.1 on the following page.

$$F_i(x_j) = \begin{cases} f_i + \lambda_i x_j & \text{if the facility exists} \\ 0 & \text{if it does not} \end{cases} \quad (3.1)$$

Thus, the total variable cost (r_{ij}) for a unit of product produced at plant i and shipped to demand center j is:

$$r_{ij} = t_{ij} + v_i + \lambda_i \quad (3.2)$$

Let x_{ij} be the amount of product shipped from plant i to demand center j . It is also assumed that plant i produces only the necessary

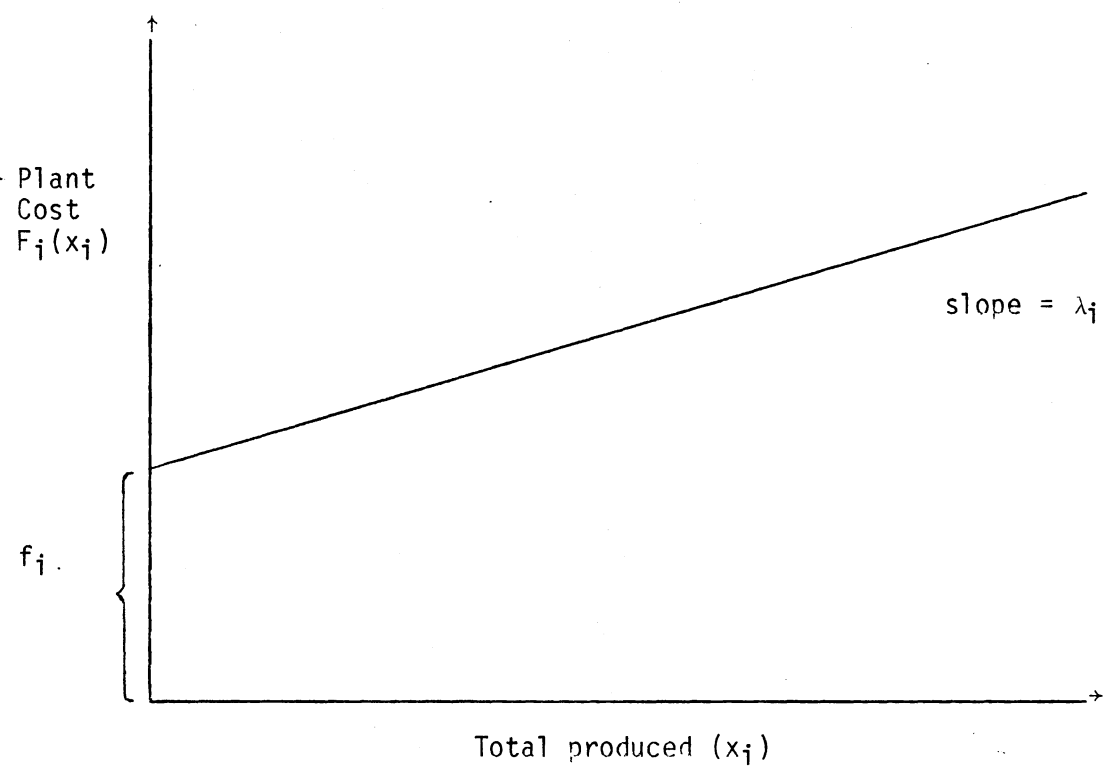


Figure 3.1. Plot of Plant Cost vs. Total Produced

quantity $x_i = \sum_{j=1}^n x_{ij}$, but is otherwise capable of satisfying all demands. y_i is a binary variable (0, 1) where 1 indicates the plant i being open while 0 for it being closed. The problem can, therefore, be formulated as a cost minimization function given by:

Problem 1A:

Minimize

$$Z = \sum_{i=1}^m \sum_{j=1}^n r_{ij} x_{ij} + \sum_{i=1}^m f_i y_i \quad (3.3)$$

subject to:

$$\sum_{i=1}^m x_{ij} = d_j \quad ; \quad j = 1, \dots, n \quad (3.4)$$

$$\sum_{j=1}^n x_{ij} \leq y_i \gamma_i \quad ; \quad i = 1, \dots, m \quad (3.5)$$

$$y_i = 0 \text{ or } 1 \quad (3.6)$$

$$x_{ij} \geq 0 \quad (3.7)$$

The γ_i is introduced in constraint (3.5) to indicate suitable upper bounds on the supplies made to the demand centers. For instance, it can be set to $\sum_{j=1}^n d_j$ independently of i . The constraint (3.5) also ensures that no supplies can be made from plants that are closed, i.e., $y_i = 0$.

Notice that Problem 1A is an ordinary transportation problem for a given location vector $y = (y_1, \dots, y_m)$. The reason is that, for a given y , the second part of the objective function $\sum_{i=1}^m f_i y_i$ is known. What remains then is the total variable cost component given by

$\sum_{i=1}^m \sum_{j=1}^n r_{ij} x_{ij}$ which is minimized if for a given demand center, the route (i, j) with the least r_{ij} is chosen.

By incorporating the above idea, the problem can be reformulated to give a much simpler formulation. Let s_{ij} represent the fraction of demand d_j satisfied by plant i .

$$\therefore s_{ij} = \frac{x_{ij}}{d_j} \quad (3.8)$$

As noted above, in the optimal solution, i.e., when the location vector y is fixed, the values taken by s_{ij} will either be 0 or 1. Since the set of constraints (3.5) is incorporated to ensure that no supplies are made from closed plants, those can be replaced by

$$\sum_{j=1}^n s_{ij} \leq n y_i \quad (3.9)$$

The total cost Z can now be expressed in terms of s_{ij} by replacing the unit variable cost r_{ij} by $c_{ij} = r_{ij} \cdot d_j$. Thus the reformulated problem is:

Problem 1B:

Minimize

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} s_{ij} + \sum_{i=1}^m f_i y_i \quad (3.10)$$

subject to:

$$\sum_{i=1}^m s_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.11)$$

$$\sum_{j=1}^n s_{ij} \leq n y_i \quad ; \quad i = 1, \dots, m \quad (3.12)$$

$$y_i = 0 \text{ or } 1 \quad (3.13)$$

$$s_{ij} \geq 0 \quad (3.14)$$

A modified version of Problem 1B can be obtained by incorporating a few "prohibited" routes (i, j) into the model. These routes will have large costs such as $c_{ij} = M$, and the n in constraints (3.12) will be replaced by n_i which is equal to the number of $c_{ij} \neq M$ in row i of the cost matrix. The resulting problem is, therefore, represented as:

Problem 1C:

Minimize

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} s_{ij} + \sum_{i=1}^m f_i y_i \quad (3.15)$$

subject to:

$$\sum_{i \in N_j} s_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.16)$$

$$\sum_{j \in P_i} s_{ij} \leq n_i y_i \quad ; \quad i = 1, \dots, m \quad (3.17)$$

$$y_i = 0 \text{ or } 1 \quad (3.18)$$

$$s_{ij} \geq 0 \quad (3.19)$$

where

N_j = set of plants which can supply receiving site j

P_i = set of receiving sites that can be supplied by plant i

n_i = number of elements in P_i

To summarize, Problem 1B is a simplified version of Problem 1A while Problem 1C, though a restricted version of Problem 1B, has not changed the structure of the latter significantly.

Problem type 2, [13], [17], [49], is generally described as the capacitated plant location problem. This is more constrained than type 1 because limitations on capacity are now imposed. They are of the following form:

$$\sum_{j=1}^n x_{ij} \leq A_i y_i \quad ; \quad i = 1, \dots, m \quad (3.20)$$

where A_i is the capacity of the plant at i . These constraints will replace constraints (3.5) in Problem 1A, thus resulting in:

Problem 2A:

Minimize

$$Z = \sum_{i=1}^m \sum_{j=1}^n r_{ij} x_{ij} + \sum_{i=1}^m f_i y_i \quad (3.21)$$

subject to:

$$\sum_{i=1}^m x_{ij} = d_j \quad ; \quad j = 1, \dots, n \quad (3.22)$$

$$\sum_{j=1}^n x_{ij} \leq A_i y_i \quad ; \quad i = 1, \dots, m \quad (3.23)$$

$$y_i = 0 \text{ or } 1 \quad (3.24)$$

$$x_{ij} \geq 0 \quad (3.25)$$

Another dimension that is sometimes added in the investigation of a type 2 problem is the configuration constraints. Those are represented by:

$$\sum_{i \in S_t} y_i \leq r_t \quad ; \quad t = 1, \dots, p \quad (3.26)$$

This defines a set of p system configuration constraints where S_t is a subset of the m source locations and $r_t < m$. A common system configuration constraint of this type which restricts the number of locations used is:

$$\sum_{i=1}^m y_i \leq r_1 \quad (3.27)$$

where

$$r_1 < m$$

Type 3 problem, [19], [32], includes the key dimensions of price sensitivity into a type 1 problem. The formulation of a general model for this problem is:

Problem 3A:

Maximize

$$Z = \sum_{j=1}^n p_j(d_j) \cdot d_j - \sum_{i=1}^m \sum_{j=1}^n r_{ij} x_{ij} - \sum_{i=1}^m f_i y_i \quad (3.28)$$

subject to:

$$\sum_{i=1}^m x_{ij} = d_j \quad ; \quad j = 1, \dots, n \quad (3.29)$$

$$\sum_{j=1}^n x_{ij} \leq M y_i \quad ; \quad i = 1, \dots, m - 1 \quad (3.30)$$

$$y_i = 0 \text{ or } 1 \quad (3.31)$$

$$x_{ij} \geq 0 \quad (3.32)$$

Where $p_j(d_j)$ is the identical price for each unit of demand at location j and M is a large positive number. The reason for considering only $(m - 1)$ potential locations in the above formulation is the following. From a fact observed by Samuelson [50], it is noted that a solution to Problem 1B is a necessary condition for optimality given any fixed values for the demands in Problem 3A. Particularly, this holds true for optimal d_j^* , for which it has already been proven that the demand will be met from a single plant. This enabled reformulating Problem 3A in the form of Problem 1B which included a dummy plant indexed m with $c_{mj} = 0$ and $f_m = 0$. As a consequence, any approach used for solving Problem 1B can also be used for solving the reformulated problem.

The formulation of a model for problem type 4, [58], is somewhat different from the previous three because of having to incorporate price sensitivity under capacitated plants. The model for this problem can be documented as:

Problem 4A:

Maximize

$$Z = \sum_{i=1}^m \left\{ \sum_{j=1}^n [(p_{ij} q_{ij} - R_{ij}) z_{ij}] - f_i y_i \right\} \quad (3.33)$$

subject to:

$$\sum_{i=1}^m z_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.34)$$

$$\sum_{i=1}^m y_i \leq r_1 \quad (3.35)$$

$$\sum_{j=1}^n q_{ij} \cdot z_{ij} \leq A_i y_i \quad ; \quad i = 1, \dots, m \quad (3.36)$$

$$z_{ij}, y_i = 0 \text{ or } 1 \quad (3.37)$$

where

p_{ij} = unit price received at demand center j when supplied by plant i

q_{ij} = the number of units demanded at center j when supplied by plant i

$R_{ij} = q_{ij} \cdot r_{ij}$ = the cost of throughput of q_{ij} units at plant i and delivery of q_{ij} units from plant i to demand center j

$$z_{ij} = \begin{cases} 1 & \text{if plant } i \text{ supplies demand center } j \\ 0 & \text{otherwise} \end{cases}$$

One interesting feature of this formulation is that a set of allocation constraints (3.34) requiring that each demand center be supplied by exactly one facility is incorporated. With capacitated plants, this feature eliminates possible input differences associated

with heterogenous demand effects of multiple servers/plants. This is essential in order to establish an identical price for each unit demanded at center j .

Appropriateness of an Expected Value/Profit Model
as Opposed to a Chance Constrained Model

Overview

The deterministic linear programming problem can be represented as:

Maximize

$$Z = c'x \quad (3.38)$$

subject to:

$$Ax \leq b \quad (3.39)$$

$$x \geq 0 \quad (3.40)$$

Where A is an $m \times n$ matrix of constants and c , b are corresponding constant vectors. The objective in the above problem is to determine an optimal decision vector x , that does not violate the constraint set specified above. In contrast, if some or all of the elements in the set (A, b, c) are stochastic, then one faces a stochastic programming problem due to stochastic variations introduced into the problem through random variations in the elements. In most realistic problems, it is possible to determine some probability distribution that can explain each of these variations.

Generally, a chance constrained formulation can be represented as:

Maximize

$$Z = c'x \quad (3.41)$$

subject to:

$$P (Ax \leq b) \geq \alpha \quad (3.42)$$

$$x \geq 0 \quad (3.43)$$

Where some or all of the elements of A , b , and c are random variables. The vector α represents a set of constants that are probability measures indicating the degree to which the constraint violations are permissible. Charnes and Cooper [6] were the first to develop the theory behind transforming a chance constraint into its deterministic equivalent, resulting in a problem that is usually nonlinear.

In the expected value formulation, the objective is to maximize the expected value or profit derived over a desired period of time; Jucker and Carlson [36]. The constraint set is deterministic with quantifiable probabilistic terms introduced in the objective function. These terms, for instance, can be due to randomness in demands and/or prices.

Conclusions

As noted previously, demand is the only source of uncertainty introduced in this research. As such, with a chance constrained model, the only constraints that are associated with probabilities are those constraints that require that the sum of the supplies received at any demand center exceed the random demand realized. For customer satisfaction, the management would generally prefer to have such probabilities of realization be in the range of 0.90 to 0.95. In other words, 90 to 95 percent of the time the management is capable of satisfying the random demand realized in each of the demand centers.

The model for this research problem should also focus on maximizing

profits given by an objective function composed of revenues minus costs. Considering the revenues generated would require including the same random demands in the objective function. Such a model is not meaningful, and will not fit the characterization of any of the three chance constrained models (E-model, V-model, and P-model) described by Charnes and Cooper [6]. Furthermore, sufficient revenue generation for the management is implicit in setting the probability values for satisfying demands as high as 0.9 to 0.95 or even higher. The chance constrained model for a plant location problem with stochastic demand is only logical if the focus is on minimizing total costs as in static models; Gonzalez-Valenzuela [29]. Needless to say, such a model disregards the key dimension of price sensitivity.

It is assumed that the allocation to each of the demand centers will be made before the actual demands become known. In this connection, it is more meaningful to consider an expected value model as opposed to a chance constrained model for the research problem.

Model Formulation for Problem Type 8

The Model

When the demand is stochastic, the number of units (q_{ij}) demanded at center j when supplied by plant i is assumed to follow a probability distribution. For instance, a normal distribution with mean $\mu_{q_{ij}}$ and variance $\sigma_{q_{ij}}^2$ or a uniform distribution is a good representation of such variability in demand; Gonzalez-Valenzuela [29], Jucker and Carlson [36]. All other notations used in this model have the same illustration previously indicated in this chapter.

The revenue received at a given demand center j if the product is shipped from plant i is

$$= \begin{cases} q_{ij} p_{ij} & \text{if } q_{ij} < x_{ij} \\ x_{ij} p_{ij} & \text{if } q_{ij} \geq x_{ij} \end{cases} \quad (3.44)$$

The cost incurred in shipping x_{ij} units of product from plant i to the demand center $j = r_{ij} x_{ij}$. Additionally, the fixed cost incurred in setting up the plants for a given location vector $y = (y_i)$ is

$$= \sum_{i=1}^m f_i y_i \quad (3.45)$$

Thus, the expected profit realized at the demand center j is

$$= \sum_{i=1}^m z_{ij} \left\{ E(q_{ij} p_{ij} \mid q_{ij} < x_{ij}) \cdot P(q_{ij} < x_{ij}) + x_{ij} p_{ij} \cdot P(q_{ij} \geq x_{ij}) - r_{ij} x_{ij} \right\} - \sum_{i=1}^m f_i y_i \quad (3.46)$$

The total expected profit realized in all of the demand centers is

$$= \sum_{j=1}^n \left[\sum_{i=1}^m z_{ij} \left\{ E(q_{ij} p_{ij} \mid q_{ij} < x_{ij}) \cdot P(q_{ij} < x_{ij}) + x_{ij} p_{ij} \cdot P(q_{ij} \geq x_{ij}) - r_{ij} x_{ij} \right\} \right] - \sum_{i=1}^m f_i y_i \quad (3.47)$$

For an expected value formulation, the objective is to maximize the expected total profit. Hence, the complete model for problem type 8 is

Maximize

$$Z = \sum_{j=1}^n \left[\sum_{i=1}^m z_{ij} \left\{ E(q_{ij} p_{ij} \mid q_{ij} < x_{ij}) \cdot P(q_{ij} < x_{ij}) + x_{ij} p_{ij} \cdot P(q_{ij} \geq x_{ij}) - r_{ij} x_{ij} \right\} \right] - \sum_{i=1}^m f_i y_i \quad (3.48)$$

subject to:

$$\sum_{i=1}^m z_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.49)$$

$$\sum_{j=1}^n x_{ij} < A_i y_i \quad ; \quad i = 1, \dots, m \quad (3.50)$$

$$z_{ij}, y_i = 0 \text{ or } 1 \quad (3.51)$$

$$x_{ij} \geq 0 \quad (3.52)$$

In the above model, a set of allocation constraints (3.49) require that each demand center be supplied by exactly one facility. The constraints set (3.50) ensures that the supplies made from each "open" plant do not exceed its capacity and that no supplies are made from closed plants. The expected profit contribution made to total expected profit by supplies received at demand center j by the product shipped from plant i ¹ is

¹In the subsequent material this will be termed the profit contribution made by cell (i, j) .

$$L(x_{ij}) = E(q_{ij} p_{ij} \mid q_{ij} < x_{ij}) \cdot P(q_{ij} < x_{ij}) \\ + x_{ij} p_{ij} \cdot P(q_{ij} \geq x_{ij}) - r_{ij} x_{ij} \quad (3.53)$$

Let $f(q_{ij})$ be the density function of the random demand q_{ij} , given that x_{ij} units are shipped from plant i to demand center j . As noted previously, both normal and uniform distributions have been found to be good representations of variability in demand. Also let ub_{ij} and lb_{ij} be the upper and lower limits of the range of values taken by the random demand q_{ij} .

Introducing these parameters, the expected profit contribution due to cell (i, j) can be expanded as

$$L(x_{ij}) = p_{ij} \mu_{q_{ij}} - p_{ij} \int_{x_{ij}}^{ub_{ij}} (q_{ij} - x_{ij}) f(q_{ij}) dq_{ij} \\ - r_{ij} x_{ij} \quad (3.54)$$

Thus, the expected profit model or the deterministic programming model for problem type 8 is

Maximize

$$Z = \sum_{j=1}^n \left[\sum_{i=1}^m z_{ij} \left\{ p_{ij} \mu_{q_{ij}} - p_{ij} \int_{x_{ij}}^{ub_{ij}} (q_{ij} - x_{ij}) f(q_{ij}) dq_{ij} \right. \right. \\ \left. \left. - r_{ij} x_{ij} \right\} \right] - \sum_{i=1}^m f_i y_i \quad (3.55)$$

subject to:

$$\sum_{i=1}^m z_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.56)$$

$$\sum_{j=1}^n x_{ij} \leq A_i y_i \quad ; \quad i = 1, \dots, m \quad (3.57)$$

$$z_{ij}, y_i = 0 \text{ or } 1 \quad (3.58)$$

$$x_{ij} \geq 0 \quad (3.59)$$

Proof for Concavity of $L(x_{ij})$

For simplicity if the subscripts ij are disregarded in equation (3.54), the expected profit contribution made by cell (i, j) can be represented as

$$L(x) = p\mu - rx - \int_x^{ub} p(q - x) f(q) dq \quad (3.60)$$

Also, for a differentiable function $f(x, t)$, the following holds true by Leibnitz's rule; Hildebrand [33].

$$\frac{d}{dx} \int_A^B f(x, t) dt = \int_A^B \frac{\partial}{\partial x} f(x, t) dt + f(x, B) \frac{dB}{dx} - f(x, A) \frac{dA}{dx} \quad (3.61)$$

The function $L(x)$ is continuous and twice differentiable.

Thus

$$\frac{\partial L}{\partial x} = -r + p \int_x^{ub} f(q) dq \quad (3.62)$$

The necessary and sufficient conditions for a stationary point to be maximum (concave) are:

$$\frac{\partial L}{\partial x} = 0 \quad \rightarrow \quad \int_{x^*}^{ub} f(q) dq = \frac{r}{p}, \text{ and} \quad (3.63)$$

$$\frac{\partial^2 L}{\partial x^2} < 0 \quad \rightarrow \quad -p f(x) < 0 ; \text{ which is true} \quad (3.64)$$

where x^* is the optimal supply quantity. This establishes the proof.

Example Problem

As a simple example, a problem with two plants ($m = 2$) and two demand centers ($n = 2$) is considered. The possible location vectors $y = (y_i)$; $i = 1, 2$ are $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. The infeasible vector $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is disregarded. For the location vector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, the four possible allocation matrices $z = (z_{ij})$; $i = 1, 2$ are

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each of these allocation matrices will result in a subproblem which is really the maximization of the sum of concave functions subject to a convex set of constraints. This is true because for a given location vector $\sum_{i=1}^m f_i y_i = \text{constant}$. Let l be the number of plants that are "open" in a given location vector. A close look reveals the fact that for each location vector there are l^n number of subproblems.

For the example problem, the number of subproblems contributed by each location vector are as follows:

$$y = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow 1^2 = 1 \text{ subproblem}$$

$$y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow 1^2 = 1 \text{ subproblem}$$

$$y = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow 2^2 = 4 \text{ subproblems}$$

Total of 6 subproblems

General Case

For a general case with m plants and n demand centers, the number of subproblems are given by

$${}^m C_m \cdot (m)^n + {}^m C_{m-1} \cdot (m-1)^n + {}^m C_{m-2} \cdot (m-2)^n + \dots + {}^m C_1 \cdot (1)^n$$

For a simple problem with $m = 5$ and $n = 10$, there are

$$\begin{aligned} & {}^5 C_5 (5)^{10} + {}^5 C_4 (4)^{10} + {}^5 C_3 (3)^{10} + {}^5 C_2 (2)^{10} + {}^5 C_1 (1)^{10} \\ & = 15,609,240 \text{ subproblems} \end{aligned}$$

Each of these subproblems, however, will be aimed at maximizing the sum of concave functions subject to a convex set of constraints.

As in the one dimensional case, it can be proven that the sum of concave functions is also concave for a multidimensional situation;

Bazaraa and Jarvis [2]. Consequently, each of these subproblems will be comprised of a concave objective function subject to a convex set of constraints. However, feasibility may not be assured for all of the subproblems developed. As such, given the assurance of feasibility for a subproblem, optimality is guaranteed if a stationary point can be obtained.

Optimality Conditions for Subproblems

In general, each of these subproblems can be expressed as
Maximize

$$\begin{aligned} Z = f(X_{ij}) & ; \quad i = 1, \dots, m \\ & \quad j = 1, \dots, n \end{aligned} \quad (3.65)$$

subject to:

$$g_i(X_{ij}) \leq A_i \quad ; \quad i = 1, \dots, m \quad (3.66)$$

$$X_{ij} \geq 0 \quad (3.67)$$

where f is a continuous function in the X_{ij} 's and the g_i 's are of the form

$$\sum_j X_{ij} \leq A_i \quad (3.68)$$

Elmaghraby [18] established optimality conditions for a stochastic transportation problem with continuous demand distributions. The objective was to minimize the expected total cost of production. Although the objective of this research problem is to maximize the expected total profit, the same conditions can be usefully modified to

give the optimality conditions for each of the subproblems if it exists. This is true since both the stochastic transportation problem and each of the subproblems have the same structure despite their contrasting objectives. As such, the optimality conditions are stated below without proof.

For a feasible solution X^0 to be optimal, it is necessary that

a. if
$$\sum_{j=1}^n x_{ij}^0 < a_i$$

then
$$\int_{x_{ij}^0}^{ub_{ij}} f(q_{ij}) dq_{ij} = \frac{r_{ij}}{p_{ij}} \quad (3.69)$$

for all $x_{ij}^0 > 0$.

b. if
$$\sum_{j=1}^n x_{ij}^0 = a_i$$

then
$$\int_{x_{ij}^0}^{ub_{ij}} f(q_{ij}) dq_{ij} = (\lambda_i + r_{ij}) / p_{ij} \quad (3.70)$$

for all $x_{ij}^0 > 0$

where $\lambda_i \geq 0$ is fixed for a given i .

The above two conditions, together with the following two are sufficient for optimality.

c. if
$$\sum_{j=1}^n x_{ij}^0 < a_i$$

then
$$\int_{x_{ij}^0}^{ub_{ij}} f(q_{ij}) dq_{ij} < \frac{r_{ij}}{p_{ij}} \quad (3.71)$$

for all $x_{ij}^0 = 0$.

d. if
$$\sum_{j=1}^n x_{ij}^0 = a_i$$

then
$$\int_{x_{ij}^0}^{ub_{ij}} f(q_{ij}) dq_{ij} \leq (\lambda_i + r_{ij}) / p_{ij} \quad (3.72)$$

for all $x_{ij}^0 = 0$.

For each of the subproblems, an optimal solution, if it exists, will satisfy the above mentioned conditions.

Difficulties in Obtaining the Optimal Solution

It is now evident that problem type 8 has a unique optimal solution and is given by the maximum of the optimal objective function values determined for all of the subproblems. Elmaghraby [18] documented an iterative solution methodology for determining the optimal solution of a stochastic transportation problem. Although the methodology has sound mathematical backing, the iterative steps involved are

so tedious that it is practically unsuited for solving even a problem of reasonable size.

As noted before, the number of subproblems even for a simple problem with $m = 5$ and $n = 10$ are exceedingly high (15,609,240). In contrast, the number of subproblems that needed to have been considered for problem type 6 investigated by Gonzalez-Valenzuela [29] were only 31. Even then, the iterative methodology was not employed due to the complexities involved. Therefore, model simplifications are necessary in order to establish a solution technique with a reasonable degree of difficulty.

Model Simplifications

The linear approximation technique proposed by Wilson [63] for a stochastic transportation problem was used to solve problem type 6. As a result, for a given location vector, the problem reduces to solving a deterministic transportation problem which is essentially a linear programming problem. Stated differently, the approximated problem of problem type 6 had the features of problem type 2 which had already been investigated by Ellwein [16].

However, such a linear approximation cannot be employed on problem type 8. The reason is, for a given location vector, the problem does not reduce to a stochastic transportation problem due to having included the dimension of price sensitivity of demand. The implications of price sensitivity are reflected in the model by the binary variables z_{ij} . A close look at the objective function for problem type 8 would reveal the fact that an attempt to linearize its non-linear terms will introduce terms composed of the product of two variables, such as

$z_{ij} \cdot x_{ij}$. Such an approximated model will be considerably difficult to solve, if not impossible.

For the reasons stated above, an approximation is sought that can lead to a revised model which will be logical and easy to solve. The profit contribution $L(x_{ij})$ due to cell (i, j) is given by equation (3.54). Assuming that x_{ij} is restricted only by $x_{ij} \geq 0$, the optimal supply quantities x_{ij}^* are determined by

$$\int_{x_{ij}^*}^{ub_{ij}} f(q_{ij}) dq_{ij} = \frac{r_{ij}}{p_{ij}} \quad (3.73)$$

For each of these optimal supply quantities, the optimal profit contributions due to the cell (i, j) can be deduced from

$$L(x_{ij}^*) = p_{ij} \mu_{q_{ij}} - p_{ij} \int_{x_{ij}^*}^{ub_{ij}} (q_{ij} - x_{ij}^*) f(q_{ij}) dq_{ij} - r_{ij} x_{ij}^* \quad (3.74)$$

Thus, the revised model for problem type 8 will focus on allocating the optimal supply quantities for each of the demand centers within capacity limitations. Yet which plant a given demand center is allocated to will be determined by the binary variables z_{ij} introduced in the revised model.

Revised Model

Notationally, the revised model for problem type 8 can be documented as

Maximize

$$Z = \sum_{j=1}^n \sum_{i=1}^m z_{ij} L(x_{ij}^*) - \sum_{i=1}^m f_i y_i \quad (3.75)$$

subject to:

$$\sum_{i=1}^m z_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.76)$$

$$\sum_{j=1}^n x_{ij}^* z_{ij} \leq A_i y_i \quad ; \quad i = 1, \dots, m \quad (3.77)$$

$$z_{ij}, y_i = 0 \text{ or } 1 \quad (3.78)$$

The above model is a binary linear programming model. The binary variables are comprised of both the location vector $y = (y_i)$ and the allocation matrix $z = (z_{ij})$. It should, however, be emphasized that the optimal solution determined by the revised model is only optimal for the approximated version of the original model for problem type 8.

Model for Problem Type 7

The model for problem type 7 can be obtained by relaxing the constraints on capacity limitations imposed on the model developed for problem type 8.² If M is a very large positive value representative of infinite capacities of the plants, the model can be documented as

Maximize

²See page 37.

$$Z = \sum_{j=1}^n \left[\sum_{i=1}^m z_{ij} \left\{ p_{ij} \mu_{q_{ij}} - p_{ij} \int_{x_{ij}}^{ub_{ij}} (q_{ij} - x_{ij}) f(q_{ij}) dq_{ij} - r_{ij} x_{ij} \right\} \right] - \sum_{i=1}^m f_i y_i \quad (3.79)$$

subject to:

$$\sum_{i=1}^m z_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.80)$$

$$\sum_{j=1}^n x_{ij} \leq M y_i \quad ; \quad i = 1, \dots, m \quad (3.81)$$

$$z_{ij}, y_i = 0 \text{ or } 1 \quad (3.82)$$

$$x_{ij} \geq 0 \quad (3.83)$$

Following the same arguments detailed before, the revised model for problem type 7 can be obtained as

Maximize

$$Z = \sum_{j=1}^n \sum_{i=1}^m z_{ij} L(x_{ij}^*) - \sum_{i=1}^m f_i y_i \quad (3.84)$$

subject to:

$$\sum_{i=1}^m z_{ij} = 1 \quad ; \quad j = 1, \dots, n \quad (3.85)$$

$$\sum_{j=1}^n x_{ij}^* z_{ij} \leq M y_i \quad ; \quad i = 1, \dots, m \quad (3.86)$$

$$z_{ij}, y_i = 0 \text{ or } 1$$

For an "open" plant the constraint (3.86) will not be binding at any time. As such, the optimality conditions given by a^3 will hold for any optimal location and allocation determined by the revised model stated above. Therefore, the optimal solution determined by the revised model for problem type 7 is also the true optimal for the original model documented before.⁴ This is in contrast to problem type 8 where the revised model is only guaranteed to provide the optimal solution for an approximated version of the original model.

Potential Applications of the Model

The investigation of this research is limited to a single product-single period problem. The model, however, does not take into account any salvage value at the end of the period. The production and distribution of perishable agricultural commodities do exhibit the features represented by this model. Products such as butter, cheese, meat, and eggs are good examples to name a few. The demand realized for these items are both price sensitive and stochastic, yet being perishable items, they do not have any salvage value.

³See page 42.

⁴See page 46.

CHAPTER IV

SOLUTION ALGORITHMS FOR THE REVISED MODEL

Introduction

The revised model for problem type 8 is a pure binary linear programming model. With m plants and n demand centers, there are $(m + n)$ constraints and $(m \cdot n + m)$ $(0, 1)$ variables. Due to having included the dimension of price sensitivity into the research problem, the number of binary variables has been increased by $m \cdot n$ as compared to the price insensitive demand problem described by type 6. For instance, the number of binary variables even for a small problem with $m = 6$ and $n = 8$ are 54. As such, the computation time required by an optimizing algorithm seeking the exact optimum can be exorbitant. In contrast, a good heuristic algorithm exploiting the structural properties of the revised model can provide near optimal or good solutions with much less computation time than those algorithms seeking the exact optimum.

The solution space for the revised model can be assumed to possess a finite number of possible feasible points. The purpose of employing an explicit enumeration algorithm for solving a binary linear model is to exhaustively enumerate all such points. The optimal solution in this case is determined by the point that yields the best maximum value of the objective function. With this technique the number of solution points may become impractically large, resulting in a situation where

the solution cannot be determined in a reasonable amount of time. As opposed to this, the implicit or partial enumeration considers only a portion of all possible solution points while automatically discarding the remaining ones as nonpromising; Taha [56].

Even an explicit enumeration can be made attractive if a good or near optimal solution determined by a heuristic algorithm can be incorporated as a lower bound (Z) in the enumeration procedure. A lower bound is considered because the revised model focuses on maximizing the objective function value. The approach of heuristically aiding an enumeration technique enables discarding those solution points having a functional value lower than the lower bound incorporated. As a result, considerable savings in computational time can be obtained. The consequences of heuristically aiding an implicit enumeration technique can be more rewarding than aiding an explicit enumeration technique for the following reasons. First, only a portion of all possible solution points is considered. Second, within this subset of points, those having an objective function value lower than the lower bound are also discarded.

The remaining material presented in this chapter describes the development of solution algorithms in the following order. A heuristic algorithm that exploits the structural properties of the revised model is described. Then the applicability of the extreme point ranking technique for the revised model is illustrated. The applicability of the branch-and-bound technique for the revised model is also illustrated. Finally, the chapter is concluded with a summary of all the techniques explored.

Applicability of a Heuristic Algorithm for the Revised Model

Overview

The use of "add" heuristics for determining the near optimal or good solutions for problem type 2 is described by Kuehn and Hamburger [41].¹ Feldman, Lehrer and Ray [22] later continued the development by employing "drop" heuristics. The "add" heuristics, which begins with the best single plant, relies on sequential addition of warehouses one at a time. The "drop" heuristics, on the other hand, starts with all plants in use and drops out plants one at a time. Both these heuristics were employed for a cost minimization objective function which is usually the concern of the management with price insensitive demand. The results reported in [22] show that the "drop" approach gave solutions that were significantly lower in total cost than those given by the "add" approach.

Another advantage in using the "drop" approach with capacitated plants is that it begins with a feasible solution and attains infeasibility as quickly as possible when plants are dropped out. This is advantageous because those plants losing the potential for future drop can be identified with greater relative ease than with "add" heuristics. For the above mentioned reasons, a heuristic algorithm that uses the basic ideas of "drop" heuristics is presented below.

¹See page 14.

Notations

The following notations are added to those introduced before in order to illustrate the structure of the heuristic algorithm. This, however, is not a comprehensive list to perform the actual computations. A comprehensive list of notations introduced are given in the computer programs presented in the Appendix.

TPORG = The objective function value for the original location vector

TPROF² = The objective function value for the location vector presently under consideration in this cycle

TPFIN = The maximum objective function value of all the location vectors so far considered in this cycle

NCON = Number of plants that have no potential for future drop

ADROP_i = The remaining capacity of plant *i* after allocating a subset of the demand centers

IDROP_i = $\begin{cases} 0 & \text{Plant } i \text{ has no potential for drop} \\ 1 & \text{Plant } i \text{ has potential for drop} \end{cases}$

NORG = $\begin{cases} 0 & \text{A feasible solution has not been found for the problem} \\ 1 & \text{A feasible solution has been found for the problem} \end{cases}$

$$S = \{i \mid y_i = 1\}$$

$$S'_j = \{i \mid i \in S \text{ and } A_i \geq x_{ij}^*\}$$

²A location vector presently under consideration is obtained by dropping any "one" plant that has the potential for drop from the original location vector.

$$S_j'' = \{i \mid i \in S \text{ and } ADROP_i \geq x_{ij}^*\}$$

Priority Rules

The following priority rules are considered:

Rule 1:

Rank the demand centers (j's) in a descending order of profit contributions ($L(x_{ij}^*)$) made by each cell (i, j). The demand center that contributes to the maximum profit will be given the highest priority of allocation.

Rule 2:

Rank the plants (i's) in a descending order of fixed costs (f_i) as potential plants for drop. The plant that contributes to the largest fixed cost will be given the highest priority for drop.

Algorithmic Steps

The steps for the heuristic algorithm can be documented as follows.

Step 1:

Initialize by setting TPORG, TPROF, TPFIN, and NCON to zero. Also set

$$y_i = 1, \text{ and} \\ IDROP_i = 1 ; i = 1, \dots, m \quad (4.1)$$

Step 2:

Perform the global test for feasible capacity. This test is performed by considering the following inequality.

$$\sum_{i \in S} A_i > \sum_{j=1}^n \min_{i \in S'_j} \{x_{ij}^*\} \quad (4.2)$$

Go to step 3 if the inequality holds. If not, check whether an original feasible solution has been found, i.e., NORG = 1. If it has not been found, go to step 15. Otherwise go to step 10.

Step 3:

Perform the global test for optimal profits. This test is performed by considering the following inequality.

$$\sum_{j=1}^n \max_{i \in S'_j} \{L(x_{ij}^*)\} - \sum_{i \in S} f_i \geq \text{TPROF} \quad (4.3)$$

If the inequality holds, go to step 4. If not, check whether an original feasible solution has been found, i.e., NORG = 1. If it has not been found go to step 13. Otherwise go to step 9.

Step 4:

Allocate the first/next available demand center. Priority rule 1 is observed in this step. Check whether all demand centers have been allocated. If so, go to step 7. Otherwise go to step 5.

Step 5:

Perform the partial test for feasible capacity. This test is performed by considering the following inequality.

$$\sum_{i \in S} \text{ADROP}_i > \sum_{j=1}^n \min_{i \in S''_j \text{ and } z_{ij} \neq 1} \{x_{ij}^*\} \quad (4.4)$$

Go to step 6 if the inequality holds. If not, check whether an original feasible solution has been found, i.e., $NORG = 1$. If it has not been found, go to step 15. Otherwise go to step 10.

Step 6:

Perform the partial test for optimal profits. This test is performed by considering the following inequality.

$$\sum_{j=1}^n L(x_{ij}^*) \text{ for all } i \in S \text{ and } z_{ij} = 1$$

$$+ \sum_{j=1}^n \max_{i \in S_j'' \text{ and } z_{ij} \neq 1} \{L(x_{ij}^*)\} - \sum_{i \in S} f_i \geq TPROF \quad (4.5)$$

Go to step 4 if the inequality holds. If not, check whether an original feasible solution has been found; i.e., $NORG = 1$. If it has not been found, go to step 13. Otherwise go to step 9.

Step 7:

Test whether $TPROF > TPORG$. If the inequality holds, go to step 8. Otherwise go to step 9.

Step 8:

Test whether $TPROF > TPFIN$. If the inequality holds, preserve the current location-allocation pattern as the final location-allocation pattern and go to step 11. Otherwise go directly to step 11.

Step 9:

Test whether the reduction in profits given by $(TPORG - TPROF)$ is greater than the fixed cost of the plant currently dropped. If so, go to step 10. Otherwise go to step 11.

Step 10:

Set $IDROP_i = 0$ for the plant currently dropped. Increment $NCON$ by 1, and test whether $NCON \geq m$. If so, go to step 14. Otherwise go to step 11.

Step 11:

Test whether all plants have been dropped once in this cycle. If so, go to step 12. Otherwise go to step 13.

Step 12:

Set the original location-allocation pattern equal to the final location-allocation pattern currently stored. Also set $IDROP_i = 0$ for the plant dropped in the final pattern. Increment $NCON$ by 1, and test whether $NCON \geq m$. If so, go to step 14. Otherwise go to step 13.

Step 13:

Drop the first/next available plant. Priority rule 2 is observed in this step. Go to step 2.

Step 14:

Stop. Final location-allocation pattern is obtained.

Step 15:

Stop. No feasible solution exists for the heuristics employed due to capacity limitations.

The flow chart given in Figure 4.1 clearly explains the sequence of steps undertaken by the heuristic algorithm to establish the decisions on location and allocation for a given problem.

Weakness

At step 5 a partial test for feasible capacity is performed. If the test fails prior to finding an original feasible solution,

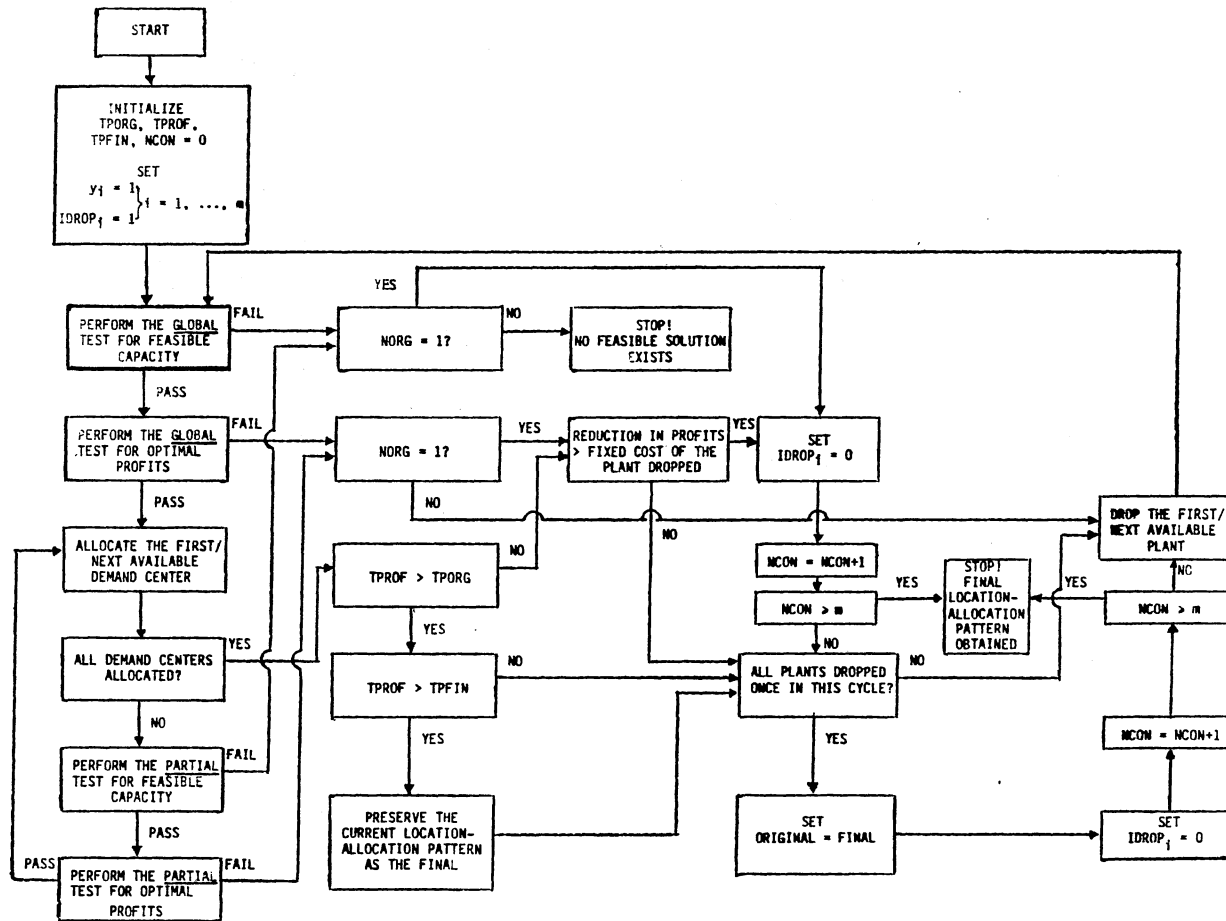


Figure 4.1. Flow Chart for Heuristic Algorithm

the algorithm is terminated. It should be emphasized, however, that it is only true when the priority rules are observed while performing the algorithmic steps. Yet it may be possible to find a feasible solution by prioritizing the demand centers in an ascending order of optimal supply quantities. Thereby the demand center that requires the least optimal supply quantity will be given the highest priority of allocation. However, it cannot be determined how close such a solution is to the optimal. The reason is that an improvement on this solution cannot be obtained due to not being able to perform the remaining algorithmic steps repetitively. It is possible that for medium³ and large⁴ size problems even a heuristically aided optimal enumeration technique fails to find the optimal solution within a reasonable length of computation time. In such cases, it is inevitable for the management to accept the heuristic solution as a near optimal or a good solution. Thus, it is not worth trading off the benefits derived in finding just a feasible solution to the loss in profits which is presumably not known. This seems to be the only weakness in the heuristic algorithm presented above.

Applicability of the Extreme Point Ranking Technique for the Revised Model

Theorem

It has been proved, Taha [56], that a binary linear problem has the unique property that the optimal solution must occur at an extreme point

³A medium size problem usually ranges from $m = 10$ and $n = 15$ to $m = 20$ and $n = 25$.

⁴A large size problem usually has number of plants and demand centers equal to and above $m = 30$ and $n = 40$.

of the convex polyhedron representing the continuous solution space. As such, the theorem is stated below without proof.

In a 0-1 binary linear problem if the integer constraint is replaced by the continuous range $0 \leq x_j \leq 1$; where $x_j = (0, 1)$, then

1. The resulting continuous solution space contains no feasible points⁵ in its interior, and
2. The optimum feasible solution of the integer problem occurs at an extreme point of the continuous convex space.

Methodology

The basic principles illustrated by the above theorem can be usefully applied to solve for the optimal solution of the revised model. First, the model can be solved for the continuous optimum by relaxing the integer requirement on the binary variables. The integer requirement when relaxed produces constraints on both the location vector $y = (y_j)$ and the allocation matrix $z = (z_{ij})$. All of these constraints are bounded between 0 and 1. As such, a primal simplex method for bounded variables, Taha [56], is used to perform the computations efficiently to determine the continuous optimum. The lower bound (Z) on the optimal integer solution for the revised model can be determined by employing the heuristic algorithm. Therefore, the optimal integer solution cannot be any lower than this lower bound or any higher than the continuous optimum.

Assume that the continuous optimum is not all integer. The adjacent extreme points to the continuous optimum can be determined by introducing nonbasic variables, one at a time, into the basis. The

⁵Implies no points satisfying the integer conditions.

next ranked extreme point is identified as the one yielding the largest objective function value amongst these adjacent points. If any of these extreme points are binary feasible and have a functional value greater than the lower bound Z , then the lower bound should be updated by the newly found integer solution. Additionally, only those adjacent points having an objective function value less than or equal to the ranked extreme point presently under consideration needs to be stored for further computations. Should both values match, a test has to be performed by comparing the bases along with the associated nonbasic variables at zero and one level, to avoid redundancy. Moreover, the extreme point may be adjacent to more than one of the ranked extreme points. Therefore, the same test should be performed on all of the promising adjacent points generated by comparing the bases with the stored points that have potential for being considered in the future.

Notations

The following notations are included with those introduced before in order to describe the mechanism of the extreme point ranking technique. This is not a comprehensive list to perform the actual computations. A comprehensive list of notations introduced is given in the computer programs presented in the Appendix.

E_1 = The first ranked extreme point corresponding to the continuous optimal solution.

Z_1 = The objective function value of the first ranked extreme point.

Similarly,

E_t = The t^{th} ranked extreme point.

Z_t = The objective function value of the t^{th} ranked extreme point.

Z = The objective function value of the adjacent point generated.

$$IIND_i = \begin{cases} 0 & \text{The } i^{\text{th}} \text{ adjacent extreme point has no potential for} \\ & \text{being considered in the future.} \\ 1 & \text{The } i^{\text{th}} \text{ adjacent extreme point has potential for being} \\ & \text{considered in the future.} \end{cases}$$

$$NOPER = \begin{cases} 0 & \text{An improved binary feasible solution has not been found} \\ & \text{in the current cycle.} \\ 1 & \text{An improved binary feasible solution has been found in} \\ & \text{the current cycle.} \end{cases}$$

Algorithmic Steps

Step 1:

Solve the equivalent linear program of the revised model by relaxing the binary requirements. If E_1 is binary feasible, stop. E_1 is optimum. Otherwise go to step 2.

Step 2:

This step is common to all of the subsequent ranked extreme points to be considered. As such, the illustration is generalized for the t^{th} ranked extreme point. At E_t , generate the adjacent point by introducing the first nonbasic variable into the basis. Discard if the inequality $Z < Z < Z_t$ does not hold. If it does, then test for redundancy with the points stored in the list. If the point is not redundant, go to step 3. Otherwise discard and repeat step 2 for the next nonbasic variable. If the variable considered is the last nonbasic variable, go to step 5.

Step 3:

Test for binary feasibility. If a point is binary feasible, test whether $Z > \bar{Z}$. If the inequality holds, update \bar{Z} , set $\text{NOPER} = 1$, and go to step 4. Otherwise go directly to step 4.

Step 4:

Store the adjacent extreme point generated in the list. Set $\text{IIND}_j = 1$ and go to step 2 if a nonbasic variable is still available. If none, go to step 5.

Step 5:

If $\text{NOPER} = 0$ in the last cycle go to step 6. Otherwise perform the test to identify whether any of the stored adjacent points can be discarded. This test is performed using the inequality $Z < \bar{Z}$, where \bar{Z} is the value of the newly found binary feasible solution. If the inequality holds, set $\text{IIND}_j = 0$ for those adjacent points and go to step 6.

Step 6:

Test whether there are any potential adjacent points for consideration, i.e., at least one point should have $\text{IIND}_j = 1$. If none go to step 8. Otherwise go to step 7.

Step 7:

Set $\text{NOPER} = 0$. Select the adjacent point that has the largest objective function value amongst those points that have $\text{IIND}_j = 1$. Set this point to be the next ranked extreme point and go to step 2.

Step 8:

Stop. Optimal binary solution is found.

Conclusions

The effectiveness of the extreme point ranking technique for solving the revised model is primarily dependent on the number of extreme points between the continuous optimum and the lower bound \bar{Z} . If this number is relatively large, this technique can burden the computer memory. It is especially true for problems with several linear constraints.

Applicability of the Branch-and-Bound Technique for the Revised Model

The branch-and-bound technique is based on implicit enumeration. This technique essentially ranks the extreme points in a selective manner for a binary linear problem. As such, it is applicable for the revised model also. More specifically, those extreme points violating the binary condition and having a functional value between those of the current node and the new node are implicitly ranked and need not be considered explicitly; Taha [56]. As a consequence, this technique does not tax the computer memory. Due to this reason, the branch-and-bound technique is usually preferred over the extreme point ranking technique for solving binary linear problems with several constraints.

The IBM Mathematical Programming System Extended/370, (MPSX/370), Mixed Integer Programming/370 (MIP/370), [35], is a programming package that uses a branch-and-bound technique for solving mixed and pure integer problems. The user is required to supply the data for the problem to be solved as well as write a simple program in a language called Mathematical Programming System Control Language (MPSC). This package also facilitates the use of an objective parameter, `MXDROP`. This parameter

can be set equal to the functional value determined for the revised model using the heuristic algorithm. This value, being the lower bound on the optimal binary solution, can be used to avoid paths that lead to worse solutions and, consequently, can speed up the search.

Summary

A heuristic and two optimizing algorithms for solving the revised model are described in this chapter. The extreme point ranking technique based on generating all the adjacent extreme points is aided by the heuristic solution. Only then the nonpromising adjacent points can be discarded, resulting in some savings in computer storage. The branch-and-bound technique, on the other hand, may prove to be efficient on small problems even without being aided by the heuristic solution. This is because of the implicit enumeration scheme built into this technique. However, with large problems, aiding the branch-and-bound technique with a heuristic solution may bring about savings in computation time.

CHAPTER V

DATA GENERATION, COMPUTATIONAL EXPERIENCE, SENSITIVITY ANALYSIS AND MODEL VALIDATION

Introduction

The material documented in this chapter is divided into five sections. They are:

1. Data generation,
2. Computational experience,
3. Sensitivity analysis,
4. Model validation, and
5. Summary

Data Generation

The Need

There is no real world data of knowledge to the author that documents the importance of having to include the dimensions of price sensitivity and stochastic demand in a plant location-allocation problem. Also, there have not been any theoretical studies where data could be readily obtained. As such, the revised model can only be tested and validated using the randomly generated test data.

Data Required

The main items of data necessary for the revised model are evident from model parameter definitions given in Chapter III. They are, however, restated below for convenience.

1. The fixed cost (f_i) of plant i .
2. The capacity (A_i) of plant i .
3. The total per unit variable cost (r_{ij}) of the product produced at plant i and shipped to demand center j .
4. The per unit price (p_{ij}) of the product received at demand center j when supplied by plant i .
5. The mean of the demand distribution ($\mu_{q_{ij}}$) descriptive of the number of units demanded at center j when supplied by plant i .
6. For a two parameter distribution such as a normal distribution, the standard deviation ($\sigma_{q_{ij}}$) descriptive of the number of units demanded at center j when supplied by plant i is also required.

Price Sensitivity and Demand

"Elasticity of Demand" is an important economic concept descriptive of a drop in demand with an increase in price. This concept, however, can easily be applied to deterministic demands. If the demand is stochastic, the question as to what exactly is the effect of price sensitivity in the random demand arises. For the purpose of this research, it is assumed that the influence of price sensitivity is only reflected in the mean of the distribution of demand. As such, a hyperbolic relationship between the mean of the demand distribution and the per unit price of the product described by the following equation is considered.

$$p_{ij} \cdot \mu_{q_{ij}} = k \quad (5.1)$$

Where k is a constant > 0 . It should be noted that the above relationship is limited to the first quadrant only.

Test Data

In order to test and validate the revised model, a series of test problems has to be solved using the algorithmic procedures delineated in Chapter IV. All test problems require randomly generated input data for the parameters listed above. It is reported in [58] that the use of data randomly generated from uniform distributions is advantageous for the following reasons.

1. The use of uniform distributions results in a high degree of variability in the test data.
2. The bias towards extremes in the number of facilities required for feasibility is eliminated due to the use of uniform distributions for generating capacities and demands.
3. The data generated for p_{ij} and r_{ij} being uniformly distributed avoids loss centers which could have been the case if too high a price and too low a cost were set for the product sold in a selected number of demand centers.

In conclusion, this method of generating test data enables solving problems that are representative of situations typically encountered in the real world. As such, the data on all of the above parameters were generated using the uniform distributions. The IBM subroutine RANDU was

used to generate a uniformly distributed random number between 0 and 1 and was then converted between the appropriate limits established for the parameters in a subroutine UNFRM written by the author.

Computational Experience

Organization

It has been noted that both normal and uniform distributions provide adequate description for the random demand realized in each of the demand centers. The normal, being a two parameter distribution, requires data on both $\mu_{q_{ij}}$ and $\sigma_{q_{ij}}$. In contrast, a uniform distribution is well described simply by specifying the interval where it is defined. It has been assumed that the price sensitivity influences only the mean of the distribution. As such, it is meaningful to perform test runs with varying means. Therefore, for the normal distribution of demand two cases are considered. First, only the mean of the distribution is varied and the standard deviation is kept constant. Second, both the mean and the standard deviation of the distributions are varied. For simplicity, the former is termed the type 1 normally distributed demands, while the latter the type 2 normally distributed demands. In the subsequent text these terms will be used appropriately. With this classification, there are three types of test runs required to be performed on a series of test problems. Two for normally distributed demands and one for uniformly distributed demands. Added to this is the classification of capacitated and uncapacitated plants. As a result, there are a total of six test runs that are necessitated. Test problems are again classified as small, medium, and large. For the purpose of this research, problem sizes ranging from

2 x 2 to 8 x 10 are classified as small size problems. While those ranging from 10 x 15 to 20 x 25 are classified as medium size problems. Finally, those of 30 x 40 and above are classified as large size problems. In the above classification, the first parameter is the number of plants (m) while the second the number of demand centers (n). All of these classifications are simply the author's judgment based upon the research undertaken in this area.

Data For Test Runs

The following data were used for the parameters of the revised model for problem type 8 with three distinct demand distributions. For problem type 7, which is uncapacitated, all of the data were the same with the exception of capacity of the plants. In this case, large numbers that are representative of infinite capacity were used as limits on capacity. The data chosen were based on the author's intuitive judgment.

Type 1 Normally Distributed Demand.

f_j is uniform on (200, 300)

A_j is uniform on (300, 350)

r_{ij} is uniform on (0.5, 2.5)

p_{ij} is uniform on (3, 6)

$\mu_{q_{ij}} = 600/p_{ij}$

$\sigma_{q_{ij}} = 10$

Type 2 Normally Distributed Demand.

f_j is uniform on (200, 300)

A_j is uniform on (300, 350)

r_{ij} is uniform on (0.5, 2.5)

p_{ij} is uniform on (3, 6)

$\mu_{q_{ij}} = 600/p_{ij}$

$\sigma_{q_{ij}}$ uniform on (7, 13)

Uniformly Distributed Demand.

f_i is uniform on (200, 300)

A_i is uniform on (400, 450)

r_{ij} is uniform on (0.5, 2.5)

q_{ij} is uniform on (a, b)

where

a is uniform on (50, 100)

b is uniform on (150, 300)

$p_{ij} = 600/\mu_{q_{ij}}$

Programming Concerns

Computer programs for the heuristic algorithm and the extreme point ranking technique were written in FORTRAN Language Level 77, and a VS FORTRAN compiler was used. As per the revised model of problem type 8, both the extreme point ranking technique and the branch-and-bound technique require values of the coefficients such as $L(x_{ij}^*)$, x_{ij}^* , f_i , and A_i . The MPSX/370, MIP/370 package program was used to implement the branch-and-bound technique. The program written for the extreme point ranking technique and the MPSX/370, MIP/370 package required significantly different data deck set ups. Therefore, the code written for the heuristic algorithm was suitably modified to produce two different versions that accomplished these tasks. This enabled setting up the data deck automatically for both optimizing algorithms by executing the

program written for the heuristic algorithm. Considerable labor was saved by not having to set up the data decks individually. Only the computer programs for the heuristic algorithm that included the code for setting up the data deck for the extreme point ranking technique are presented in the Appendix.

Programming Difficulties

Storage problems are inherent with the extreme point ranking technique. As will be noted later, even solving a problem of any more than 3×4 taxed the computer memory significantly. This difficulty was overcome by storing the values of the variables of each adjacent extreme point generated in direct access files contained in a temporary disc storage area. However, the functional value Z_j and the value of the variable $IIND_j$ indicating whether an adjacent extreme point has potential for being considered in the future were still stored in the main memory. The reason is that these two variables alone enabled making decisions in most instances, without having to retrieve the values of the remaining variables from the direct access files. This feature avoided unnecessary storage and retrieval of data. The program written for the extreme point ranking technique was modified accordingly to read and write data pertaining to adjacent extreme points in direct access files.

Notations

The notations used in the subsequent tables are explained below for clarity.

FVCON = Functional value of the continuous optimum.

FVHEUR = Functional value of the integer solution determined by the heuristic algorithm.

FV_{EXT} = Functional value of the optimal/best integer solution determined by the extreme point ranking technique.

FV_{BB} = Functional value of the optimal/best integer solution determined by the branch-and-bound technique.

CPU_{HEUR} = Central processor unit execution time in seconds for the heuristic algorithm.

CPU_{EXT} = Central processor unit execution time in seconds for the extreme point ranking technique.

CPU_{BB} = Central processor unit execution time in seconds for the branch-and-bound technique.

$$\text{Percentage deviation} = \frac{\text{Optimal/best value given by } FV_{EXT} \text{ or } FV_{BB} - FV_{HEUR}}{\text{Optimal/best value given by } FV_{EXT} \text{ or } FV_{BB}} \times 100$$

Computational Results for Problem Type 8

Tables 5.1 through 5.3 present the computational results for small size problems with capacitated plants and the three different types of demand distributions. With all of these problems, the heuristic algorithm required the lowest computation time followed by the branch-and-bound technique and then the extreme point ranking technique. For problems of 2 x 2 through 3 x 4, the computation times of both the extreme point ranking technique and the branch-and-bound technique were somewhat comparable. These were the only problems solved using the main memory storage with the extreme point ranking technique. For a 4 x 5 problem, the number of adjacent extreme points generated became too

TABLE 5.1

RESULTS FOR SMALL SIZE PROBLEMS WITH CAPACITATED
PLANTS AND TYPE 1 NORMALLY DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVEXT	FVBB	CPU _{HEUR} (Sec.)	CPU _{EXT} (Sec.)	CPU _{BB} (Sec.)	Percent- age deviation
2 x 2	599.46	557.80	557.80	557.80	2.60	4.63	3.90	0.00
2 x 3	1104.51	981.64	981.64	981.64	2.62	4.70	3.94	0.00
3 x 4	1490.52	1368.13	1368.13	1368.13	2.64	5.28	3.97	0.00
4 x 5	2082.94	1907.67	1950.98	1950.98	2.67	35.24	4.09	2.22
5 x 6	2418.34	2168.35	2228.89	2228.89	2.73	440.08	4.38	2.72
6 x 8	3227.69	2991.04	-	3054.59	2.82	-	8.14	2.08
8 x 10	3913.28	3648.72	-	3678.30	3.09	-	69.43	0.80
8 x 10	3913.28	3648.72	-	3678.30	3.09	-	69.32*	0.80

*Aided by the lower bound determined from the heuristic algorithm.

TABLE 5.2

RESULTS FOR SMALL SIZE PROBLEMS WITH CAPACITATED
PLANTS AND TYPE 2 NORMALLY DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVEXT	FVBB	CPUHEUR (Sec.)	CPUEXT (Sec.)	CPUBB (Sec.)	Percent- age deviation
2 x 2	763.27	663.43	663.43	663.43	2.59	4.63	3.85	0.00
2 x 3	1100.73	940.29	968.67	968.67	2.63	4.70	3.98	2.93
3 x 4	1519.79	1456.95	1461.43	1461.43	2.64	4.79	3.95	0.31
4 x 5	2008.76	1762.97	1815.89	1815.89	2.67	23.59	4.47	2.91
5 x 6	2394.05	2182.73	2184.61	2184.61	2.77	689.59	6.88	0.09
6 x 8	3250.11	3075.79	-	3075.79	2.82	-	11.30	0.00
8 x 10	4008.37	3739.46	-	3797.44	3.02	-	17.91	1.53
8 x 10	4008.37	3739.46	-	3797.44	3.02	-	17.40*	1.53

*Aided by the lower bound determined from the heuristic algorithm.

TABLE 5.3

RESULTS FOR SMALL SIZE PROBLEMS WITH CAPACITATED
PLANTS AND UNIFORMLY DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVEXT	FVBB	CPUHEUR (Sec.)	CPUEXT (Sec.)	CPUBB (Sec.)	Percent- age deviation
2 x 2	642.01	574.34	574.34	574.34	2.45	4.49	3.73	0.00
2 x 3	1031.80	887.80	887.80	887.80	2.48	4.59	3.80	0.00
3 x 4	1369.03	1280.65	1280.65	1280.65	2.50	4.78	3.84	0.00
4 x 5	1873.95	1660.22	1660.22	1660.22	2.54	52.76	4.19	0.00
5 x 6	2155.78	2001.45	2001.45	2001.45	2.57	169.53	4.33	0.00
6 x 8	2964.83	2741.34	-	2818.57	2.70	-	6.91	2.74
8 x 10	3744.39	3410.60	-	3461.80	2.96	-	29.30	1.48
8 x 10	3744.39	3410.60	-	3461.80	2.96	-	27.95*	1.48

*Aided by the lower bound determined from the heuristic algorithm.

large to have them stored in the main memory. This required storing them in direct access files contained in a temporary disc storage area. Even after resorting to this feature, the only problems that were solved were problem sizes 4×5 and 5×6 . Yet, even for a 5×6 problem with type 1 and type 2 normally distributed demands, the CPU times were exceedingly high. Two reasons can be attributed to this case. One, the use of direct access files for the purposes of reading and writing requires considerable time for the system to initially format the requested space in temporary discs. Two, the number of adjacent points generated between the continuous optimum and the lower bound determined by the heuristic algorithm are too many to determine an optimal binary solution within a reasonable computation time. Since the computation times required to solve the 5×6 problem were considerably high, the 6×8 and 8×10 problems were not attempted using the extreme point ranking technique.

The branch-and-bound technique, on the other hand, determined the optimal binary solution of all of the seven small problems attempted within a reasonable length of computation time. Because these problems were solved with relative ease, they were all attempted even without specifying a lower bound determined by the heuristic algorithm. As indicated in the tables, solving the 8×10 problem aided by the lower bound did bring about savings in computation time. However, they are not significant enough to comment on.

As for the functional values determined by the heuristic algorithm, they were only zero to three percent off from the optimal objective function values determined by either one of the optimizing algorithms. Particularly, for the case of uniformly distributed demands five out of

the seven problems attempted had the heuristic algorithm determine the optimal objective function value. This indicates that the heuristic has produced solutions that can be classified as near optimal solutions acceptable to the management. Also, the fact that the computation time is the lowest for the heuristic algorithm makes it preferable over the two optimizing algorithms.

Tables 5.4 through 5.6 document the computational results for medium and large size problems with capacitated plants and the three different demand distributions. The heuristic algorithm solved all of these problems within a reasonable computation time. Having found the extreme point ranking technique unsuited for some small problems, none of the medium or large size problems were attempted using this technique. Even with the branch-and-bound technique, the results obtained were not too attractive. For a 10 x 15 problem aided with the lower bound determined by the heuristic algorithm, only a best integer solution was found within the specified CPU time of 600 Seconds (10 minutes). Optimality was not proved with all three types of demand distributions. However, the best functional values obtained were well within three percent of those values determined by the heuristic algorithm. The 10 minutes limit on CPU execution time was perceived to be reasonable for solving a 10 x 15 problem. It seems the most logical and practical approach for solving medium and large size problems is to use the heuristic algorithm. Experience with small problems supports the possibility that near optimal solutions can be obtained for medium and large size problems.

TABLE 5.4
RESULTS FOR MEDIUM AND LARGE SIZE PROBLEMS WITH
CAPACITATED PLANTS AND TYPE 1 NORMALLY
DISTRIBUTED DEMANDS

Problem size (m x n)	FV _{CON}	FV _{HEUR}	FV _{BB}	CPU _{HEUR} (Sec.)	CPU _{BB} (Sec.)	Percentage deviation	Comments
10 x 15	6400.63	5783.66	5939.48	3.71	600.00	2.62	Optimality not proved
15 x 20	-	8043.87	-	5.65	-	-	-
20 x 25	-	9733.59	-	13.50	-	-	-
30 x 40	-	16085.99	-	80.66	-	-	-

TABLE 5.5
RESULTS FOR MEDIUM AND LARGE SIZE PROBLEMS WITH
CAPACITATED PLANTS AND TYPE 2 NORMALLY
DISTRIBUTED DEMANDS

Problem size (m x n)	FV _{CON}	FV _{HEUR}	FV _{BB}	CPU _{HEUR} (Sec.)	CPU _{BB} (Sec.)	Percentage deviation	Comments
10 x 15	6321.50	5738.11	5848.11	3.61	600.00	1.88	Optimality not proved
15 x 20	-	8010.54	-	6.32	-	-	-
20 x 25	-	9719.95	-	14.12	-	-	-
30 x 40	-	16158.57	-	93.46	-	-	-

TABLE 5.6
 RESULTS FOR MEDIUM AND LARGE SIZE PROBLEMS WITH
 CAPACITATED PLANTS AND UNIFORMLY
 DISTRIBUTED DEMANDS

Problem size (m x n)	FV _{CON}	FV _{HEUR}	FV _{BB}	CPU _{HEUR} (Sec.)	CPU _{BB} (Sec.)	Percentage deviation	Comments
10 x 15	5858.89	5362.54	5422.66	3.29	600.00	1.11	Optimality not proved
15 x 20	-	7115.90	-	7.17	-	-	-
20 x 25	-	8948.01	-	16.93	-	-	-
30 x 40	-	14894.28	-	80.41	-	-	-

Computational Results for Problem Type 7

The computational results for small size problems with uncapacitated plants are presented in Tables 5.7 through 5.9. In all of the 21 problems attempted with the three different demand distributions, only one had functional value determined by the heuristic algorithm that deviated from the optimal objective function value. Yet it was well within three percent. This indicates that the heuristic has performed excellently on all of the problems with uncapacitated plants. Coupled to this is the fact that the solution determined by the revised model is also optimal to the original model formulated for problem type 7.

Compared to the capacitated problem, however, the performance of the extreme point ranking technique on uncapacitated problems declined as the problem size became larger. Even for a problem of 5 x 6 with a CPU execution time limit of 900 sec. (15 minutes), the optimal solution was not determined. The results were very encouraging with the branch-and-bound technique. All of the seven problems with the three types of demand distributions were solved within a reasonable length of computation time. As is the case with capacitated problems, the heuristically aided branch-and-bound technique did not produce appreciable savings in computation time. It is evident from the times noted for both approaches in Tables 5.7 through 5.9.

The computational results for medium and large size problems are reported in Tables 5.10 through 5.12. Only two out of the nine problems had functional values determined by the heuristic algorithm different from those determined by the branch-and-bound technique. But the differences were within one percent. All of these problems were aided by the lower bound determined from the heuristic algorithm.

TABLE 5.7

RESULTS FOR SMALL SIZE PROBLEMS WITH UNCAPACITATED
PLANTS AND TYPE 1 NORMALLY DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVEXT	FVBB	CPUHEUR (Sec.)	CPUEXT (Sec.)	CPUBB (Sec.)	Percentage deviation
2 x 2	805.06	557.80	557.80	557.80	2.61	4.64	3.87	0.00
2 x 3	1377.77	1162.48	1162.48	1162.48	2.64	4.70	3.97	0.00
3 x 4	1843.89	1426.16	1426.16	1426.16	2.68	7.37	4.05	0.00
4 x 5	2494.66	2080.34	2080.34	2080.34	2.70	51.23	4.20	0.00
5 x 6	2871.51	2411.51	-	2411.51	2.78	900.00	4.24	0.00
6 x 8	3936.35	3377.85	-	3377.85	2.87	-	4.58	0.00
8 x 10	4891.72	4276.92	-	4276.92	3.16	-	4.81	0.00
8 x 10	4891.72	4276.92	-	4276.92	3.16	-	4.78*	0.00

*Aided by the lower bound determined from the heuristic algorithm.

TABLE 5.8

RESULTS FOR SMALL SIZE PROBLEMS WITH UNCAPACITATED
PLANTS AND TYPE 2 NORMALLY DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVEXT	FVBB	CPUHEUR (Sec.)	CPUEXT (Sec.)	CPUBB (Sec.)	Percent- age deviation
2 x 2	919.21	663.43	663.43	663.43	2.58	4.62	3.92	0.00
2 x 3	1397.71	1201.79	1201.79	1201.79	2.64	5.55	3.97	0.00
3 x 4	1886.21	1490.55	1490.55	1490.55	2.67	5.70	4.00	0.00
4 x 5	2446.59	2024.25	2024.25	2024.25	2.75	129.01	4.08	0.00
5 x 6	2842.18	2483.58	-	2483.58	2.76	900.00	4.13	0.00
6 x 8	3941.15	3384.03	-	3384.03	2.89	-	4.58	0.00
8 x 10	4939.42	4302.53	-	4302.53	3.15	-	4.88	0.00
8 x 10	4939.42	4302.53	-	4302.53	3.15	-	4.86*	0.00

*Aided by the lower bound determined from the heuristic algorithm.

TABLE 5.9
RESULTS FOR SMALL SIZE PROBLEMS WITH UNCAPACITATED
PLANTS AND UNIFORMLY DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVEXT	FVBB	CPUHEUR (Sec.)	CPUEXT (Sec.)	CPUBB (Sec.)	Percentage deviation
2 x 2	802.02	574.34	574.34	574.34	2.44	4.49	3.79	0.00
2 x 3	1282.97	1073.66	1073.66	1073.66	2.47	4.55	3.88	0.00
3 x 4	1681.96	1280.65	1280.65	1280.65	2.55	7.48	3.85	0.00
4 x 5	2271.80	1805.88	1805.88	1805.88	2.59	166.56	4.08	0.00
5 x 6	2620.27	2134.26	-	2134.26	2.62	900.00	4.17	0.00
6 x 8	3700.00	3174.64	-	3174.64	2.69	-	4.21	0.00
8 x 10	4628.11	3741.19	-	3837.11	2.96	-	4.93	2.50
8 x 10	4628.11	3741.19	-	3837.11	2.96	-	4.92*	2.50

*Aided by the lower bound determined from the heuristic algorithm.

TABLE 5.10

RESULTS FOR MEDIUM AND LARGE SIZE PROBLEMS WITH
 UNCAPACITATED PLANTS AND TYPE 1 NORMALLY
 DISTRIBUTED DEMANDS

Problem size (m x n)	FV _{CON}	FV _{HEUR}	FV _{BB}	CPU _{HEUR} (Sec.)	CPU _{BB} (Sec.)	Percentage deviation	Comments
10 x 15	7659.29	6715.97	6715.97	3.66	9.76	0.00	Optimality proved
15 x 20	10383.82	9101.64	9194.06	6.61	74.90	1.00	Optimality proved
20 x 25	12746.41	11601.83	11601.83	14.89	161.53	0.00	Optimality proved
30 x 40	-	18883.10	-	89.12	-	-	-

TABLE 5.11

RESULTS FOR MEDIUM AND LARGE SIZE PROBLEMS WITH
 UNCAPACITATED PLANTS AND TYPE 2 NORMALLY
 DISTRIBUTED DEMANDS

Problem size (m x n)	FV _{CON}	FV _{HEUR}	FV _{BB}	CPU _{HEUR} (Sec.)	CPU _{BB} (Sec.)	Percentage deviation	Comments
10 x 15	7612.77	6695.34	6695.34	3.68	9.88	0.00	Optimality proved
15 x 20	10323.97	9083.13	9083.13	6.72	139.75	0.00	Optimality proved
20 x 25	12706.50	11395.02	11395.02	15.10	587.53	0.00	Optimality proved
30 x 40	-	18793.11	-	87.45	-	-	-

TABLE 5.12

RESULTS FOR MEDIUM AND LARGE SIZE PROBLEMS WITH
 UNCAPACITATED PLANTS AND UNIFORMLY
 DISTRIBUTED DEMANDS

Problem size (m x n)	FVCON	FVHEUR	FVBB	CPUHEUR (Sec.)	CPUBB (Sec.)	Percentage deviation	Comments
10 x 15	7140.76	6132.25	6132.25	3.53	10.48	0.00	Optimality proved
15 x 20	9762.19	8312.40	8312.40	6.62	99.31	0.00	Optimality proved
20 x 25	12007.86	10404.13	10506.10	14.39	600.00	0.97	Optimality not proved
30 x 40	-	17630.66	-	87.25	-	-	-

Interestingly enough, for the 20 x 25 problem with uniform distribution of demands only the best integer solution was found. The optimality, however, was not proved within the 600 seconds (10 minutes) limit specified on CPU time. The large size problem of 30 x 40 has more than twice as many binary variables as does the largest medium size (20 x 25) problem. Experiences derived from solving the medium size problems enabled foreseeing unpromising results with the branch-and-bound technique for the large size problem. Therefore, the large size problem was not attempted with the branch-and-bound technique.

Sensitivity Analysis

Design of Test Runs

The test runs have to be designed appropriately for each demand distribution in order to perform a sensitivity analysis. The purpose of doing this is to determine the changes in revised model responses measured by the optimal objective function values for different values of demands and/or prices. Four test problems, comprised of two small and two medium size problems, were chosen to accomplish this task. All of the test problems were designed to be solved by the heuristic algorithm.

Type 1 Normally Distributed Demand. For the type 1 normally distributed demand, the values of p_{ij} were varied between the ranges (3, 6) and (5, 10). And for each range of values of p_{ij} , the values of $\mu_{q_{ij}}$ were determined between two different ranges. $\sigma_{q_{ij}}$ was kept constant at 10. As a result, the following four test runs were designed.

Run 1

p_{ij} uniform on (3, 6)

$$\mu_{q_{ij}} = 450/p_{ij} \text{ or}$$

$$75 \leq \mu_{q_{ij}} \leq 150$$

$$\sigma_{q_{ij}} = 10$$

Run 2

p_{ij} uniform on (3, 6)

$$\mu_{q_{ij}} = 600/p_{ij} \text{ or}$$

$$100 \leq \mu_{q_{ij}} \leq 200$$

$$\sigma_{q_{ij}} = 10$$

Run 3

p_{ij} uniform on (5, 10)

$$\mu_{q_{ij}} = 750/p_{ij} \text{ or}$$

$$75 \leq \mu_{q_{ij}} \leq 150$$

$$\sigma_{q_{ij}} = 10$$

Run 4

p_{ij} uniform on (5, 10)

$$\mu_{q_{ij}} = 1000/p_{ij} \text{ or}$$

$$100 \leq \mu_{q_{ij}} \leq 200$$

$$\sigma_{q_{ij}} = 10$$

Type 2 Normally Distributed Demand. The same variations in p_{ij} and $\mu_{q_{ij}}$ as per type 1 were maintained, except that the value of $\sigma_{q_{ij}}$ was also varied between two different ranges. Consequently, the following four test runs were designed.

Run 1

p_{ij} uniform on (3, 6)

$$\mu_{q_{ij}} = 450/p_{ij}$$

$\sigma_{q_{ij}}$ uniform on (5, 11)

$$\text{i.e., } 75 \leq \mu_{q_{ij}} \leq 150$$

$$5 \leq \sigma_{q_{ij}} \leq 11$$

Run 2

p_{ij} uniform on (3, 6)

$$\mu_{q_{ij}} = 600/p_{ij}$$

$\sigma_{q_{ij}}$ uniform on (7, 13)

$$\text{i.e., } 100 \leq \mu_{q_{ij}} \leq 200$$

$$7 \leq \sigma_{q_{ij}} \leq 13$$

Run 3

p_{ij} uniform on (5, 10)

$$\mu_{q_{ij}} = 750/p_{ij}$$

$\sigma_{q_{ij}}$ uniform on (5, 11)

$$\text{i.e., } 75 \leq \mu_{q_{ij}} \leq 150$$

$$5 \leq \sigma_{q_{ij}} \leq 11$$

Run 4

p_{ij} uniform on (5, 10)

$$\mu_{q_{ij}} = 1000/p_{ij}$$

$\sigma_{q_{ij}}$ uniform on (7, 13)

$$\text{i.e., } 100 \leq \mu_{q_{ij}} \leq 200$$

$$7 \leq \sigma_{q_{ij}} \leq 13$$

Uniformly Distributed Demand. For the uniformly distributed demand, the values of q_{ij} were determined from a uniform distribution on (a, b), where a and b were determined from appropriate uniform distributions. Depending upon the variations in $\mu_{q_{ij}}$, p_{ij} was determined appropriately. As a result, the following four test runs were designed.

Run 1

q_{ij} uniform on (a, b)

where

a uniform on (35, 75)

b uniform on (115, 225)

$$p_{ij} = 450/\mu_{q_{ij}}$$

$$\text{i.e., } 75 \leq \mu_{q_{ij}} \leq 150$$

$$23.09 \leq \sigma_{q_{ij}} \leq 43.30$$

$$3 \leq p_{ij} \leq 6$$

Run 2

q_{ij} uniform on (a, b)

where

a uniform on (50, 100)

b uniform on (150, 300)

$$p_{ij} = 600/\mu_{q_{ij}}$$

$$\text{i.e., } 100 \leq \mu_{q_{ij}} \leq 200$$

$$28.87 \leq \sigma_{q_{ij}} \leq 57.74$$

$$3 \leq p_{ij} \leq 6$$

Run 3

q_{ij} uniform on (a, b)

where

a uniform on (35, 75)

b uniform on (115, 225)

$$p_{ij} = 750/\mu_{q_{ij}}$$

$$\text{i.e., } 75 \leq \mu_{q_{ij}} \leq 150$$

$$23.09 \leq \sigma_{q_{ij}} \leq 43.30$$

$$5 \leq p_{ij} \leq 10$$

Run 4

q_{ij} uniform on (a, b)

where

a uniform on (50, 100)

b uniform on (150, 300)

$$p_{ij} = 1000/\mu_{q_{ij}}$$

$$\text{i.e., } 100 \leq \mu_{q_{ij}} \leq 200$$

$$28.87 \leq \sigma_{q_{ij}} \leq 57.74$$

$$5 \leq p_{ij} \leq 10$$

Results of the Sensitivity AnalysisFor Problem Type 8

The results of the sensitivity analysis for capacitated problems are presented in Tables 5.13 through 5.15. For the type 1 normally distributed demand, the functional values increased with the increase in variations on the mean of the distributions, keeping the price variations constant.¹ So is the case when the variations in the mean of the distributions are held constant and the prices are increased.² Interestingly, the changes in model responses given by the objective function values seem more sensitive to price variations than variations in the mean of the demand distributions. This can be observed by comparing runs 2 and 3 in Table 5.13. In run 3, as compared to run 2, the price variations are increased, but the variations in mean of the demand distributions are

¹Compare run 1 with run 2 and run 3 with run 4 in Table 5.13.

²Compare run 1 with run 3 and run 2 with run 4 in Table 5.13.

TABLE 5.13
 RESULTS OF SENSITIVITY ANALYSIS FOR PROBLEMS
 WITH CAPACITATED PLANTS AND TYPE 1
 NORMALLY DISTRIBUTED DEMANDS

FV _{HEUR}				
Problem size (m x n)	Run 1	Run 2	Run 3	Run 4
5 x 6	1607.78	2168.35	3381.72	4549.08
8 x 10	2615.44	3648.72	5588.88	7497.83
10 x 15	4278.69	5783.66	8702.03	11755.04
15 x 20	5926.66	8043.87	11894.23	16012.18

TABLE 5.14
 RESULTS OF SENSITIVITY ANALYSIS FOR PROBLEMS
 WITH CAPACITATED PLANTS AND TYPE 2
 NORMALLY DISTRIBUTED DEMANDS

FV _{HEUR}				
Problem size (m x n)	Run 1	Run 2	Run 3	Run 4
5 x 6	1654.96	2182.73	3441.96	4564.09
8 x 10	2737.63	3739.46	5642.27	7717.65
10 x 15	4484.61	5738.11	8960.50	11710.87
15 x 20	6086.40	8010.54	11940.67	15936.32

TABLE 5.15
RESULTS OF SENSITIVITY ANALYSIS FOR PROBLEMS
WITH CAPACITATED PLANTS AND UNIFORMLY
DISTRIBUTED DEMANDS

FV _{HEUR}				
Problem size (m x n)	Run 1	Run 2	Run 3	Run 4
5 x 6	1458.48	2001.45	3185.24	4374.87
8 x 10	2473.95	3410.60	5357.13	7371.63
10 x 15	4045.98	5362.54	8405.01	11294.66
15 x 20	5334.21	7115.90	11194.15	15085.60

decreased. However, the increase in functional values is more with the increase in prices than with the increase in mean of the demand distributions. As it is expected, the increase in functional value by increasing the variations in both prices and mean of the demand distributions is significantly more than increasing only the variations of either one of the parameters. This can be seen by comparing the increase in functional values of runs 2 and 3 over 1 against that of run 4 over 1 in Table 5.13.

The only difference between the type 1 and type 2 normally distributed demands is that the standard deviation is also varied in the latter while it is kept constant in the former. With the type 2, the functional values increased with the increase in mean and standard deviation of the demand distributions, keeping the price variations constant.³ Similarly, the functional values increased with the increase in price variations, keeping the variations in mean and standard deviation of the demand distributions constant.⁴ Also, as in type 1 normally distributed demands, the changes in model responses given by the objective function values seem more sensitive to price variations than variations in mean and standard deviation of the demand distributions. Comparison of runs 2 and 3 in Table 5.14 evidences this fact. Finally, the increase in functional values by increasing both prices and mean and standard deviation of the demand distributions is significantly more than increasing only the variations of either one of the parameters. This can be observed by comparing the increase in

³Compare run 1 with run 2 and run 3 with run 4 in Table 5.14.

⁴Compare run 1 with run 3 and run 2 with run 4 in Table 5.14.

functional values of runs 2 and 3 over 1 against that of run 4 over 1 in Table 5.14.

The results for the uniformly distributed demands are presented in Table 5.15. The structure of the variations in prices and the mean and standard deviation of the uniformly distributed demands is similar to that of type 2 normally distributed demands. As such, the same reasoning and comparison documented for type 2 holds true for uniformly distributed demands as well.

Results of the Sensitivity Analysis for Problem Type 7

The results of the sensitivity analysis for the uncapacitated problems are presented in Tables 5.16 through 5.18. As the changes in functional values with the three types of demand distributions follow the same pattern as in problem type 8, the discussion is not repeated here. One important point to note though is the corresponding problem for each demand distribution in type 7 has an objective function value greater than that of type 8. The reason that can be attributed to this observation is type 7 being uncapacitated is less restrictive. As such, improved optimal supply patterns leading to eventual increase in profits are realized.

Model Validation

There have not been any attempts in the past to develop a model that included the dimensions of price sensitivity and stochastic demands. Even the model that incorporated the dimension of stochastic demand, [29], considered an objective of minimizing the expected total costs as

TABLE 5.16
RESULTS OF SENSITIVITY ANALYSIS FOR PROBLEMS
WITH UNCAPACITATED PLANTS AND TYPE 1
NORMALLY DISTRIBUTED DEMANDS

Problem size (m x n)	FV _{HEUR}			
	Run 1	Run 2	Run 3	Run 4
5 x 6	1735.62	2411.51	3514.89	4793.35
8 x 10	3076.35	4276.92	6052.38	8252.95
10 x 15	4889.41	6715.97	9354.40	12680.96
15 x 20	6610.13	9101.64	12567.54	17059.05

TABLE 5.17
 RESULTS OF SENSITIVITY ANALYSIS FOR PROBLEMS
 WITH UNCAPACITATED PLANTS AND TYPE 2
 NORMALLY DISTRIBUTED DEMANDS

FV _{HEUR}				
Problem size (m x n)	Run 1	Run 2	Run 3	Run 4
5 x 6	1809.82	2483.58	3596.24	4865.91
8 x 10	3120.66	4302.53	6102.26	8279.35
10 x 15	4908.65	6695.34	9378.99	12658.57
15 x 20	6579.77	9083.13	12548.55	17043.44

TABLE 5.18
 RESULTS OF SENSITIVITY ANALYSIS FOR PROBLEMS
 WITH UNCAPACITATED PLANTS AND UNIFORMLY
 DISTRIBUTED DEMANDS

FV _{HEUR}				
Problem size (m x n)	Run 1	Run 2	Run 3	Run 4
5 x 6	1540.08	2134.26	3305.05	4489.26
8 x 10	2656.67	3741.19	5597.44	7697.38
10 x 15	4397.21	6132.25	8837.02	12068.97
15 x 20	5978.47	8312.40	11923.99	16245.25

opposed to an objective of maximizing the expected net profits featured by the proposed model. Thus, it is impossible to perform any model validation by comparing the results of the proposed model with those of any existing models. Therefore, only a thorough check is undertaken to determine the appropriateness of the parameters used in the model and their relationships. It is emphasized that all of these checks are focused on the revised model for problem types 8 and 7. The validity of the heuristic solution algorithm has already been tested for several test problems by comparing the functional values determined by the heuristic with those obtained by either one of the optimizing algorithms. This test was carried out on both problem types to an extent that both optimizing algorithms fail to determine the optimal binary solution within a reasonable length of computation time. The results have shown that the heuristic algorithm solves both problem types reasonably well, producing solutions that are only 0-3 percent off from the optimal objective function values. Descriptions on the checks performed to determine the appropriateness of some of the parameters used in the revised model and their relationships are presented below.

Appropriateness of x_{ij}^*

Given that x_{ij} is only limited by $x_{ij} \geq 0$, the optimal supply quantities are determined from

$$\int_{x_{ij}}^{ub_{ij}} f(q_{ij}) dq_{ij} = \frac{r_{ij}}{p_{ij}} \quad (5.2)$$

First, the previous equation holds true only if the condition $r_{ij} \leq p_{ij}$ is satisfied. Adequate care is taken in the computer program when data on both parameters r_{ij} and p_{ij} were randomly generated. In all of the test problems, the values generated for p_{ij} were greater than those generated for r_{ij} . Second, with normally distributed demands, it may also be possible to determine negative optimal supply quantities which is not realistic. As an example, for $\mu_{q_{ij}} = 8$, $\sigma_{q_{ij}} = 5$, and a ratio of $\frac{r_{ij}}{p_{ij}} = 0.95$, the value of $x_{ij}^* = -0.25 < 0$. Both with type 1 and type 2 normally distributed demands the parameters were carefully chosen in all of the test problems to avoid this result.

Relationship Between p_{ij} and q_{ij}

It is necessary that the concept of "elasticity of demand"⁵ be observed in all of the test problems solved. This is accomplished by instituting the previously noted assumption in the computer program. That is, the influence of price sensitivity is only reflected in the mean of the distribution of demand. A hyperbolic relationship was used for this purpose.

Summary

This chapter has clearly documented the important steps taken toward the accomplishment of both primary and secondary objectives of this research. The relationships between the parameters of the revised model, the data required, and the need for having to use randomly generated data are reported clearly. Computational experiences with small, medium, and large size problems with the heuristic algorithm

⁵See page 66.

and the two optimizing algorithms are also reported. Both capacitated and uncapacitated problems descriptive of problems types 8 and 7, respectively, are analyzed separately. The computational results show that a good heuristic algorithm producing near optimal solutions is preferred for solving medium and large size plant location-allocation problems in the presence of price sensitive stochastic demands. The sensitivity analysis performed on several test problems appears to determine functional values that are comparable with solutions one would anticipate if the model were to be applied to real world situations.

CHAPTER VI

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Prior to this study, problem types 7 and 8 identified and reported in Chapter I were the only ones that had not been attempted among the eight problems that are characterized as single product and single period. This research has made a significant extension to the state of the art by completing the investigation of all eight problems. Moreover, this study is the first of its kind in incorporating the key dimensions of price sensitivity and stochastic demand that led to the investigation of a more comprehensive problem.

In order to fulfill the primary and secondary objectives, the following accomplishments have been achieved.

1. A comprehensive model for problem type 8 that focuses on maximizing the expected profits has been developed.
2. Due to the complexities involved in solving this model, model simplifications were instituted. As a result, a revised model that is manageable and easy to solve has been developed.
3. An appropriate model for problem type 7 has been deduced from the revised model developed for problem type 8.
4. A heuristic algorithm and an optimal algorithm based on the extreme point ranking technique have been developed for solving both problem types. In addition, an existing optimizing algorithm based on the branch-and-bound technique has been used to compare the performances of both optimizing algorithms.

5. The relationships between price and the quantity demanded introduced by the dimension of price sensitivity have been taken into account. Both normal and uniform distribution of demands have been considered. The normal distribution of demand has further been extended to type 1 and type 2 due to the assumptions made on the relationships between the model parameters.
6. Test problems have been categorized into small, medium, and large. Several test problems have been solved with the three demand distributions for both problem types. Computational results with the heuristic and the two optimizing algorithms have been reported. Appropriateness of the heuristic algorithm for solving medium and large size problems has been identified. Reasonable computation time and the capability of producing near optimal solutions acceptable to the management have been found to be the two major factors supportive of the heuristic algorithm. The superiority of the branch-and-bound technique over the extreme point ranking technique has been recognized due to lower memory and computation time requirements. Although the savings in computation time have not been very significant, it has been shown that aiding the optimizing algorithm with a good heuristic can improve upon the efficiency of the search.
7. A sensitivity analysis has been performed separately on test problems with the three different demand distributions for both problem types. The correctness in the performance of the algorithmic steps of the heuristic has been established from the results produced by this analysis.

Based on the results obtained from this research, it can be concluded that a good heuristic that produces near optimal solutions within

reasonable limits of percentage deviation is preferred for solving medium and large size problems. With capacitated plants, the incapability of finding an optimal solution even for a medium size problem using an optimizing algorithm further evidences this fact.

In a real world application, it is of interest to know the effect of using incorrect estimates of the input parameters. Also, if past experience or expert opinion has been used previously to implement the decisions on location and allocation, the model developed in this research can be used to determine the effect of using incorrect values of decision variables, if such were the case. In summary, prior to any proposed use of this model and the subsequent implementation of decisions in a real world situation, it is necessary to know:

1. The sensitivity of the model to the use of incorrect (non-optimal) values of decision variables.
2. The sensitivity of the model to the use of incorrect estimates of the input parameters.
3. The sensitivity of the model to the use of incorrect properties of the system that define the model.

Another area for future research can be identified from the problem types presented in Table 1.1. For a non-perishable product, considering a problem that is attributed with multi-period is more meaningful than considering one with single-period. The remaining items at the end of the period have value and the implications of carrying them over to the next period should be considered. As such, problem types 11 and 12 and 13 and 14 that include one of the key dimensions are potential problems for future research.

BIBLIOGRAPHY

- [1] Baumol, W. J., and P. Wolfe. "A Warehouse-Location Problem." Operations Research, 6, 3 (1958), 252-263.
- [2] Bazarraa, M. S., and J. J. Jarvis. Linear Programming and Network Flows. New York: John Wiley & Sons, Inc., 1977.
- [3] Bernd, D. P. "The Transportation Problem with Stochastic Demand." (Unpub. Ph.D. dissertation, Arizona State University, 1972.)
- [4] Bhalla, H. S. "Multi-Time Period, Facilities Location Problems: A Heuristic Algorithm With Applications to Waste Water Treatment Systems." (Unpub. Ph.D. dissertation, University of Massachusetts, 1970.)
- [5] Brown, W. J. "A Generalized Location Problem." (Unpub. Ph.D. dissertation, Case Western Reserve University, 1973.)
- [6] Charnes, A., and W. W. Cooper. "Deterministic Equivalents for Optimizing and Satisficing Under Chance Constraints." Operations Research, 11, 1 (1963), 18-39.
- [7] Cho, C. C. "Fixed Charge Capacitated Facility Location-Allocation Problem: Single and Multi-Period Cases." (Unpub. Ph.D. dissertation, University of Wisconsin at Madison, 1975.)
- [8] Cooper, L. "Location-Allocation Problems." Operations Research, 11, 3 (1963), 331-343.
- [9] Cornuejols, G. P. "Analysis of Algorithms for a Class of Location Problems." (Unpub. Ph.D. dissertation, Cornell University, 1978.)
- [10] Curry, G. L., and R. W. Skeith. "A Dynamic Programming Algorithm for Facility Location and Allocation." AIIE Transactions, 1, 2 (1969), 133-138.
- [11] Dantzig, G. B., and P. Wolfe. "The Decomposition Algorithm for Linear Programming." Econometrica, 29, 4 (1961), 767-778.
- [12] Dantzig, G. B. Linear Programming and Extensions. Princeton, New Jersey: Princeton University Press, 1963.
- [13] Davis, P. S., and T. L. Ray. "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem." Naval Research Logistics Quarterly, 16, 3 (1965), 331-344.

- [14] Drysdale, J. K., and P. J. Sandiford. "Heuristic Warehouse Location - A Case History Using a New Method." Canadian Operational Research Society Journal, 7, 1 (1969), 45-61.
- [15] Efromson, M. A., and T. L. Ray. "A Branch-Bound Algorithm for Plant Location." Operations Research, 14, 3 (1966), 361-368.
- [16] Ellwein, L. B. Fixed Charge Location-Allocation Problems With Capacity and Configuration Constraints. Stanford, California: Stanford University, Dept. of Industrial Engineering Report 70-2, 1970.
- [17] Ellwein, L. B., and P. Gray. "Solving Fixed Charge Location-Allocation Problems With Capacity and Configuration Constraints." AIIE Transactions, 3, 4 (1971), 290-297.
- [18] Elmaghraby, S. E. "Allocation Under Uncertainty When the Demand Has Continuous D. F." Management Science, 6, 3 (1960), 270-294.
- [19] Erlenkotter, D. "Facility Location With Price-Sensitive Demands: Private, Public, and Quasi-Public." Management Science, 24, 4 (1977), 378-386.
- [20] Erlenkotter, D. "A Dual Based Procedure for Uncapacitated Facility Location." Operations Research, 26, 6 (1978), 992-1009.
- [21] Eschenbach, T. G. "The Multi-Period Location-Allocation Problem." (Unpub. Ph.D. dissertation, Stanford University, 1975.)
- [22] Feldman, E., Lehrer, F. A., and T. L. Ray. "Warehouse Location Under Continuous Economies of Scale." Management Science, 12, 9 (1966), 670-684.
- [23] Ferguson, A. R., and G. B. Dantzig. "The Allocation of Aircraft to Routes - An Example of Linear Programming Under Uncertain Demands." Management Science, 3, 1 (1956), 45-73.
- [24] Francis, R. L., and J. A. White. Facility Layout and Location - An Analytical Approach. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1974.
- [25] Francis, R. L., and J. M. Goldstein. "Location Theory: A Selective Bibliography." Operations Research, 22, 2 (1974), 400-410.
- [26] Freund, R. J. "The Introduction of Risk Into a Programming Model." Econometrica, 24, 3 (1956), 253-263.
- [27] Garvin, W. W. Introduction to Linear Programming. New York: McGraw-Hill, Inc., 1960.

- [28] Gerson, M. L., and R. B. Maffei. "Technical Characteristics of Distribution Simulators." Management Science, 10, 1 (1963), 62-69.
- [29] Gonzalez-Valenzuela, F. "Simple and Capacitated Warehouse Location Problems with Stochastic Demands." (Unpub. Ph.D. dissertation, Stanford University, 1975.)
- [30] Gray, P. Mixed Integer Programming Algorithm for Site Selection and Other Fixed Charge Problems Having Capacity Constraints. Stanford, California: Stanford University, Dept. of Operations Research Technical Report 6, 1967.
- [31] Hadley, G., and T. M. Whitin. Analysis of Inventory Systems. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1963.
- [32] Hansen, P., and J.-F. Thisse. "Multiplant Location for Profit Maximization." Environment and Planning A, 9, (1977), 63-73.
- [33] Hildebrand, F. B. Advanced Calculus for Applications. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1976.
- [34] IBM Mathematical Programming System Extended/370 (MPSX/370) Basic Reference Manual, SH19-1127-0. Paris, France: Compagnie IBM France, 1976.
- [35] IBM Mathematical Programming System Extended/370 (MPSX/370), Mixed Integer Programming/370 (MIP/370) Program Reference Manual, SH19-1099-1. Paris, France: Compagnie IBM France, 1975.
- [36] Jucker, J. V., and R. C. Carlson. "The Simple Plant-Location Problem Under Uncertainty." Operations Research, 24, 6 (1976), 1045-1055.
- [37] Karanicolos, P. C. "Multiperiod Plant Location Problems." (Unpub. Ph.D. dissertation, Syracuse University, 1979.)
- [38] Karkazis, J., and T. B. Boffey. "The Multi-Commodity Facilities Location Problem." Journal of the Operational Research Society, 32, 9 (1981), 803-814.
- [39] Kelly, D. L. "An Analysis of the Multiperiod Warehouse Location Problem." (Unpub. Ph.D. dissertation, University of North Carolina at Chapel Hill, 1976.)
- [40] Khumawala, B. M. "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem." Management Science, 18, 12 (1972), B718-B731.
- [41] Kuehn, A. A., and M. J. Hamburger. "A Heuristic Program for Locating Warehouses." Management Science, 9, 11 (1963), 643-666.

- [42] Kuester, J. L., and J. H. Mize. Optimization Techniques With FORTRAN. New York: McGraw-Hill, 1973.
- [43] Manne, A. S. "Plant Location Under Economies-of-Scale-Decentralization and Computation." Management Science, 11, 2 (1964), 213-235.
- [44] Marks, D. H. "Facility Location and Routing Models in Solid Waste Collection Systems." (Unpub. Ph.D. dissertation, The Johns Hopkins University, 1969.)
- [45] McGinnis, L. F. "Approximate and Exact Solution Procedures for a Class of Facilities Location Problems." (Unpub. Ph.D. dissertation, North Carolina State University at Raleigh, 1975.)
- [46] Neebe, A. W., and B. M. Khumawala. "An Improved Algorithm for the Multi-Commodity Location Problem." Journal of the Operational Research Society, 32, 2 (1981), 143-149.
- [47] Renyi, A. Probability Theory. New York: American Elsevier Publishing Co., Inc., 1970.
- [48] ReVelle, C., Marks, D., and J. C. Liebman. "An Analysis of Private and Public Sector Location Models." Management Science, 16, 11 (1970), 692-707.
- [49] Sa, G. "Branch-and-Bound and Approximate Solutions to the Capacitated Plant Location Problem." Operations Research, 17, 6 (1969), 1005-1016.
- [50] Samuelson, P. A. "Spatial Price Equilibrium and Linear Programming." American Economic Review, 42, 3 (1952), 283-303.
- [51] Shannon, R. E., and J. P. Ignizio. "A Heuristic Algorithm for Warehouse Location." AIIE Transactions, 2, 4 (1970), 334-339.
- [52] Shycon, H. N., and R. B. Maffei. "Simulation-Tool for Better Distribution." Harvard Business Review, 38, 6 (1960), 65-75.
- [53] Spielberg, K. "Plant Location With Generalized Search Origin." Management Science, 16, 3 (1969), 165-178.
- [54] Spielberg, K. "Algorithms for the Simple Plant-Location Problem With Some Side Conditions." Operations Research, 17, 1 (1969), 85-111.
- [55] Swarc, W. "The Transportation Problem With Stochastic Demand." Management Science, 11, 1 (1964), 33-50.
- [56] Taha, H. A. Integer Programming Theory, Applications, and Computations. New York: Academic Press, Inc., 1975.

- [57] Truscott, W. G. "A Profit Maximization Approach for Facility Location in a Multi-Product Environment." (Unpub. Ph.D. dissertation, Indiana University, 1973.)
- [58] Truscott, W. G. "The Treatment of Revenue Generation Effects of Facility Location." AIEE Transactions, 7, 1 (1975), 63-69.
- [59] Warszawski, A. "Optimization of Location on the Building Site." (Unpub. Ph.D. dissertation, Technion-Israel Institute of Technology, 1971.)
- [60] Warszawski, A. "Multidimensional Location Problems." Operational Research Quarterly, 24, 2 (1973), 165-179.
- [61] Warszawski, A., and S. Peer. "Optimizing the Location of Facilities on a Building Site." Operational Research Quarterly, 24, 1 (1973), 35-44.
- [62] Williams, A. C. "A Stochastic Transportation Problem." Operations Research, 11, 5 (1963), 759-770.
- [63] Wilson, D. "An a Priori Bounded Model for Transportation Problems With Stochastic Demand and Integer Solutions." AIEE Transactions, 4, 3 (1972), 186-193.
- [64] Wilson, D. "Tighter Bounds for Stochastic Transportation Models." AIEE Transactions, 5, 2 (1973), 180-185.
- [65] Yagiz, O. "Heuristically-Aided Single-Branch Search Procedures for Plant Location Problems." (Unpub. Ph.D. dissertation, University of North Carolina at Chapel Hill, 1973.)

APPENDIX

FORTRAN PROGRAMS LISTING

```

C*****0000100
C
C      THIS PROGRAM PERFORMS THE HEURISTIC ALGORITHM FOR PLANT      0000200
C      LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE             0000300
C      STOCHASTIC DEMAND                                           0000400
C                                                                    0000500
C      ALSO IT SETS UP THE INPUT DATA DECK FOR EXTREME POINT      0000600
C      RANKING TECHNIQUE                                           0000700
C                                                                    0000800
C                                                                    0000900
C      THE PLANTS ARE ASSUMED CAPACITATED                          0001000
C      AND A TYPE 1 NORMALLY DISTRIBUTED DEMAND IS ASSUMED         0001100
C                                                                    0001200
C      THIS ALGORITHM TAKES INTO ACCOUNT OF THE FACT THAT          0001300
C      ALL DEMAND CENTERS NEED TO BE ALLOCATED AND THAT            0001400
C      EACH DEMAND CENTER MUST RECEIVE THE SUPPLY                   0001500
C      FROM AT MOST ONE PLANT                                       0001600
C                                                                    0001700
C                                                                    0001800
C      WRITTEN BY LOGENDRAN RASARATNAM                             0001900
C      SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT              0002000
C      OKLAHOMA STATE UNIVERSITY                                    0002100
C                                                                    0002200
C      DISSERTATION ADVISER: DR. M. PALMER TERRELL                 0002300
C                                                                    0002400
C      VERSION 1 -- AUGUST, 1984                                    0002500
C*****0002600
C                                                                    0002700
C*****0002800
C                                                                    0002900
C*** GENERAL STRUCTURE AND INPUT REQUIREMENTS:                     0003000
C (MAIN PROGRAM DRIVES THE SUBROUTINES UNFRM AND GLOBE)            0003100
C (UNFRM DRIVES SUBROUTINE RANDU)                                  0003200
C (GLOBE DRIVES SUBROUTINE ALOC)                                   0003300
C (ALOC DRIVES SUBROUTINE PARTL)                                  0003400
C                                                                    0003500
C*****0003600
C                                                                    0003700
C      SUBROUTINE          FUNCTION                                0003800
C      -----          -----                                0003900
C      RANDU              TO GENERATE A UNIFORMLY DISTRIBUTED      0004000
C                        RANDOM NUMBER BETWEEN 0 AND 1             0004100
C      UNFRM              TO CONVERT THE RANDOM NUMBER GENERATED   0004200
C                        BY RANDU BETWEEN APPROPRIATE LIMITS      0004300
C                        ESTABLISHED FOR THE UNIFORM               0004400
C                        DISTRIBUTION TO BE USED                   0004500
C      GLOBE              PERFORMS THE GLOBAL TESTS FOR OPTIMAL    0004600
C                        PROFITS AND FEASIBLE CAPACITY            0004700
C      ALOC               PERFORMS THE FEASIBLE ALLOCATIONS FOR    0004800
C                        FOR EACH OF THE DEMAND CENTERS           0004900
C                        DICTATED BY PRIORITY RULE 1              0005000
C      PARTL             PERFORMS THE PARTIAL TESTS FOR OPTIMAL    0005100
C                        PROFITS AND FEASIBLE CAPACITY            0005200
C                                                                    0005300
C*****0005400
C                                                                    0005500
C*** EXTERNAL FUNCTIONS REQUIRED                                     0005600
C (1) REGULAR SYSTEM SUPPLIED FORTRAN FUNCTIONS                   0005700
C (2) TWO IMSL SUBROUTINES                                         0005800
C      MDNOR -- CUMULATIVE PROBABILITY FUNCTION OF                 0005900
C              STANDARD NORMAL                                     0006000
C      MDNRIS -- INVERSE FUNCTION OF MDNOR                         0006100
C                                                                    0006200
C*****0006300
C                                                                    0006400
C*** COMMON BLOCK VARIABLE DEFINITIONS                             0006500
C ONLY THOSE VARIABLES THAT REQUIRE EXPLANATION ARE LISTED       0006600
C                                                                    0006700
C /BLOCK1/                                                         0006800
C                                                                    0006900
C      TOFIX - TOTAL FIXED COST FOR THE ORIGINAL LOCATION VECTOR  0007000
C      UNDER CONSIDERATION                                         0007100

```

C	TFIX	- TOTAL FIXED COST FOR THE LOCATION VECTOR PRESENTLY	00007200
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007300
C		FROM THE ORIGINAL LOCATION VECTOR	00007400
C	TFFIX	- TOTAL FIXED COST FOR THE FINAL LOCATION VECTOR	00007500
C		CURRENTLY STORED	00007600
C	TRORG	- TOTAL REVENUE FOR THE ORIGINAL LOCATION VECTOR	00007700
C		UNDER CONSIDERATION	00007800
C	TREV	- TOTAL REVENUE FOR THE LOCATION VECTOR PRESENTLY	00007900
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008000
C		FROM THE ORIGINAL LOCATION VECTOR	00008100
C	TRFIN	- TOTAL REVENUE FOR THE FINAL LOCATION VECTOR	00008200
C		CURRENTLY STORED	00008300
C	TPORG	- TOTAL PROFIT FOR THE ORIGINAL LOCATION VECTOR	00008400
C		UNDER CONSIDERATION	00008500
C	TPROF	- TOTAL PROFIT FOR THE LOCATION VECTOR PRESENTLY	00008600
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008700
C		FROM THE ORIGINAL LOCATION VECTOR	00008800
C	TPFIN	- TOTAL PROFIT FOR THE FINAL LOCATION VECTOR	00008900
C		CURRENTLY STORED	00009000
C			00009100
C	/BLOCK2/		00009200
C			00009300
C	IVECT1-	INDICATOR VARIABLE FOR THE TEST ON FEASIBLE CAPACITY	00009400
C	IVECT2-	INDICATOR VARIABLE FOR THE TEST ON OPTIMAL PROFITS	00009500
C	NORG	- VARIABLE INDICATING THAT THE ORIGINAL FEASIBLE	00009600
C		SOLUTION HAS BEEN FOUND	00009700
C	NCON	- TOTAL NUMBER OF PLANTS DROPPED SO FAR	00009800
C	NALOC	- TOTAL NUMBER OF DEMAND CENTERS ALLOCATED SO FAR	00009900
C	MPT	- TOTAL NUMBER OF PLANTS FOR THE PROBLEM SOLVED	00010000
C	NCT	- TOTAL NUMBER OF DEMAND CENTERS FOR THE PROBLEM SOLVED	00010100
C			00010200
C	/BLOCK3/		00010300
C			00010400
C	NYD	- THE ORIGINAL LOCATION VECTOR UNDER CONSIDERATION	00010500
C	NY	- THE LOCATION VECTOR PRESENTLY CONSIDERED BY	00010600
C		DROPPING ONE PLANT AT A TIME FROM THE	00010700
C		ORIGINAL LOCATION VECTOR	00010800
C	NYF	- THE FINAL LOCATION VECTOR CURRENTLY STORED	00010900
C	NPLANT-	VARIABLE USED FOR THE PLANTS IN RANKING THE OPTIMAL	00011000
C		PROFITS EMPLOYING PRIORITY RULE 1	00011100
C	NCENT	- VARIABLE USED FOR THE DEMAND CENTERS IN RANKING THE	00011200
C		OPTIMAL PROFITS EMPLOYING PRIORITY RULE 1	00011300
C	ICENT	- VARIABLE INDICATING WHETHER A DEMAND CENTER HAS	00011400
C		ALREADY BEEN ALLOCATED	00011500
C	IDROP	- VARIABLE INDICATING THE PLANT PRESENTLY DROPPED	00011600
C		FROM THE ORIGINAL LOCATION VECTOR	00011700
C	NDROP	- VARIABLE USED FOR RANKING THE PLANTS FOR DROP	00011800
C		EMPLOYING PRIORITY RULE 2	00011900
C			00012000
C	/BLOCK4/		00012100
C			00012200
C	XSTAR	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR EACH	00012300
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00012400
C	XO	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE ORIGINAL	00012500
C		LOCATION VECTOR UNDER CONSIDERATION	00012600
C	X	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE LOCATION	00012700
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00012800
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00012900
C	XF	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE FINAL	00013000
C		LOCATION VECTOR CURRENTLY STORED	00013100
C	PR	- OPTIMAL PROFITS DETERMINED FOR EACH COMBINATION OF	00013200
C		DEMAND CENTERS AND PROFITS	00013300
C	PROF	- THE RANKED OPTIMAL PROFITS DETERMINED FOR EACH	00013400
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00013500
C		USING PRIORITY RULE 1	00013600
C			00013700
C	/BLOCK5/		00013800
C			00013900
C	NZO	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE ORIGINAL	00014000
C		LOCATION VECTOR UNDER CONSIDERATION	00014100
C	NZ	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE LOCATION	00014200

```

C          VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT      00014300
C          AT A TIME FROM THE ORIGINAL LOCATION VECTOR            00014400
C          NZF - COMPONENTS OF THE ALLOCATION MATRIX FOR THE FINAL  00014500
C          LOCATION VECTOR CURRENTLY STORED                        00014600
C                                                                00014700
C /BLOCK6/                                                       00014800
C                                                                00014900
C          A - CAPACITY OF EACH PLANT                             00015000
C          F - FIXED COST OF EACH PLANT                           00015100
C          ADROP- THE REMAINING CAPACITY OF EACH PLANT AFTER HAVING 00015200
C          ALLOCATED A SUBSET OF DEMAND CENTERS                   00015300
C                                                                00015400
C*****00015500
C                                                                00015600
C*** OTHER VARIABLE DEFINITIONS                                  00015700
C                                                                00015800
C          MU - MEAN OF THE NORMAL DISTRIBUTION OF DEMAND         00015900
C          DETERMINED FOR EACH COMBINATION OF                     00016000
C          DEMAND CENTERS AND PLANTS                              00016100
C          SIGMA- STANDARD DEVIATION OF THE NORMAL DISTRIBUTION OF 00016200
C          DEMAND DETERMINED FOR EACH COMBINATION OF             00016300
C          DEMAND CENTERS AND PLANTS                              00016400
C          R - PER UNIT VARIABLE COST OF THE PRODUCT SUPPLIED FOR 00016500
C          EACH COMBINATION OF DEMAND CENTERS AND PLANTS         00016600
C          PC - PER UNIT PRICE OF THE PRODUCT RECEIVED FOR EACH  00016700
C          COMBINATION OF DEMAND CENTERS AND PLANTS              00016800
C*****00016900
C          INTEGER RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,      00017000
C          *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00017100
C          *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)  00017200
C          REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00017300
C          *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROP,ADROP,  00017400
C          *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,  00017500
C          *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,  00017600
C          *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00017700
C          *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),  00017800
C          *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00017900
C          DATA RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS/2HRO,2HMA,  00018000
C          *2HFI,2HEO,1HO,3HPRO,2HFT,3HEQU,3HINE,1HZ,1HY,1H+/  00018100
C          COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,  00018200
C          *TPFIN,TPROF                                          00018300
C          COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO  00018400
C          *,KK,JJ                                              00018500
C          COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP  00018600
C          COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF                  00018700
C          COMMON/BLOCK5/NZO,NZ,NZF                              00018800
C          COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS         00018900
C          COMMON/BLOCK7/B,C,D                                   00019000
C          COMMON/BLOCK8/IX,IY,YFL                              00019100
C*****00019200
C          NO=12                                                 00019300
C          NOT=14                                                00019400
C          MPT=3                                                  00019500
C          NCT=4                                                  00019600
C          CONST1=1.0                                            00019700
C          CONST2=0.0                                            00019800
C          MNTOT=MPT*NCT                                         00019900
C          MQ=NCT+MPT                                            00020000
C          MR=MPT*NCT+MPT                                        00020100
C*****00020200
C          GENERATE VALUES RANDOMLY FOR F(MM)                  00020300
C          SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM 00020400
C          DISTRIBUTION                                          00020500
C*****00020600
C          B=200.                                                00020700
C          C=300.                                                00020800
C          IX=13                                                 00020900
C          WRITE(NO,4025)                                        00021000
C          4025 FORMAT(//5X,'THE PLANT FIXED COSTS ARE',//)     00021100
C          WRITE(NO,5248)                                        00021200
C          5248 FORMAT(5X,'FIXED COST',5X,'PLANT'//)           00021300

```



```

C*****
DO 900 MM=1,MPT
DO 901 LL=1,NCT
P=1.-R(MM,LL)/PC(MM,LL)
CALL MDNRIS(P,S,IER)
XSTAR(MM,LL)=S*SIGMA(MM,LL)+MU(MM,LL)
IF(XSTAR(MM,LL)-O.) 902,902,903
903 C1=PC(MM,LL)*MU(MM,LL)
TWOPI=2.*3.141593
FAC1=SQRT(TWOPI)
FAC2=1./FAC1
FAC3=XSTAR(MM,LL)-MU(MM,LL)
FAC4=FAC3/SIGMA(MM,LL)
FAC5=- (FAC4**2)/2.
FAC6=EXP(FAC5)
FAC7=-PC(MM,LL)*SIGMA(MM,LL)
C2=FAC7*FAC2*FAC6
S=FAC4
CALL MDNOR(S,P)
FAC8=1.-P
C3=PC(MM,LL)*FAC3*FAC8
C4=-R(MM,LL)*XSTAR(MM,LL)
PR(MM,LL)=C1+C2+C3+C4
IF(PR(MM,LL)-O.) 902,902,901
902 XSTAR(MM,LL)=O.
PR(MM,LL)=O.
901 CONTINUE
900 CONTINUE
WRITE(NO,4029)
4029 FORMAT(//5X,'THE OPTIMAL SUPPLY QUANTITIES AND THE PROFITS ARE',//
*)
WRITE(NO,5256)
5256 FORMAT(1X,'OPTIMAL SUPPLY QUANTITY',8X,'PROFIT',8X,'PLANT',5X,
*'DEMAND CENTER'//)
DO 3003 MM=1,MPT
DO 3004 LL=1,NCT
3004 WRITE(NO,5257) XSTAR(MM,LL),PR(MM,LL),MM,LL
5257 FORMAT(4X,F14.6,9X,F14.6,6X,I2,13X,I2//)
3003 CONTINUE
C*****
C GENERATE THE INPUT DATA DECK FOR THE
C EXTREME POINT RANKING TECHNIQUE
C*****
WRITE(NOT,137) MQ,MR
137 FORMAT(2I5)
WRITE(NOT,138) RO
138 FORMAT(1X,A2)
WRITE(NOT,139) ZERO,PRO,FT
139 FORMAT(11X,A1,1X,A3,A2)
DO 141 LL=1,NCT
IF(LL-9) 142,142,143
142 WRITE(NOT,144) ZERO,EQU,ZERO,LL
144 FORMAT(11X,A1,1X,A3,A1,I1)
GO TO 141
143 WRITE(NOT,145) ZERO,EQU,LL
145 FORMAT(11X,A1,1X,A3,I2)
141 CONTINUE
DO 146 MM=1,MPT
IF(MM-9) 147,147,148
147 WRITE(NOT,144) POS,INE,ZERO,MM
GO TO 146
148 WRITE(NOT,145) POS,INE,MM
146 CONTINUE
WRITE(NOT,149) MA
149 FORMAT(1X,A2)
DO 150 LL=1,NCT
DO 151 MM=1,MPT
IF(MM-9) 152,152,153
152 IF(LL-9) 154,154,155
154 WRITE(NOT,733) Z,ZERO,MM,ZERO,LL,PRO,FT,PR(MM,LL)
733 FORMAT(7X,A1,A1,I1,A1,I1,1X,A3,A2,1X,D20.10)
00028500
00028600
00028700
00028800
00028900
00029000
00029100
00029200
00029300
00029400
00029500
00029600
00029700
00029800
00029900
00030000
00030100
00030200
00030300
00030400
00030500
00030600
00030700
00030800
00030900
00031000
00031100
00031200
00031300
00031400
00031500
00031600
00031700
00031800
00031900
00032000
00032100
00032200
00032300
00032400
00032500
00032600
00032700
00032800
00032900
00033000
00033100
00033200
00033300
00033400
00033500
00033600
00033700
00033800
00033900
00034000
00034100
00034200
00034300
00034400
00034500
00034600
00034700
00034800
00034900
00035000
00035100
00035200
00035300
00035400
00035500

```

```

WRITE(NOT,734) Z,ZERO,MM,ZERO,LL,EQU,ZERO,LL,CONST1          00035600
734  FORMAT(7X,A1,A1,I1,A1,I1,1X,A3,A1,I1,1X,D20.10)          00035700
WRITE(NOT,734) Z,ZERO,MM,ZERO,LL,INE,ZERO,MM,XSTAR(MM,LL)    00035800
      GO TO 151                                                00035900
155  WRITE(NOT,736) Z,ZERO,MM,LL,PRO,FT,PR(MM,LL)            00036000
736  FORMAT(7X,A1,A1,I1,I2,1X,A3,A2,1X,D20.10)              00036100
WRITE(NOT,737) Z,ZERO,MM,LL,EQU,LL,CONST1                    00036200
737  FORMAT(7X,A1,A1,I1,I2,1X,A3,I2,1X,D20.10)              00036300
WRITE(NOT,738) Z,ZERO,MM,LL,INE,ZERO,MM,XSTAR(MM,LL)        00036400
738  FORMAT(7X,A1,A1,I1,I2,1X,A3,A1,I1,1X,D20.10)          00036500
      GO TO 151                                                00036600
153  IF(LL-9) 157,157,158                                     00036700
157  WRITE(NOT,739) Z,MM,ZERO,LL,PRO,FT,PR(MM,LL)            00036800
739  FORMAT(7X,A1,I2,A1,I1,1X,A3,A2,1X,D20.10)              00036900
WRITE(NOT,740) Z,MM,ZERO,LL,EQU,ZERO,LL,CONST1              00037000
740  FORMAT(7X,A1,I2,A1,I1,1X,A3,A1,I1,1X,D20.10)          00037100
WRITE(NOT,756) Z,MM,ZERO,LL,INE,MM,XSTAR(MM,LL)             00037200
756  FORMAT(7X,A1,I2,A1,I1,1X,A3,I2,1X,D20.10)            00037300
      GO TO 151                                                00037400
158  WRITE(NOT,742) Z,MM,LL,PRO,FT,PR(MM,LL)                00037500
742  FORMAT(7X,A1,I2,I2,1X,A3,A2,1X,D20.10)                00037600
WRITE(NOT,743) Z,MM,LL,EQU,LL,CONST1                        00037700
743  FORMAT(7X,A1,I2,I2,1X,A3,I2,1X,D20.10)                00037800
WRITE(NOT,743) Z,MM,LL,INE,MM,XSTAR(MM,LL)                  00037900
151  CONTINUE                                                00038000
150  CONTINUE                                                00038100
      DO 159 MM=1,MPT                                         00038200
          FMS(MM)=-F(MM)                                       00038300
          AMS(MM)=-A(MM)                                       00038400
          IF(MM-9) 160,160,161                                 00038500
160  WRITE(NOT,745) Y,ZERO,ZERO,ZERO,MM,PRO,FT,FMS(MM)      00038600
WRITE(NOT,746) Y,ZERO,ZERO,ZERO,MM,INE,ZERO,MM,AMS(MM)      00038700
      GO TO 159                                                00038800
161  WRITE(NOT,747) Y,ZERO,ZERO,MM,PRO,FT,FMS(MM)          00038900
WRITE(NOT,748) Y,ZERO,ZERO,MM,INE,MM,AMS(MM)                00039000
745  FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A2,1X,D20.10)          00039100
746  FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A1,I1,1X,D20.10)      00039200
747  FORMAT(7X,A1,A1,A1,I2,1X,A3,A2,1X,D20.10)            00039300
748  FORMAT(7X,A1,A1,A1,I2,1X,A3,I2,1X,D20.10)            00039400
159  CONTINUE                                                00039500
WRITE(NOT,749) FI                                           00039600
749  FORMAT(1X,A2)                                           00039700
      DO 162 LL=1,NCT                                         00039800
          IF(LL-9) 163,163,164                                 00039900
163  WRITE(NOT,755) EQU,ZERO,LL,CONST1                       00040000
      GO TO 162                                                00040100
164  WRITE(NOT,750) EQU,LL,CONST1                             00040200
755  FORMAT(13X,A3,A1,I1,1X,D20.10)                          00040300
750  FORMAT(13X,A3,I2,1X,D20.10)                             00040400
162  CONTINUE                                                00040500
      DO 165 MM=1,MPT                                         00040600
          IF(MM-9) 166,166,167                                 00040700
166  WRITE(NOT,751) INE,ZERO,MM,CONST2                       00040800
      GO TO 165                                                00040900
167  WRITE(NOT,752) INE,MM,CONST2                             00041000
751  FORMAT(13X,A3,A1,I1,1X,D20.10)                          00041100
752  FORMAT(13X,A3,I2,1X,D20.10)                             00041200
165  CONTINUE                                                00041300
WRITE(NOT,753) EO                                           00041400
753  FORMAT(1X,A2)                                           00041500
C*****
C SORT THE PROFITS IN A DECENDING ORDER
C*****
      K=0
      KONST=0
      DO 7 I=1,MPT
          DO 10 J=1,NCT
              K=KONST+J
              PROF(K)=PR(I,J)
              NPLANT(K)=I
              NCENT(K)=J

```



```

10  CONTINUE                                00042700
    KONST=I*NCT                             00042800
7   CONTINUE                                00042900
    MN=1                                     00043000
15  PROFM=PROF(MN)                          00043100
    NPLAM=NPLANT(MN)                        00043200
    NCENTM=NCENT(MN)                        00043300
    DO 20 I=MN,MNTOT                         00043400
      IF (PROFM-PROF(I)) 25,25,20           00043500
25  PROFM=PROF(I)                           00043600
    NPLAM=NPLANT(I)                         00043700
    NCENTM=NCENT(I)                         00043800
    IMAX=I                                   00043900
20  CONTINUE                                00044000
    PROF(IMAX)=PROF(MN)                     00044100
    NPLANT(IMAX)=NPLANT(MN)                 00044200
    NCENT(IMAX)=NCENT(MN)                   00044300
    PROF(MN)=PROFM                           00044400
    NPLANT(MN)=NPLAM                         00044500
    NCENT(MN)=NCENTM                         00044600
    MN=MN+1                                 00044700
    IF (MN-MNTOT) 15,15,35                  00044800
35  WRITE(NO,4030)                           00044900
4030 FORMAT(5X,'THE SORTED PROFITS AND THE RESPECTIVE PLANTS AND THE DEMO00045000
    *MAND CENTERS ARE'//)                   00045100
    WRITE(NO,4034)                           00045200
4034 FORMAT(5X,'RANKED PROFIT',10X,'PLANT',10X,'DEMAND CENTER'//) 00045300
    DO 5033 I=1,MNTOT                       00045400
5033 WRITE(NO,4035) PROF(I),NPLANT(I),NCENT(I) 00045500
4035 FORMAT(3X,F14.6,12X,I2,17X,I2//)       00045600
C*****                                     00045700
C SORT THE PLANTS IN A DECENDING ORDER OF FIXED COSTS 00045800
C*****                                     00045900
    DO 333 I=1,MPT                           00046000
      FDROP(I)=F(I)                           00046100
      ADROP(I)=A(I)                           00046200
333  NDROP(I)=I                               00046300
      MJ=1                                     00046400
37  FDROPM=FDROP(MJ)                         00046500
      ADROPM=ADROP(MJ)                       00046600
      NDROPM=NDROP(MJ)                       00046700
    DO 40 I=MJ,MPT                           00046800
      IF (FDROPM-FDROP(I)) 45,45,40          00046900
45  FDROPM=FDROP(I)                         00047000
      ADROPM=ADROP(I)                       00047100
      NDROPM=NDROP(I)                       00047200
      IMAX=I                                  00047300
40  CONTINUE                                00047400
      FDROP(IMAX)=FDROP(MJ)                  00047500
      ADROP(IMAX)=ADROP(MJ)                  00047600
      NDROP(IMAX)=NDROP(MJ)                  00047700
      FDROP(MJ)=FDROPM                       00047800
      ADROP(MJ)=ADROPM                       00047900
      NDROP(MJ)=NDROPM                       00048000
      IDROP(MJ)=1                            00048100
      MJ=MJ+1                                00048200
      IF (MJ-MPT) 37,37,50                   00048300
50  WRITE(NO,4031)                           00048400
4031 FORMAT(5X,'THE SORTED FIXED COSTS AND OTHER RESPECTIVE PARAMETERS 00048500
    *ARE'//)                                 00048600
    WRITE(NO,4032)                           00048700
4032 FORMAT(1X,'RANKED PLANTS FOR DROP',10X,'FIXED COST',11X,'CAPACITY'00048800
    *,11X,'FIXED COST',10X,'CAPACITY',12X,'IDROP VALUES'//) 00048900
    DO 5034 I=1,MPT                           00049000
5034 WRITE(NO,4033) NDROP(I),FDROP(I),ADROP(I),F(NDROP(I)), 00049100
    *A(NDROP(I)),IDROP(I)                   00049200
4033 FORMAT(10X,I2,17X,F14.6,6X,F14.6,6X,F14.6,6X,F14.6,10X,I7//) 00049300
C*****                                     00049400
C START WITH ALL PLANTS OPEN                 00049500
C SET ALL NY(I) EQUAL TO 1                  00049600
C*****                                     00049700

```

```

DO 55 K=1,MPT                                00049800
55  NY(NDROP(K))=1                            00049900
C*****                                       00050000
C INITIALIZE ALL PARAMETERS                    00050100
C*****                                       00050200
      TOFIX=0.0                                00050300
      TFFIX=0.0                                00050400
      TRORG=0.0                                00050500
      TRFIN=0.0                                00050600
      TPORG=0.0                                00050700
      TPFIN=0.0                                00050800
      IVECT1=1                                 00050900
      IVECT2=1                                 00051000
      NORG=0                                   00051100
      NCON=0                                   00051200
DO 610 MM=1,MPT                               00051300
  NYO(NDROP(MM))=0                            00051400
  NYF(NDROP(MM))=0                            00051500
  AF(NDROP(MM))=0.0                          00051600
  FF(NDROP(MM))=0.0                          00051700
DO 611 LL=1,NCT                               00051800
  XO(NDROP(MM),LL)=0.0                       00051900
  XF(NDROP(MM),LL)=0.0                       00052000
  X(NDROP(MM),LL)=0.0                        00052100
  NZO(NDROP(MM),LL)=0                       00052200
  NZF(NDROP(MM),LL)=0                       00052300
611  NZ(NDROP(MM),LL)=0                      00052400
610  CONTINUE                                00052500
      DO 477 LL=1,NCT                          00052600
477  ICENT(LL)=0                              00052700
C*****                                       00052800
C CALL THE SUBROUTINE GLOBE TO PERFORM THE GLOBAL TESTS 00052900
C FOR OPTIMAL PROFITS AND FEASIBLE CAPACITY 00053000
C*****                                       00053100
      CALL GLOBE                               00053200
      IF(IVECT1=0) 325,323,325                 00053300
323  WRITE(NO,103)                             00053400
103  FORMAT(1X,'FOR THE HEURISTICS EMPLOYED NO FEASIBLE SOLUTION CAN BE
*FOUND DUE TO CAPACITY LIMITATIONS'//)        00053600
999  STOP                                       00053700
5074 WRITE(NO,5075)                             00053800
5075 FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS - STOP'//*****
*****'//)                                    00053900
      GO TO 999                                00054100
325  IF(NORG=0) 327,328,327                    00054200
328  IVECT2=1                                  00054300
C*****                                       00054400
C CONSIDER THE DROP OF EACH PLANT              00054500
C*****                                       00054600
327  DO 520 KK=1,MPT                           00054700
      IF(IDROP(KK)-1) 520,530,520             00054800
530  NY(NDROP(KK))=0                           00054900
      FDROP(KK)=0.0                           00055000
      ADROP(KK)=0.0                           00055100
      CALL GLOBE                               00055200
      IF(IVECT1=0) 535,540,535                 00055300
540  IVECT1=1                                  00055400
      IDROP(KK)=0                              00055500
      NCON=NCON+1                             00055600
      IF(NCON=MPT) 560,435,435                 00055700
535  IF(IVECT2=0) 550,555,550                 00055800
555  IVECT2=1                                  00055900
      IF(TPORG-TPROF) 560,565,565             00056000
565  DIF=TPORG-TPROF                          00056100
      IF(DIF-F(NDROP(KK))) 560,560,575        00056200
575  IDROP(KK)=0                              00056300
      NCON=NCON+1                             00056400
      IF(NCON=MPT) 560,435,435                 00056500
550  IF(TPORG-TPROF) 585,696,696             00056600
696  IDN=1                                     00056700
      GO TO 693                                00056800

```

```

585     IF(TPFIN-TPROF) 590,693,693                                00056900
590     TFFIX=TFFIX                                              00057000
        TRFIN=TRFIN                                              00057100
        TPFIN=TPROF                                             00057200
        NN=KK                                                    00057300
        DO 592 MM=1,MPT                                         00057400
            NYF(NDROP(MM))=NY(NDROP(MM))                        00057500
            IF(NYF(NDROP(MM))-1) 783,784,783                    00057600
784     AF(NDROP(MM))=A(NDROP(MM))                                00057700
            FF(NDROP(MM))=F(NDROP(MM))                          00057800
            GO TO 997                                             00057900
783     AF(NDROP(MM))=O.O                                        00058000
            FF(NDROP(MM))=O.O                                    00058100
997     DO 593 LL=1,NCT                                         00058200
            XF(NDROP(MM),LL)=X(NDROP(MM),LL)                   00058300
593     NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)                     00058400
592     CONTINUE                                                00058500
        WRITE(NO,8889)                                           00058600
8889     FORMAT(/5X,'THE INTERMEDIATE FINAL SOLUTION HAS BEEN FOUND'//) 00058700
        WRITE(NO,2927)                                           00058800
2927     FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X, 00058900
            *'CAPACITY',/1X,'=====',8X,'=====',5X,'=====', 00059000
            *5X,'=====',//)                                     00059100
        DO 2928 MM=1,MPT                                         00059200
2928     WRITE(NO,2929) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)), 00059300
            *AF(NDROP(MM))                                       00059400
2929     FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)                00059500
        WRITE(NO,2931)                                           00059600
2931     FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X, 00059700
            *,'PROFIT',/1X,'=====',5X,'=====',8X,'===== 00059800
            *,12X,'=====',//)                                   00059900
        DO 2422 LL=1,NCT                                         00060000
        DO 2423 MM=1,MPT                                         00060100
            IF(NZF(NDROP(MM),LL)-1) 2423,2424,2423              00060200
2424     WRITE(NO,2425) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00060300
2423     CONTINUE                                                00060400
2422     CONTINUE                                                00060500
        WRITE(NO,3379) TPFIN                                      00060600
3379     FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)            00060700
2425     FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)              00060800
693     DO 694 MM=1,MPT                                         00060900
            IF(NY(NDROP(MM))-1) 694,699,694                    00061000
699     ADROP(MM)=A(NDROP(MM))                                    00061100
            DO 695 LL=1,NCT                                       00061200
                X(NDROP(MM),LL)=O.O                             00061300
695     NZ(NDROP(MM),LL)=O                                       00061400
694     CONTINUE                                                00061500
            IF(IDN-1) 560,565,560                                00061600
560     NY(NDROP(KK))=1                                           00061700
            FDROP(KK)=F(NDROP(KK))                               00061800
            ADROP(KK)=A(NDROP(KK))                              00061900
            IF(IDN-1) 520,562,520                                00062000
562     IDN=O                                                       00062100
520     CONTINUE                                                00062200
C*****                                                         00062300
C SET ORIGINAL PARAMETERS EQUAL TO THE FINAL PARAMETERS AND 00062400
C THE INTERMEDIATE PARAMETERS AS APPROPRIATE                    00062500
C*****                                                         00062600
        TOFIX=TFFIX                                              00062700
        TRORG=TRFIN                                              00062800
        TPORG=TPFIN                                              00062900
        DO 801 MM=1,MPT                                         00063000
            NYO(NDROP(MM))=NYF(NDROP(MM))                      00063100
            NY(NDROP(MM))=NYF(NDROP(MM))                      00063200
            IF(NN-MM) 803,802,803                                00063300
802     IDROP(MM)=O                                               00063400
            NCON=NCON+1                                          00063500
            IF(NCON-MPT) 803,435,435                             00063600
803     IF(NY(NDROP(MM))-1) 804,805,804                        00063700
805     ADROP(MM)=A(NDROP(MM))                                    00063800
            FDROP(MM)=F(NDROP(MM))                              00063900

```

```

      GO TO 806                                00064000
804   ADROP(MM)=0.0                            00064100
      FDROP(MM)=0.0                            00064200
806   DO 807 LL=1,NCT                          00064300
      XO(NDROP(MM),LL)=XF(NDROP(MM),LL)       00064400
807   NZO(NDROP(MM),LL)=NZF(NDROP(MM),LL)     00064500
801   CONTINUE                                  00064600
      DO 741 LL=1,NCT                          00064700
741   ICENT(LL)=0                              00064800
      GO TO 327                                  00064900
C*****                                        00065000
C PRINT ALL FINAL VALUES                      00065100
C*****                                        00065200
435   IF(NORG=0) 493,5074,493                 00065300
493   WRITE(NOT,967) TPFIN                     00065400
967   FORMAT(D20.10)                           00065500
      WRITE(ND,8883)                            00065600
8883  FORMAT(//5X,'THE FINAL SOLUTION HAS BEEN FOUND'//) 00065700
      WRITE(ND,2001)                            00065800
2001  FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'====',8X,'=====',5X,'=====',
* 5X,'=====',//)                             00066100
      DO 2002 MM=1,MPT                          00066200
2002  WRITE(ND,2003) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
* AF(NDROP(MM))                                00066300
2003  FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)  00066400
      WRITE(ND,2004)                            00066500
2004  FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
* ,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
* ,12X,'=====',//)                          00066800
      DO 2005 LL=1,NCT                          00066900
      DO 2006 MM=1,MPT                          00067000
      IF(NZF(NDROP(MM),LL)-1) 2006,2023,2006  00067100
2023  WRITE(ND,2007) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00067200
2006  CONTINUE                                  00067300
2005  CONTINUE                                  00067400
2007  FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//) 00067500
      WRITE(ND,3379) TPFIN                     00067600
      STOP                                       00067700
      END                                         00067800
C*****                                        00067900
C SUBROUTINE TO PERFORM GLOBAL TESTS FOR      00068000
C OPTIMAL PROFITS AND FEASIBLE CAPACITY      00068100
C*****                                        00068200
SUBROUTINE GLOBE                              00068300
  INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00068400
* NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00068500
* IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00068600
  REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00068700
* FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00068800
* SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00068900
* TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00069000
* SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00069100
* XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00069200
* A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00069300
  COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00069400
* TPFIN,TPROF
  COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00069500
* ,KK,JJ
  COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00069600
  COMMON/BLOCK4/XSTAR,XD,X,XF,PR,PROF 00069700
  COMMON/BLOCK5/NZO,NZ,NZF 00070000
  COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00070100
  TCAP=0.0 00070200
  TRGLBL=0.0 00070300
  DO 70 LL=1,NCT 00070400
  PRMAX=0.0 00070500
  CAPMIN=1.0D20 00070600
  DO 75 MM=1,MPT 00070700
  IF(NY(NDROP(MM))-1) 75,80,75 00070800
80   IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 75,85,85 00070900

```

```

85     IF(CAPMIN-XSTAR(NDROP(MM),LL)) 90,90,95          00071100
95     CAPMIN=XSTAR(NDROP(MM),LL)                    00071200
90     IF(PRMAX-PR(NDROP(MM),LL)) 100,75,75          00071300
100    PRMAX=PR(NDROP(MM),LL)                        00071400
75     CONTINUE                                       00071500
      TCAP=TCAP+CAPMIN                               00071600
      TRGLBL=TRGLBL+PRMAX                            00071700
70     CONTINUE                                       00071800
      SUMC=0.0                                         00071900
      TFIX=0.0                                         00072000
      DO 105 MM=1,MPT                                  00072100
        IF(NY(NDROP(MM))-1) 105,110,105              00072200
110    SUMC=SUMC+ADROP(MM)                            00072300
        TFIX=TFIX+FDROP(MM)                          00072400
105    CONTINUE                                       00072500
      TPGLBL=TRGLBL-TFIX                              00072600
      IF(SUMC-TCAP) 115,120,120                      00072700
115    IVECT1=0                                        00072800
      WRITE(NO,5888)                                   00072900
5888   FORMAT(5X,'THE GLOBAL TEST FOR FEASIBLE CAPACITY FAILED'//) 00073000
      GO TO 133                                       00073100
120    IF(TPORG-TPGLBL) 125,117,117                  00073200
117    IF(NORG=0) 118,119,118                       00073300
118    TPROF=TPGLBL                                   00073400
119    IVECT2=0                                        00073500
      WRITE(NO,5889)                                   00073600
5889   FORMAT(5X,'THE GLOBAL TEST FOR OPTIMAL PROFITS FAILED'//) 00073700
      GO TO 133                                       00073800
125   WRITE(NO,5890)                                   00073900
5890   FORMAT(5X,'PASSED BOTH GLOBAL TESTS FOR FEASIBLE CAPACITY AND OPTI 00074000
      *MAL PROFITS'//)                                00074100
C*****                                              00074200
C CALL THE SUBROUTINE ALOC TO PERFORM FEASIBLE ALLOCATIONS 00074300
C*****                                              00074400
      CALL ALOC                                       00074500
133   RETURN                                         00074600
      END                                             00074700
C*****                                              00074800
C SUBROUTINE TO PERFORM FEASIBLE ALLOCATIONS          00074900
C*****                                              00075000
      SUBROUTINE ALOC
      INTEGER RD,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
      *NY(40),NYD(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
      *IDROP(40),NDROP(40),NZD(40,50),NZ(40,50),NZF(40,50)
      REAL *8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
      *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM,
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
      *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
      *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
      *TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
      *,KK,JJ
      COMMON/BLOCK3/NY,NYD,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,XD,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
      NALOC=0
      NTEST=1
      TREV=0.0
      DO 135 JJ=1,MNTOT
        IF(ICENT(NCENT(JJ))-1) 7290,135,7290
7290   IF(NY(NPLANT(JJ))-1) 135,707,135
707   DO 708 IB=1,MPT
        IF(NPLANT(JJ)-NDROP(IB)) 708,140,708
708   CONTINUE
140   IF(ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ))) 135,150,150
150   X(NPLANT(JJ),NCENT(JJ))=XSTAR(NPLANT(JJ),NCENT(JJ))
      NZ(NPLANT(JJ),NCENT(JJ))=1

```

```

          ICENT(NCENT(JJ))=1                      00078200
          ADROP(IB)=ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ)) 00078300
          TREV=TREV+PROF(JJ)                      00078400
          NALOC=NALOC+1                          00078500
          IF(NALOC-NCT) 152,157,157              00078600
152      IF(AMOD(FLOAT(NALOC),FLOAT(NTEST))-O.O) 135,155,135 00078700
C*****
C CALL THE SUBROUTINE PARTL TO PERFORM PARTIAL TESTS 00078800
C FOR OPTIMAL PROFITS AND FEASIBLE CAPACITY        00078900
C*****
155      CALL PARTL                              00079000
          IF(IVECT1-O) 132,137,132              00079100
132      IF(IVECT2-O) 135,137,135              00079200
135      CONTINUE                              00079300
          IF(NALOC-NCT) 149,157,157            00079400
157      TPROF=TREV-TFIX                        00079500
          WRITE(NO,6237) TPROF                  00079600
6237     FORMAT(/5X,'THE TOTAL PROFIT FOR THIS CONFIGURATION IS= ', 00079700
          *F14.6//)                             00079800
          IF(NORG-O) 151,139,151              00079900
139      IF(TPORG-TPROF) 141,137,137          00080000
141      TOFIX=TFIX                             00080100
          TFFIX=TFIX                           00080200
          TRORG=TREV                           00080300
          TRFIN=TREV                           00080400
          TPORG=TPROF                          00080500
          TPFIN=TPROF                          00080600
          DO 142 MM=1,MPT                       00080700
              NYO(NDROP(MM))=NY(NDROP(MM))     00080800
              NYF(NDROP(MM))=NY(NDROP(MM))     00080900
              IF(NY(NDROP(MM))-1) 142,146,142  00081000
146      ADROP(MM)=A(NDROP(MM))                00081100
              AF(NDROP(MM))=A(NDROP(MM))       00081200
              FF(NDROP(MM))=F(NDROP(MM))       00081300
          DO 143 LL=1,NCT                       00081400
              XO(NDROP(MM),LL)=X(NDROP(MM),LL) 00081500
              XF(NDROP(MM),LL)=X(NDROP(MM),LL) 00081600
              NZO(NDROP(MM),LL)=NZ(NDROP(MM),LL) 00081700
              NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL) 00081800
              X(NDROP(MM),LL)=O.O              00081900
143      NZ(NDROP(MM),LL)=O                   00082000
142      CONTINUE                              00082100
          NORG=1                                00082200
          WRITE(NO,5893)                        00082300
5893     FORMAT(5X,'THE ORIGINAL FEASIBLE SOLUTION HAS BEEN FOUND'//) 00082400
4232     WRITE(NO,3022)                         00082500
3022     FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X, 00082600
          *'CAPACITY',/1X,'=====',8X,'=====',5X,'=====', 00082700
          *5X,'=====',//)                     00082800
          DO 3023 MM=1,MPT                      00082900
3023     WRITE(NO,3024) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)), 00083000
          *AF(NDROP(MM))                       00083100
3024     FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//) 00083200
          WRITE(NO,3025)                        00083300
3025     FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X 00083400
          *,'PROFIT',/1X,'=====',5X,'=====',8X,'===== 00083500
          *,12X,'=====',//)                  00083600
          DO 3026 LL=1,NCT                      00083700
          DO 3027 MM=1,MPT                      00083800
              IF(NZF(NDROP(MM),LL)-1) 3027,3051,3027 00083900
3051     WRITE(NO,3028) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00084000
3027     CONTINUE                              00084100
3026     CONTINUE                              00084200
3028     FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//) 00084300
          WRITE(NO,3380) TPFIN                  00084400
3380     FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//) 00084500
          GO TO 151                             00084600
149      IVECT1=O                              00084700
137      DO 153 MM=1,MPT                       00084800
          IF(NY(NDROP(MM))-1) 153,710,153     00084900
710      ADROP(MM)=A(NDROP(MM))                00085000

```

```

DO 154 LL=1,NCT                                00085300
  X(NDROP(MM),LL)=0.0                          00085400
154 NZ(NDROP(MM),LL)=0                        00085500
153 CONTINUE                                   00085600
151 DO 738 LL=1,NCT                            00085700
738 ICENT(LL)=0                                00085800
  RETURN                                        00085900
  END                                           00086000
C*****                                        00086100
C SUBROUTINE TO PERFORM PARTIAL TESTS FOR      00086200
C OPTIMAL PROFITS AND FEASIBLE CAPACITY       00086300
C*****                                        00086400
  SUBROUTINE PARTL                              00086500
    INTEGER RO,MA,FI,EO,SLSH,ZERO,PRD,FT,EQU,INE,Z,Y,POS, 00086600
    *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00086700
    *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00086800
    REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00086900
    *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00087000
    *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00087100
    *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00087200
    *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00087300
    *XD(40,50),X(40,50),XF(40,50),PR(40,50),PRDF(2000), 00087400
    *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00087500
    COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00087600
    *TPFIN,TPROF 00087700
    COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00087800
    *,KK,JJ 00087900
    COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00088000
    COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00088100
    COMMON/BLOCK5/NZO,NZ,NZF 00088200
    COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00088300
    RCAP=0.0 00088400
    TRPTL=0.0 00088500
    DO 160 LL=1,NCT 00088600
      PRMAX=0.0 00088700
      CAPMIN=1.0D20 00088800
      IF(ICENT(LL)-1) 736,160,736 00088900
736 DO 165 MM=1,MPT 00089000
      IF(NY(NDROP(MM))-1) 165,175,165 00089100
175 IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 165,180,180 00089200
180 IF(CAPMIN-XSTAR(NDROP(MM),LL)) 185,185,190 00089300
190 CAPMIN=XSTAR(NDROP(MM),LL) 00089400
185 IF(PRMAX-PR(NDROP(MM),LL)) 195,165,165 00089500
195 PRMAX=PR(NDROP(MM),LL) 00089600
165 CONTINUE 00089700
      RCAP=RCAP+CAPMIN 00089800
      TRPTL=TRPTL+PRMAX 00089900
160 CONTINUE 00090000
      TRPTL=TREV+TRPTL 00090100
      SUMR=0.0 00090200
      DO 200 MM=1,MPT 00090300
        IF(NY(NDROP(MM))-1) 200,205,200 00090400
205 SUMR=SUMR+ADROP(MM) 00090500
200 CONTINUE 00090600
        TPPTL=TRPTL-TFIX 00090700
        IF(SUMR-RCAP) 210,215,215 00090800
210 IVECT1=0 00090900
        WRITE(NO,5891) 00091000
5891 FORMAT(5X,'THE PARTIAL TEST FOR FEASIBLE CAPACITY FAILED'///) 00091100
        GO TO 220 00091200
215 IF(TPORG-TPPTL) 220,225,225 00091300
225 IF(NORG=0) 213,221,213 00091400
213 TPROF=TPPTL 00091500
221 IVECT2=0 00091600
        WRITE(NO,5892) 00091700
5892 FORMAT(5X,'THE PARTIAL TEST FOR OPTIMAL PROFITS FAILED'///) 00091800
        GO TO 220 00091900
220 RETURN 00092000
      END 00092100
C*****                                        00092200
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES 00092300

```



```

C*****0000100
C      0000200
C      THIS PROGRAM PERFORMS THE HEURISTIC ALGORITHM FOR PLANT      0000300
C      LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE             0000400
C      STOCHASTIC DEMAND                                           0000500
C      0000600
C      ALSO IT SETS UP THE INPUT DATA DECK FOR EXTREME POINT     0000700
C      RANKING TECHNIQUE                                           0000800
C      0000900
C      THE PLANTS ARE ASSUMED CAPACITATED                          0001000
C      AND A TYPE 2 NORMALLY DISTRIBUTED DEMAND IS ASSUMED        0001100
C      0001200
C      THIS ALGORITHM TAKES INTO ACCOUNT OF THE FACT THAT         0001300
C      ALL DEMAND CENTERS NEED TO BE ALLOCATED AND THAT           0001400
C      EACH DEMAND CENTER MUST RECEIVE THE SUPPLY                 0001500
C      FROM AT MOST ONE PLANT                                     0001600
C      0001700
C      WRITTEN BY LOGENDRAN RASARATNAM                             0001800
C      SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT            0001900
C      OKLAHOMA STATE UNIVERSITY                                  0002000
C      0002100
C      DISSERTATION ADVISER: DR. M. PALMER TERRELL               0002200
C      0002300
C      VERSION 1 -- AUGUST, 1984                                   0002400
C*****0002500
C      0002600
C*****0002700
C      0002800
C*** GENERAL STRUCTURE AND INPUT REQUIREMENTS:                   0002900
C (MAIN PROGRAM DRIVES THE SUBROUTINES UNFRM AND GLOBE)          0003000
C (UNFRM DRIVES SUBROUTINE RANDU)                                0003100
C (GLOBE DRIVES SUBROUTINE ALOC)                                  0003200
C (ALOC DRIVES SUBROUTINE PARTL)                                 0003300
C      0003400
C*****0003500
C      0003600
C      SUBROUTINE          FUNCTION                                0003700
C      -----          -----                                0003800
C      RANDU              TO GENERATE A UNIFORMLY DISTRIBUTED     0003900
C                        RANDOM NUMBER BETWEEN 0 AND 1            0004000
C      UNFRM              TO CONVERT THE RANDOM NUMBER GENERATED  0004100
C                        BY RANDU BETWEEN APPROPRIATE LIMITS     0004200
C                        ESTABLISHED FOR THE UNIFORM             0004300
C                        DISTRIBUTION TO BE USED                 0004400
C      GLOBE              PERFORMS THE GLOBAL TESTS FOR OPTIMAL   0004500
C                        PROFITS AND FEASIBLE CAPACITY          0004600
C      ALOC              PERFORMS THE FEASIBLE ALLOCATIONS FOR   0004700
C                        FOR EACH OF THE DEMAND CENTERS         0004800
C                        DICTATED BY PRIORITY RULE 1            0004900
C      PARTL             PERFORMS THE PARTIAL TESTS FOR OPTIMAL  0005000
C                        PROFITS AND FEASIBLE CAPACITY          0005100
C      0005200
C*****0005300
C      0005400
C*** EXTERNAL FUNCTIONS REQUIRED                                   0005500
C (1) REGULAR SYSTEM SUPPLIED FORTRAN FUNCTIONS                 0005600
C (2) TWO IMSL SUBROUTINES                                       0005700
C     MDNOR -- CUMULATIVE PROBABILITY FUNCTION OF                0005800
C             STANDARD NORMAL                                    0005900
C     MDNRIS -- INVERSE FUNCTION OF MDNOR                        0006000
C     0006100
C*****0006200
C      0006300
C*** COMMON BLOCK VARIABLE DEFINITIONS                          0006400
C ONLY THOSE VARIABLES THAT REQUIRE EXPLANATION ARE LISTED     0006500
C      0006600
C /BLOCK1/                                                       0006700
C      0006800
C     TOFIX - TOTAL FIXED COST FOR THE ORIGINAL LOCATION VECTOR  0006900
C             UNDER CONSIDERATION                               0007000
C     TFIX  - TOTAL FIXED COST FOR THE LOCATION VECTOR PRESENTLY 0007100

```

C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007200
C		FROM THE ORIGINAL LOCATION VECTOR	00007300
C	TFFIX	- TOTAL FIXED COST FOR THE FINAL LOCATION VECTOR	00007400
C		CURRENTLY STORED	00007500
C	TRORG	- TOTAL REVENUE FOR THE ORIGINAL LOCATION VECTOR	00007600
C		UNDER CONSIDERATION	00007700
C	TREV	- TOTAL REVENUE FOR THE LOCATION VECTOR PRESENTLY	00007800
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007900
C		FROM THE ORIGINAL LOCATION VECTOR	00008000
C	TRFIN	- TOTAL REVENUE FOR THE FINAL LOCATION VECTOR	00008100
C		CURRENTLY STORED	00008200
C	TPORG	- TOTAL PROFIT FOR THE ORIGINAL LOCATION VECTOR	00008300
C		UNDER CONSIDERATION	00008400
C	TPROF	- TOTAL PROFIT FOR THE LOCATION VECTOR PRESENTLY	00008500
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008600
C		FROM THE ORIGINAL LOCATION VECTOR	00008700
C	TPFIN	- TOTAL PROFIT FOR THE FINAL LOCATION VECTOR	00008800
C		CURRENTLY STORED	00008900
C			00009000
C	/BLOCK2/		00009100
C			00009200
C	IVECT1-	INDICATOR VARIABLE FOR THE TEST ON FEASIBLE CAPACITY	00009300
C	IVECT2-	INDICATOR VARIABLE FOR THE TEST ON OPTIMAL PROFITS	00009400
C	NORG	- VARIABLE INDICATING THAT THE ORIGINAL FEASIBLE	00009500
C		SOLUTION HAS BEEN FOUND	00009600
C	NCON	- TOTAL NUMBER OF PLANTS DROPPED SO FAR	00009700
C	NALOC	- TOTAL NUMBER OF DEMAND CENTERS ALLOCATED SO FAR	00009800
C	MPT	- TOTAL NUMBER OF PLANTS FOR THE PROBLEM SOLVED	00009900
C	NCT	- TOTAL NUMBER OF DEMAND CENTERS FOR THE PROBLEM SOLVED	00010000
C			00010100
C	/BLOCK3/		00010200
C			00010300
C	NYO	- THE ORIGINAL LOCATION VECTOR UNDER CONSIDERATION	00010400
C	NY	- THE LOCATION VECTOR PRESENTLY CONSIDERED BY	00010500
C		DROPPING ONE PLANT AT A TIME FROM THE	00010600
C		ORIGINAL LOCATION VECTOR	00010700
C	NYF	- THE FINAL LOCATION VECTOR CURRENTLY STORED	00010800
C	NPLANT-	VARIABLE USED FOR THE PLANTS IN RANKING THE OPTIMAL	00010900
C		PROFITS EMPLOYING PRIORITY RULE 1	00011000
C	NCENT	- VARIABLE USED FOR THE DEMAND CENTERS IN RANKING THE	00011100
C		OPTIMAL PROFITS EMPLOYING PRIORITY RULE 1	00011200
C	ICENT	- VARIABLE INDICATING WHETHER A DEMAND CENTER HAS	00011300
C		ALREADY BEEN ALLOCATED	00011400
C	IDROP	- VARIABLE INDICATING THE PLANT PRESENTLY DROPPED	00011500
C		FROM THE ORIGINAL LOCATION VECTOR	00011600
C	NDROP	- VARIABLE USED FOR RANKING THE PLANTS FOR DROP	00011700
C		EMPLOYING PRIORITY RULE 2	00011800
C			00011900
C	/BLOCK4/		00012000
C			00012100
C	XSTAR	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR EACH	00012200
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00012300
C	XO	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE ORIGINAL	00012400
C		LOCATION VECTOR UNDER CONSIDERATION	00012500
C	X	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE LOCATION	00012600
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00012700
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00012800
C	XF	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE FINAL	00012900
C		LOCATION VECTOR CURRENTLY STORED	00013000
C	PR	- OPTIMAL PROFITS DETERMINED FOR EACH COMBINATION OF	00013100
C		DEMAND CENTERS AND PROFITS	00013200
C	PROF	- THE RANKED OPTIMAL PROFITS DETERMINED FOR EACH	00013300
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00013400
C		USING PRIORITY RULE 1	00013500
C			00013600
C	/BLOCK5/		00013700
C			00013800
C	NZO	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE ORIGINAL	00013900
C		LOCATION VECTOR UNDER CONSIDERATION	00014000
C	NZ	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE LOCATION	00014100
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00014200

```

C          AT A TIME FROM THE ORIGINAL LOCATION VECTOR          00014300
C      NZF - COMPONENTS OF THE ALLOCATION MATRIX FOR THE FINAL    00014400
C          LOCATION VECTOR CURRENTLY STORED                      00014500
C                                                                00014600
C /BLOCK6/                                                      00014700
C                                                                00014800
C      A - CAPACITY OF EACH PLANT                               00014900
C      F - FIXED COST OF EACH PLANT                             00015000
C      ADROP- THE REMAINING CAPACITY OF EACH PLANT AFTER HAVING 00015100
C          ALLOCATED A SUBSET OF DEMAND CENTERS                 00015200
C                                                                00015300
C*****00015400
C                                                                00015500
C*** OTHER VARIABLE DEFINITIONS                                00015600
C                                                                00015700
C      MU - MEAN OF THE NORMAL DISTRIBUTION OF DEMAND          00015800
C          DETERMINED FOR EACH COMBINATION OF                   00015900
C          DEMAND CENTERS AND PLANTS                            00016000
C      SIGMA- STANDARD DEVIATION OF THE NORMAL DISTRIBUTION OF 00016100
C          DEMAND DETERMINED FOR EACH COMBINATION OF           00016200
C          DEMAND CENTERS AND PLANTS                            00016300
C      R - PER UNIT VARIABLE COST OF THE PRODUCT SUPPLIED FOR 00016400
C          EACH COMBINATION OF DEMAND CENTERS AND PLANTS       00016500
C      PC - PER UNIT PRICE OF THE PRODUCT RECEIVED FOR EACH    00016600
C          COMBINATION OF DEMAND CENTERS AND PLANTS            00016700
C                                                                00016800
C*****00016900
C      INTEGER RD,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,        00017000
C      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00017100
C      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)    00017200
C      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00017300
C      *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00017400
C      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,   00017500
C      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00017600
C      *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00017700
C      *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),    00017800
C      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40) 00017900
C      DATA RD,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS/2HRD,2HMA, 00018000
C      *2HFI,2HEO,1HO,3HPRD,2HFT,3HEQU,3HINE,1HZ,1HY,1H+/    00018100
C      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,   00018200
C      *TPFIN,TPROF                                             00018300
C      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00018400
C      *,KK,JJ                                                 00018500
C      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00018600
C      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF                    00018700
C      COMMON/BLOCK5/NZO,NZ,NZF                               00018800
C      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS           00018900
C      COMMON/BLOCK7/B,C,D                                    00019000
C      COMMON/BLOCK8/IX,IY,YFL                                00019100
C*****00019200
C      NO=12                                                    00019300
C      NOT=14                                                  00019400
C      MPT=3                                                    00019500
C      NCT=4                                                    00019600
C      CONST1=1.0                                              00019700
C      CONST2=0.0                                              00019800
C      MNTOT=MPT*NCT                                          00019900
C      MQ=NCT+MPT                                             00020000
C      MR=MPT*NCT+MPT                                         00020100
C*****00020200
C      GENERATE VALUES RANDOMLY FOR F(MM)                    00020300
C      SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM 00020400
C      DISTRIBUTION                                           00020500
C*****00020600
C      B=200.                                                  00020700
C      C=300.                                                  00020800
C      IX=13                                                   00020900
C      WRITE(NO,4025)                                          00021000
C      4025 FORMAT(//5X,'THE PLANT FIXED COSTS ARE',//)       00021100
C      WRITE(NO,5248)                                          00021200
C      5248 FORMAT(5X,'FIXED COST',5X,'PLANT'//)              00021300

```

```

DO 1001 MM=1,MPT
CALL UNFRM
F(MM)=D
WRITE(NO,5249) D,MM
5249 FORMAT(1X,F14.6,7X,I2//)
IX=IY
1001 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR A(MM)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=300.
      C=350.
      WRITE(NO,4026)
4026 FORMAT(//5X,'THE PLANT CAPACITIES ARE',//)
      WRITE(NO,5250)
5250 FORMAT(5X,'CAPACITY',5X,'PLANT'//)
DO 1002 MM=1,MPT
CALL UNFRM
A(MM)=D
WRITE(NO,5251) D,MM
5251 FORMAT(1X,F14.6,5X,I2//)
IX=IY
1002 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR R(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=0.5
      C=2.5
      WRITE(NO,4027)
4027 FORMAT(//5X,'THE PER UNIT VARIABLE COSTS ARE'//)
      WRITE(NO,5252)
5252 FORMAT(1X,'PER UNIT VARIABLE COST',5X,'PLANT',5X,'DEMAND CENTER',
*/)
DO 1003 MM=1,MPT
DO 1004 LL=1,NCT
CALL UNFRM
R(MM,LL)=D
WRITE(NO,5253) D,MM,LL
5253 FORMAT(5X,F14.6,9X,I2,13X,I2//)
1004 IX=IY
1003 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR PC(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B1=3.
      C1=6.
      B2=7.
      C2=13.
      WRITE(NO,4028)
4028 FORMAT(//1X,'THE PER UNIT PRICE AND THE MEAN AND STANDARD DEVIATIO
*N OF THE DEMAND FOR THE PRODUCTS ARE'//)
      WRITE(NO,5254)
5254 FORMAT(4X,'PER UNIT PRICE',6X,'MEAN DEMAND',5X,'STANDARD DEVIATION
*,5X,'PLANT',5X,'DEMAND CENTER'//)
DO 1005 MM=1,MPT
DO 1006 LL=1,NCT
B=B1
C=C1
CALL UNFRM
PC(MM,LL)=D
MU(MM,LL)=600./PC(MM,LL)
IX=IY
B=B2
C=C2
CALL UNFRM

```

```

00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00023000
00023100
00023200
00023300
00023400
00023500
00023600
00023700
00023800
00023900
00024000
00024100
00024200
00024300
00024400
00024500
00024600
00024700
00024800
00024900
00025000
00025100
00025200
00025300
00025400
00025500
00025600
00025700
00025800
00025900
00026000
00026100
00026200
00026300
00026400
00026500
00026600
00026700
00026800
00026900
00027000
00027100
00027200
00027300
00027400
00027500
00027600
00027700
00027800
00027900
00028000
00028100
00028200
00028300
00028400

```

```

        SIGMA(MM,LL)=D                                00028500
        WRITE(NO,5255) D,MU(MM,LL),SIGMA(MM,LL),MM,LL 00028600
5255. FORMAT(1X,F14.6,5X,F14.6,5X,F14.6,11X,I2,11X,I2//) 00028700
1006 IX=IY                                           00028800
1005 CONTINUE                                         00028900
C*****                                              00029000
C COMPUTE THE VALUES OF THE OPTIMAL SUPPLY QUANTITIES XSTAR(MM,LL) 00029100
C AND THE OPTIMAL PROFITS PR(MM,LL)                 00029200
C*****                                              00029300
        DO 900 MM=1,MPT                                00029400
        DO 901 LL=1,NCT                                00029500
            P=1.-R(MM,LL)/PC(MM,LL)                   00029600
            CALL MDNRIS(P,S,IER)                       00029700
            XSTAR(MM,LL)=S*SIGMA(MM,LL)+MU(MM,LL)     00029800
            IF(XSTAR(MM,LL)-O.) 902,902,903           00029900
903    C1=PC(MM,LL)*MU(MM,LL)                          00030000
        TWOP1=2.*3.141593                             00030100
        FAC1=SQRT(TWOP1)                              00030200
        FAC2=1./FAC1                                  00030300
        FAC3=XSTAR(MM,LL)-MU(MM,LL)                   00030400
        FAC4=FAC3/SIGMA(MM,LL)                       00030500
        FAC5=-FAC4**2)/2.                             00030600
        FAC6=EXP(FAC5)                                00030700
        FAC7=-PC(MM,LL)*SIGMA(MM,LL)                 00030800
        C2=FAC7*FAC2*FAC6                             00030900
        S=FAC4                                         00031000
        CALL MDNOR(S,P)                                00031100
        FAC8=1.-P                                     00031200
        C3=PC(MM,LL)*FAC3*FAC8                        00031300
        C4=-R(MM,LL)*XSTAR(MM,LL)                    00031400
        PR(MM,LL)=C1+C2+C3+C4                         00031500
        IF(PR(MM,LL)-O.) 902,902,901                 00031600
902    XSTAR(MM,LL)=O.                                00031700
        PR(MM,LL)=O.                                  00031800
901    CONTINUE                                       00031900
900    CONTINUE                                       00032000
        WRITE(NO,4029)                                00032100
4029  FORMAT(//5X,'THE OPTIMAL SUPPLY QUANTITIES AND THE PROFITS ARE',// 00032200
        *)                                           00032300
        WRITE(NO,5256)                                00032400
5256  FORMAT(1X,'OPTIMAL SUPPLY QUANTITY',8X,'PROFIT',8X,'PLANT',5X, 00032500
        *'DEMAND CENTER'//)                          00032600
        DO 3003 MM=1,MPT                               00032700
        DO 3004 LL=1,NCT                               00032800
3004  WRITE(NO,5257) XSTAR(MM,LL),PR(MM,LL),MM,LL 00032900
5257  FORMAT(4X,F14.6,9X,F14.6,6X,I2,13X,I2//)      00033000
3003  CONTINUE                                       00033100
C*****                                              00033200
C GENERATE THE INPUT DATA DECK FOR THE              00033300
C EXTREME POINT RANKING TECHNIQUE                   00033400
C*****                                              00033500
        WRITE(NOT,137) MQ,MR                          00033600
137   FORMAT(2I5)                                    00033700
        WRITE(NOT,138) RO                              00033800
138   FORMAT(1X,A2)                                   00033900
        WRITE(NOT,139) ZERO,PRO,FT                   00034000
139   FORMAT(11X,A1,1X,A3,A2)                        00034100
        DO 141 LL=1,NCT                                00034200
            IF(LL-9) 142,142,143                      00034300
142   WRITE(NOT,144) ZERO,EQU,ZERO,LL                00034400
144   FORMAT(11X,A1,1X,A3,A1,I1)                     00034500
            GO TO 141                                  00034600
143   WRITE(NOT,145) ZERO,EQU,LL                     00034700
145   FORMAT(11X,A1,1X,A3,I2)                         00034800
141   CONTINUE                                       00034900
        DO 146 MM=1,MPT                                00035000
            IF(MM-9) 147,147,148                      00035100
147   WRITE(NOT,144) POS,INE,ZERO,MM                 00035200
            GO TO 146                                  00035300
148   WRITE(NOT,145) POS,INE,MM                      00035400
146   CONTINUE                                       00035500

```

```

WRITE(NOT,149) MA
149 FORMAT(1X,A2)
DO 150 LL=1,NCT
DO 151 MM=1,MPT
    IF(MM-9) 152,152,153
152 IF(LL-9) 154,154,155
154 WRITE(NOT,733) Z,ZERO,MM,ZERO,LL,PRO,FT,PR(MM,LL)
733 FORMAT(7X,A1,A1,I1,A1,I1,1X,A3,A2,1X,D20.10)
WRITE(NOT,734) Z,ZERO,MM,ZERO,LL,EQU,ZERO,LL,CONST1
734 FORMAT(7X,A1,A1,I1,A1,I1,1X,A3,A1,I1,1X,D20.10)
WRITE(NOT,734) Z,ZERO,MM,ZERO,LL,INE,ZERO,MM,XSTAR(MM,LL)
GO TO 151
155 WRITE(NOT,736) Z,ZERO,MM,LL,PRO,FT,PR(MM,LL)
736 FORMAT(7X,A1,A1,I1,I2,1X,A3,A2,1X,D20.10)
WRITE(NOT,737) Z,ZERO,MM,LL,EQU,LL,CONST1
737 FORMAT(7X,A1,A1,I1,I2,1X,A3,I2,1X,D20.10)
WRITE(NOT,738) Z,ZERO,MM,LL,INE,ZERO,MM,XSTAR(MM,LL)
738 FORMAT(7X,A1,A1,I1,I2,1X,A3,A1,I1,1X,D20.10)
GO TO 151
153 IF(LL-9) 157,157,158
157 WRITE(NOT,739) Z,MM,ZERO,LL,PRO,FT,PR(MM,LL)
739 FORMAT(7X,A1,I2,A1,I1,1X,A3,A2,1X,D20.10)
WRITE(NOT,740) Z,MM,ZERO,LL,EQU,ZERO,LL,CONST1
740 FORMAT(7X,A1,I2,A1,I1,1X,A3,A1,I1,1X,D20.10)
WRITE(NOT,756) Z,MM,ZERO,LL,INE,MM,XSTAR(MM,LL)
756 FORMAT(7X,A1,I2,A1,I1,1X,A3,I2,1X,D20.10)
GO TO 151
158 WRITE(NOT,742) Z,MM,LL,PRO,FT,PR(MM,LL)
742 FORMAT(7X,A1,I2,I2,1X,A3,A2,1X,D20.10)
WRITE(NOT,743) Z,MM,LL,EQU,LL,CONST1
743 FORMAT(7X,A1,I2,I2,1X,A3,I2,1X,D20.10)
WRITE(NOT,743) Z,MM,LL,INE,MM,XSTAR(MM,LL)
151 CONTINUE
150 CONTINUE
DO 159 MM=1,MPT
    FMS(MM)=-F(MM)
    AMS(MM)=-A(MM)
    IF(MM-9) 160,160,161
160 WRITE(NOT,745) Y,ZERO,ZERO,ZERO,MM,PRO,FT,FMS(MM)
WRITE(NOT,746) Y,ZERO,ZERO,ZERO,MM,INE,ZERO,MM,AMS(MM)
GO TO 159
161 WRITE(NOT,747) Y,ZERO,ZERO,MM,PRO,FT,FMS(MM)
WRITE(NOT,748) Y,ZERO,ZERO,MM,INE,MM,AMS(MM)
745 FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A2,1X,D20.10)
746 FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A1,I1,1X,D20.10)
747 FORMAT(7X,A1,A1,A1,I2,1X,A3,A2,1X,D20.10)
748 FORMAT(7X,A1,A1,A1,I2,1X,A3,I2,1X,D20.10)
159 CONTINUE
WRITE(NOT,749) FI
749 FORMAT(1X,A2)
DO 162 LL=1,NCT
    IF(LL-9) 163,163,164
163 WRITE(NOT,755) EQU,ZERO,LL,CONST1
GO TO 162
164 WRITE(NOT,750) EQU,LL,CONST1
755 FORMAT(13X,A3,A1,I1,1X,D20.10)
750 FORMAT(13X,A3,I2,1X,D20.10)
162 CONTINUE
DO 165 MM=1,MPT
    IF(MM-9) 166,166,167
166 WRITE(NOT,751) INE,ZERO,MM,CONST2
GO TO 165
167 WRITE(NOT,752) INE,MM,CONST2
751 FORMAT(13X,A3,A1,I1,1X,D20.10)
752 FORMAT(13X,A3,I2,1X,D20.10)
165 CONTINUE
WRITE(NOT,753) EO
753 FORMAT(1X,A2)
C *****
C SORT THE PROFITS IN A DECENDING ORDER
C *****

```

```

      K=O
      KONST=O
      DO 7 I=1,MPT
      DO 10 J=1,NCT
      K=KONST+J
      PROF(K)=PR(I,J)
      NPLANT(K)=I
      NCENT(K)=J
10  CONTINUE
      KONST=I*NCT
      7  CONTINUE
      MN=1
15  PROFM=PROF(MN)
      NPLAM=NPLANT(MN)
      NCENTM=NCENT(MN)
      DO 20 I=MN,MNTOT
      IF(PROFM-PROF(I)) 25,25,20
25  PROFM=PROF(I)
      NPLAM=NPLANT(I)
      NCENTM=NCENT(I)
      IMAX=I
20  CONTINUE
      PROF(IMAX)=PROF(MN)
      NPLANT(IMAX)=NPLANT(MN)
      NCENT(IMAX)=NCENT(MN)
      PROF(MN)=PROFM
      NPLANT(MN)=NPLAM
      NCENT(MN)=NCENTM
      MN=MN+1
      IF(MN-MNTOT) 15,15,35
35  WRITE(NO,4030)
4030 FORMAT(5X,'THE SORTED PROFITS AND THE RESPECTIVE PLANTS AND THE DEO
      *MAND CENTERS ARE'//)
      WRITE(NO,4034)
4034 FORMAT(5X,'RANKED PROFIT',10X,'PLANT',10X,'DEMAND CENTER'//)
      DO 5033 I=1,MNTOT
5033 WRITE(NO,4035) PROF(I),NPLANT(I),NCENT(I)
4035 FORMAT(3X,F14.6,12X,I2,17X,I2//)
C*****
C SORT THE PLANTS IN A DECENDING ORDER OF FIXED COSTS
C*****
      DO 333 I=1,MPT
      FDROP(I)=F(I)
      ADROP(I)=A(I)
333  NDROP(I)=I
      MJ=1
37  FDROPM=FDROP(MJ)
      ADROPM=ADROP(MJ)
      NDROPM=NDROP(MJ)
      DO 40 I=MJ,MPT
      IF(FDROPM-FDROP(I)) 45,45,40
45  FDROPM=FDROP(I)
      ADROPM=ADROP(I)
      NDROPM=NDROP(I)
      IMAX=I
40  CONTINUE
      FDROP(IMAX)=FDROP(MJ)
      ADROP(IMAX)=ADROP(MJ)
      NDROP(IMAX)=NDROP(MJ)
      FDROP(MJ)=FDROPM
      ADROP(MJ)=ADROPM
      NDROP(MJ)=NDROPM
      IDROP(MJ)=1
      MJ=MJ+1
      IF(MJ-MPT) 37,37,50
50  WRITE(NO,4031)
4031 FORMAT(5X,'THE SORTED FIXED COSTS AND OTHER RESPECTIVE PARAMETERS
      *ARE',//)
      WRITE(NO,4032)
4032 FORMAT(1X,'RANKED PLANTS FOR DROP',10X,'FIXED COST',11X,'CAPACITY'
      *,11X,'FIXED COST',10X,'CAPACITY',12X,'IDROP VALUES'//)

```

```

DO 5034 I=1,MPT
5034 WRITE(NO,4033) NDROP(I),FDROP(I),ADROP(I),F(NDROP(I)),
      *A(NDROP(I)),IDROP(I)
4033 FORMAT(10X,I2,17X,F14.6,6X,F14.6,6X,F14.6,6X,F14.6,10X,I7//)
C*****
C START WITH ALL PLANTS OPEN
C SET ALL NY(I) EQUAL TO 1
C*****
DO 55 K=1,MPT
55 NY(NDROP(K))=1
C*****
C INITIALIZE ALL PARAMETERS
C*****
      TOFIX=0.0
      TFFIX=0.0
      TRORG=0.0
      TRFIN=0.0
      TPRG=0.0
      TPFIN=0.0
      IVECT1=1
      IVECT2=1
      NORG=0
      NCON=0
DO 610 MM=1,MPT
      NYO(NDROP(MM))=0
      NYF(NDROP(MM))=0
      AF(NDROP(MM))=0.0
      FF(NDROP(MM))=0.0
DO 611 LL=1,NCT
      XD(NDROP(MM),LL)=0.0
      XF(NDROP(MM),LL)=0.0
      X(NDROP(MM),LL)=0.0
      NZD(NDROP(MM),LL)=0
      NZF(NDROP(MM),LL)=0
611 NZ(NDROP(MM),LL)=0
610 CONTINUE
DO 477 LL=1,NCT
477 ICENT(LL)=0
C*****
C CALL THE SUBROUTINE GLOBE TO PERFORM THE GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
      CALL GLOBE
      IF(IVECT1=0) 325,323,325
323 WRITE(NO,103)
103 FORMAT(1X,'FOR THE HEURISTICS EMPLOYED NO FEASIBLE SOLUTION CAN BE
      *FOUND DUE TO CAPACITY LIMITATIONS'//)
999 STOP
5074 WRITE(NO,5075)
5075 FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS - STOP'/'*****
      *****'//)
      GO TO 999
325 IF(NORG=0) 327,328,327
328 IVECT2=1
C*****
C CONSIDER THE DROP OF EACH FACILITY
C*****
327 DO 520 KK=1,MPT
      IF(IDROP(KK)=1) 520,530,520
530 NY(NDROP(KK))=0
      FDROP(KK)=0.0
      ADROP(KK)=0.0
      CALL GLOBE
      IF(IVECT1=0) 535,540,535
540 IVECT1=1
      IDROP(KK)=0
      NCON=NCON+1
      IF(NCON=MPT) 560,435,435
535 IF(IVECT2=0) 550,555,550
555 IVECT2=1
      IF(TPRG=TPROF) 560,565,565

```

```

00049800
00049900
00050000
00050100
00050200
00050300
00050400
00050500
00050600
00050700
00050800
00050900
00051000
00051100
00051200
00051300
00051400
00051500
00051600
00051700
00051800
00051900
00052000
00052100
00052200
00052300
00052400
00052500
00052600
00052700
00052800
00052900
00053000
00053100
00053200
00053300
00053400
00053500
00053600
00053700
00053800
00053900
00054000
00054100
00054200
00054300
00054400
00054500
00054600
00054700
00054800
00054900
00055000
00055100
00055200
00055300
00055400
00055500
00055600
00055700
00055800
00055900
00056000
00056100
00056200
00056300
00056400
00056500
00056600
00056700
00056800

```



```

565   DIF=TPORG-TPROF                                00056900
      IF(DIF-F(NDROP(KK))) 560,560.575              00057000
575   IDROP(KK)=O                                    00057100
      NCON=NCON+1                                    00057200
      IF(NCON-MPT) 560,435,435                       00057300
550   IF(TPORG-TPROF) 585,696,696                   00057400
696   IDN=1                                           00057500
      GO TO 693                                       00057600
585   IF(TPFIN-TPROF) 590,693,693                   00057700
590   TFFIX=TFFIX                                    00057800
      TRFIN=TREV                                       00057900
      TPFIN=TPROF                                      00058000
      NN=KK                                           00058100
      DO 592 MM=1,MPT                                  00058200
          NYF(NDROP(MM))=NY(NDROP(MM))               00058300
          IF(NYF(NDROP(MM))-1) 783,784,783           00058400
784   AF(NDROP(MM))=A(NDROP(MM))                    00058500
          FF(NDROP(MM))=F(NDROP(MM))                 00058600
          GO TO 997                                    00058700
783   AF(NDROP(MM))=O.O                              00058800
          FF(NDROP(MM))=O.O                          00058900
997   DO 593 LL=1,NCT                                 00059000
          XF(NDROP(MM),LL)=X(NDROP(MM),LL)          00059100
593   NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)            00059200
592   CONTINUE                                       00059300
      WRITE(NO,8889)                                   00059400
8889  FORMAT(//5X,'THE INTERMEDIATE FINAL SOLUTION HAS BEEN FOUND'//) 00059500
      WRITE(NO,2927)                                   00059600
2927  FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'=====',8X,'=====',5X,'=====',
* 5X,'=====',//)                                   00059700
      DO 2928 MM=1,MPT                                 00060000
2928  WRITE(NO,2929) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
* AF(NDROP(MM))                                       00060200
2929  FORMAT(1X,I4,8X,I8;8X,F14.6,2X,F14.6//)        00060300
      WRITE(NO,2931)                                   00060400
2931  FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
* ,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
* ,12X,'=====',//)                                   00060600
      DO 2422 LL=1,NCT                                 00060700
      DO 2423 MM=1,MPT                                 00060800
          IF(NZF(NDROP(MM),LL)-1) 2423,2424,2423    00060900
2424  WRITE(NO,2425) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00061000
2423  CONTINUE                                       00061100
2422  CONTINUE                                       00061200
      WRITE(NO,3379) TPFIN                             00061300
3379  FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)    00061400
2425  FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)     00061500
693   DO 694 MM=1,MPT                                 00061600
          IF(NY(NDROP(MM))-1) 694,699,694           00061700
699   ADROP(MM)=A(NDROP(MM))                         00061800
      DO 695 LL=1,NCT                                 00061900
          X(NDROP(MM),LL)=O.O                       00062000
695   NZ(NDROP(MM),LL)=O                             00062100
694   CONTINUE                                       00062200
          IF(IDN-1) 560.565,560                     00062300
560   NY(NDROP(KK))=1                                 00062400
          FDROP(KK)=F(NDROP(KK))                    00062500
          ADROP(KK)=A(NDROP(KK))                    00062600
          IF(IDN-1) 520,562,520                     00062700
562   IDN=O                                           00062800
520   CONTINUE                                       00062900
C*****
C SET ORIGINAL PARAMETERS EQUAL TO THE FINAL PARAMETERS AND
C THE INTERMEDIATE PARAMETERS AS APPROPRIATE
C*****
      TOFIX=TFFIX                                     00063000
      TRORG=TRFIN                                     00063100
      TPORG=TPFIN                                     00063200
      DO 801 MM=1,MPT                                  00063300
          NYO(NDROP(MM))=NYF(NDROP(MM))             00063400
          NYO(NDROP(MM))=NYF(NDROP(MM))             00063500
          NYO(NDROP(MM))=NYF(NDROP(MM))             00063600
          NYO(NDROP(MM))=NYF(NDROP(MM))             00063700
          NYO(NDROP(MM))=NYF(NDROP(MM))             00063800
          NYO(NDROP(MM))=NYF(NDROP(MM))             00063900

```

```

      NY(NDROP(MM))=NYF(NDROP(MM))
      IF(NN-MM) 803,802,803
802  IDROP(MM)=0
      NCON=NCON+1
      IF(NCON-MPT) 803,435,435
803  IF(NY(NDROP(MM))-1) 804,805,804
805  ADROP(MM)=A(NDROP(MM))
      FDROP(MM)=F(NDROP(MM))
      GO TO 806
804  ADROP(MM)=0.0
      FDROP(MM)=0.0
806  DO 807 LL=1,NCT
      XO(NDROP(MM),LL)=XF(NDROP(MM),LL)
807  NZO(NDROP(MM),LL)=NZF(NDROP(MM),LL)
801  CONTINUE
      DO 741 LL=1,NCT
741  ICENT(LL)=0
      GO TO 327
C*****
C PRINT ALL FINAL VALUES
C*****
435  IF(NORG=0) 493,5074,493
493  WRITE(NDT,967) TPFIN
967  FORMAT(D20.10)
      WRITE(NO,8883)
8883 FORMAT(//5X,'THE FINAL SOLUTION HAS BEEN FOUND'//)
      WRITE(NO,2001)
2001 FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'=====',8X,'=====',5X,'=====',
*5X,'=====','//)
      DO 2002 MM=1,MPT
2002 WRITE(NO,2003) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
*AF(NDROP(MM))
2003 FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)
      WRITE(NO,2004)
2004 FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
*,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
* ,12X,'=====','//)
      DO 2005 LL=1,NCT
      DO 2006 MM=1,MPT
          IF(NZF(NDROP(MM),LL)-1) 2006,2023,2006
2023 WRITE(NO,2007) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2006 CONTINUE
2005 CONTINUE
2007 FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
      WRITE(NO,3379) TPFIN
      STOP
      END
C*****
C SUBROUTINE TO PERFORM GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
      SUBROUTINE GLOBE
      INTEGER RO,MA,F1,E0,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
*NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
*IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
*FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM,
*SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
*TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
*SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
*XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
*A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
*TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
* ,KK,UJ
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS

```

```

TCAP=0.0                                00071100
TRGLBL=0.0                               00071200
DO 70 LL=1,NCT                            00071300
  PRMAX=0.0                               00071400
  CAPMIN=1.0D20                           00071500
DO 75 MM=1,MPT                            00071600
  IF(NY(NDROP(MM))-1) 75,80,75            00071700
80   IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 75,85,85 00071800
85   IF(CAPMIN-XSTAR(NDROP(MM),LL)) 90,90,95 00071900
95   CAPMIN=XSTAR(NDROP(MM),LL)           00072000
90   IF(PRMAX-PR(NDROP(MM),LL)) 100,75,75 00072100
100  PRMAX=PR(NDROP(MM),LL)               00072200
75   CONTINUE                             00072300
      TCAP=TCAP+CAPMIN                     00072400
      TRGLBL=TRGLBL+PRMAX                  00072500
70   CONTINUE                             00072600
      SUMC=0.0                             00072700
      TFIX=0.0                             00072800
DO 105 MM=1,MPT                           00072900
  IF(NY(NDROP(MM))-1) 105,110,105         00073000
110  SUMC=SUMC+ADROP(MM)                   00073100
      TFIX=TFIX+FDROP(MM)                  00073200
105  CONTINUE                             00073300
      TPGLBL=TRGLBL-TFIX                    00073400
      IF(SUMC-TCAP) 115,120,120            00073500
115  IVECT1=0                             00073600
      WRITE(NO,5888)                         00073700
5888  FORMAT(5X,'THE GLOBAL TEST FOR FEASIBLE CAPACITY FAILED'//) 00073800
      GO TO 133                             00073900
120  IF(TPORG-TPGLBL) 125,117,117          00074000
117  IF(NORG=0) 118,119,118                00074100
118  TPROF=TPGLBL                          00074200
119  IVECT2=0                             00074300
      WRITE(NO,5889)                         00074400
5889  FORMAT(5X,'THE GLOBAL TEST FOR OPTIMAL PROFITS FAILED'//) 00074500
      GO TO 133                             00074600
125  WRITE(NO,5890)                         00074700
5890  FORMAT(5X,'PASSED BOTH GLOBAL TESTS FOR FEASIBLE CAPACITY AND OPTI00074800
      *MAL PROFITS'//)                       00074900
C*****                                     00075000
C CALL THE SUBROUTINE ALOC TO PERFORM FEASIBLE ALLOCATIONS 00075100
C*****                                     00075200
      CALL ALOC                              00075300
133  RETURN                                00075400
      END                                    00075500
C*****                                     00075600
C SUBROUTINE TO PERFORM FEASIBLE ALLOCATIONS 00075700
C*****                                     00075800
      SUBROUTINE ALOC                        00075900
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00076000
      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00076100
      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00076200
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00076300
      *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00076400
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00076500
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00076600
      *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00076700
      *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00076800
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00076900
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00077000
      *TPFIN,TPROF 00077100
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00077200
      *,KK,JJ 00077300
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00077400
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00077500
      COMMON/BLOCK5/NZO,NZ,NZF 00077600
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00077700
      NALOC=0 00077800
      NTEST=1 00077900
      TREV=0.0 00078000
DO 135 JJ=1,MNTOT                          00078100

```

```

          IF(ICENT(NCENT(JJ))-1) 7290,135,7290          00078200
7290    IF(NY(NPLANT(JJ))-1) 135,707,135              00078300
707    DO 708 IB=1,MPT                                00078400
          IF(NPLANT(JJ)-NDROP(IB)) 708,140,708        00078500
708    CONTINUE                                       00078600
140    IF(ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ))) 135,150,150 00078700
150    X(NPLANT(JJ),NCENT(JJ))=XSTAR(NPLANT(JJ),NCENT(JJ)) 00078800
          NZ(NPLANT(JJ),NCENT(JJ))=1                 00078900
          ICENT(NCENT(JJ))=1                          00079000
          ADROP(IB)=ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ)) 00079100
          TREV=TREV+PROF(JJ)                          00079200
          NALOC=NALOC+1                                00079300
          IF(NALOC-NCT) 152,157,157                  00079400
152    IF(AMOD(FLOAT(NALOC),FLOAT(NTEST))-O.O) 135,155,135 00079500
C*****
C CALL THE SUBROUTINE PARTL TO PERFORM PARTIAL TESTS FOR 00079600
C OPTIMAL PROFITS AND FEASIBLE CAPACITY                00079700
C*****
155    CALL PARTL                                       00080000
          IF(IVECT1-O) 132,137,132                    00080100
132    IF(IVECT2-O) 135,137,135                      00080200
135    CONTINUE                                       00080300
          IF(NALOC-NCT) 149,157,157                  00080400
157    TPROF=TREV-TFIX                                00080500
          WRITE(NO,6237) TPROF                        00080600
6237    FORMAT(/5X,'THE TOTAL PROFIT FOR THIS CONFIGURATION IS= ', 00080700
          *F14.6//)                                    00080800
          IF(NORG-O) 151,139,151                      00080900
139    IF(TPORG-TPROF) 141,137,137                  00081000
141    TOFIX=TFIX                                     00081100
          TFFIX=TFIX                                  00081200
          TRORG=TREV                                   00081300
          TRFIN=TREV                                   00081400
          TPORG=TPROF                                  00081500
          TPFIN=TPROF                                  00081600
          DO 142 MM=1,MPT                              00081700
            NYO(NDROP(MM))=NY(NDROP(MM))              00081800
            NYF(NDROP(MM))=NY(NDROP(MM))              00081900
            IF(NY(NDROP(MM))-1) 142,146,142           00082000
146    ADROP(MM)=A(NDROP(MM))                        00082100
            AF(NDROP(MM))=A(NDROP(MM))                00082200
            FF(NDROP(MM))=F(NDROP(MM))                00082300
          DO 143 LL=1,NCT                              00082400
            XO(NDROP(MM),LL)=X(NDROP(MM),LL)          00082500
            XF(NDROP(MM),LL)=X(NDROP(MM),LL)          00082600
            NZO(NDROP(MM),LL)=NZ(NDROP(MM),LL)        00082700
            NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)        00082800
            X(NDROP(MM),LL)=O.O                        00082900
143    NZ(NDROP(MM),LL)=O                            00083000
142    CONTINUE                                       00083100
          NORG=1                                       00083200
          WRITE(NO,5893)                               00083300
5893    FORMAT(5X,'THE ORIGINAL FEASIBLE SOLUTION HAS BEEN FOUND'//) 00083400
4232    WRITE(NO,3022)                                00083500
3022    FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X, 00083600
          *'CAPACITY',/1X,'====',8X,'=====',5X,'=====', 00083700
          *5X,'=====',//)                             00083800
          DO 3023 MM=1,MPT                              00083900
3023    WRITE(NO,3024) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)), 00084000
          *AF(NDROP(MM))                                00084100
3024    FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)      00084200
          WRITE(NO,3025)                                00084300
3025    FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X 00084400
          *,'PROFIT',/1X,'=====',5X,'=====',8X,'=====' 00084500
          *,12X,'=====',//)                             00084600
          DO 3026 LL=1,NCT                              00084700
          DO 3027 MM=1,MPT                              00084800
            IF(NZF(NDROP(MM),LL)-1) 3027,3051,3027    00084900
3051    WRITE(NO,3028) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00085000
3027    CONTINUE                                       00085100
3026    CONTINUE                                       00085200

```

```

3028 FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
WRITE(NO,3380) TPFIN
3380 FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)
GO TO 151
149 IVECT1=0
137 DO 153 MM=1,MPT
IF(NY(NDROP(MM))-1) 153,710,153
710 ADROP(MM)=A(NDROP(MM))
DO 154 LL=1,NCT
X(NDROP(MM),LL)=O.O
154 NZ(NDROP(MM),LL)=O
153 CONTINUE
151 DO 738 LL=1,NCT
738 ICENT(LL)=O
RETURN
END
C*****
C SUBROUTINE TO PERFORM PARTIAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
SUBROUTINE PARTL
INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
*NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
*IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
*FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROP,ADROP,
*SUMR,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
*TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
*SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
*XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
*A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
*TPFIN,TPROF
COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
*,KK,JJ
COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
COMMON/BLOCK5/NZO,NZ,NZF
COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
RCAP=O.O
TRPTL=O.O
DO 160 LL=1,NCT
PRMAX=O.O
CAPMIN=1.OD2O
IF(ICENT(LL)-1) 736,160,736
736 DO 165 MM=1,MPT
IF(NY(NDROP(MM))-1) 165,175,165
175 IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 165,180,180
180 IF(CAPMIN-XSTAR(NDROP(MM),LL)) 185,185,190
190 CAPMIN=XSTAR(NDROP(MM),LL)
185 IF(PRMAX-PR(NDROP(MM),LL)) 195,165,165
195 PRMAX=PR(NDROP(MM),LL)
165 CONTINUE
RCAP=RCAP+CAPMIN
TRPTL=TRPTL+PRMAX
160 CONTINUE
TRPTL=TREV+TRPTL
SUMR=O.O
DO 200 MM=1,MPT
IF(NY(NDROP(MM))-1) 200,205,200
205 SUMR=SUMR+ADROP(MM)
200 CONTINUE
TPPTL=TRPTL-TFIX
IF(SUMR-RCAP) 210,215,215
210 IVECT1=O
WRITE(NO,5891)
5891 FORMAT(5X,'THE PARTIAL TEST FOR FEASIBLE CAPACITY FAILED'//)
GO TO 220
215 IF(TPORG-TPPTL) 220,225,225
225 IF(NORG-O) 213,221,213
213 TPROF=TPPTL

```

```

221      IVECT2=0                                00092400
        WRITE(NO,5892)                          00092500
5892     FORMAT(5X,'THE PARTIAL TEST FOR OPTIMAL PROFITS FAILED'//) 00092600
        GO TO 220                               00092700
220     RETURN                                  00092800
        END                                     00092900
C*****                                        00093000
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES 00093100
C BETWEEN THE LOWER LIMIT B AND UPPER LIMIT C          00093200
C*****                                        00093300
        SUBROUTINE UNFRM                          00093400
        INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00093500
        *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00093600
        *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00093700
        REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00093800
        *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00093900
        *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00094000
        *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00094100
        *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00094200
        *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00094300
        *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00094400
        COMMON/BLOCK7/B,C,D                      00094500
        COMMON/BLOCK8/IX,IY,YFL                 00094600
        CALL RANDU                               00094700
        Q=YFL                                    00094800
        D=B+(C-B)*Q                             00094900
        RETURN                                  00095000
        END                                     00095100
C*****                                        00095200
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES 00095300
C BETWEEN 0 AND 1                                    00095400
C*****                                        00095500
        SUBROUTINE RANDU                          00095600
        INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00095700
        *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00095800
        *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00095900
        REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00096000
        *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00096100
        *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00096200
        *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00096300
        *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00096400
        *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00096500
        *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00096600
        COMMON/BLOCK8/IX,IY,YFL                 00096700
        IY=IX*65539                              00096800
        IF(IY) 5,6,6                             00096900
5         IY=IY+2147483647+1                    00097000
6         YFL=IY                                00097100
        YFL=YFL*0.4656613E-9                   00097200
        RETURN                                  00097300
        END                                     00097400

```

```

C*****0000100
C0000200
C THIS PROGRAM PERFORMS THE HEURISTIC ALGORITHM FOR PLANT 0000300
C LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE 0000400
C STOCHASTIC DEMAND 0000500
C 0000600
C ALSO IT SETS UP THE INPUT DATA DECK FOR EXTREME POINT 0000700
C RANKING TECHNIQUE 0000800
C 0000900
C THE PLANTS ARE ASSUMED CAPACITATED 0001000
C AND A UNIFORMLY DISTRIBUTED DEMAND IS ASSUMED 0001100
C 0001200
C THIS ALGORITHM TAKES INTO ACCOUNT OF THE FACT THAT 0001300
C ALL DEMAND CENTERS NEED TO BE ALLOCATED AND THAT 0001400
C EACH DEMAND CENTER MUST RECEIVE THE SUPPLY 0001500
C FROM AT MOST ONE PLANT 0001600
C 0001700
C WRITTEN BY LOGENDRAN RASARATNAM 0001800
C SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT 0001900
C OKLAHOMA STATE UNIVERSITY 0002000
C 0002100
C DISSERTATION ADVISER: DR. M. PALMER TERRELL 0002200
C 0002300
C VERSION 1 -- AUGUST, 1984 0002400
C*****0002500
C 0002600
C*****0002700
C 0002800
C*** GENERAL STRUCTURE AND INPUT REQUIREMENTS: 0002900
C (MAIN PROGRAM DRIVES THE SUBROUTINES UNFRM AND GLOBE) 0003000
C (UNFRM DRIVES SUBROUTINE RANDU) 0003100
C (GLOBE DRIVES SUBROUTINE ALOC) 0003200
C (ALOC DRIVES SUBROUTINE PARTL) 0003300
C 0003400
C*****0003500
C SUBROUTINE FUNCTION 0003600
C -----
C RANDU TO GENERATE A UNIFORMLY DISTRIBUTED 0003900
C RANDOM NUMBER BETWEEN 0 AND 1 0004000
C UNFRM TO CONVERT THE RANDOM NUMBER GENERATED 0004100
C BY RANDU BETWEEN APPROPRIATE LIMITS 0004200
C ESTABLISHED FOR THE UNIFORM 0004300
C DISTRIBUTION TO BE USED 0004400
C GLOBE PERFORMS THE GLOBAL TESTS FOR OPTIMAL 0004500
C PROFITS AND FEASIBLE CAPACITY 0004600
C ALOC PERFORMS THE FEASIBLE ALLOCATIONS FOR 0004700
C FOR EACH OF THE DEMAND CENTERS 0004800
C DICTATED BY PRIORITY RULE 1 0004900
C PARTL PERFORMS THE PARTIAL TESTS FOR OPTIMAL 0005000
C PROFITS AND FEASIBLE CAPACITY 0005100
C 0005200
C*****0005300
C 0005400
C*** NO EXTERNAL FUNCTIONS REQUIRED 0005500
C 0005600
C*****0005700
C 0005800
C*** COMMON BLOCK VARIABLE DEFINITIONS 0005900
C ONLY THOSE VARIABLES THAT REQUIRE EXPLANATION ARE LISTED 0006000
C 0006100
C /BLOCK1/ 0006200
C 0006300
C TOFIX - TOTAL FIXED COST FOR THE ORIGINAL LOCATION VECTOR 0006400
C UNDER CONSIDERATION 0006500
C TFIX - TOTAL FIXED COST FOR THE LOCATION VECTOR PRESENTLY 0006600
C CONSIDERED BY DROPPING ONE PLANT AT A TIME 0006700
C FROM THE ORIGINAL LOCATION VECTOR 0006800
C TFFIX - TOTAL FIXED COST FOR THE FINAL LOCATION VECTOR 0006900
C CURRENTLY STORED 0007000
C TRORG - TOTAL REVENUE FOR THE ORIGINAL LOCATION VECTOR 0007100

```

C		UNDER CONSIDERATION	00007200
C	TREV	- TOTAL REVENUE FOR THE LOCATION VECTOR PRESENTLY	00007300
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007400
C		FROM THE ORIGINAL LOCATION VECTOR	00007500
C	TRFIN	- TOTAL REVENUE FOR THE FINAL LOCATION VECTOR	00007600
C		CURRENTLY STORED	00007700
C	TPORG	- TOTAL PROFIT FOR THE ORIGINAL LOCATION VECTOR	00007800
C		UNDER CONSIDERATION	00007900
C	TPROF	- TOTAL PROFIT FOR THE LOCATION VECTOR PRESENTLY	00008000
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008100
C		FROM THE ORIGINAL LOCATION VECTOR	00008200
C	TPFIN	- TOTAL PROFIT FOR THE FINAL LOCATION VECTOR	00008300
C		CURRENTLY STORED	00008400
C			00008500
C	/BLOCK2/		00008600
C			00008700
C	IVECT1-	INDICATOR VARIABLE FOR THE TEST ON FEASIBLE CAPACITY	00008800
C	IVECT2-	INDICATOR VARIABLE FOR THE TEST ON OPTIMAL PROFITS	00008900
C	NORG	- VARIABLE INDICATING THAT THE ORIGINAL FEASIBLE	00009000
C		SOLUTION HAS BEEN FOUND	00009100
C	NCON	- TOTAL NUMBER OF PLANTS DROPPED SO FAR	00009200
C	NALOC	- TOTAL NUMBER OF DEMAND CENTERS ALLOCATED SO FAR	00009300
C	MPT	- TOTAL NUMBER OF PLANTS FOR THE PROBLEM SOLVED	00009400
C	NCT	- TOTAL NUMBER OF DEMAND CENTERS FOR THE PROBLEM SOLVED	00009500
C			00009600
C	/BLOCK3/		00009700
C			00009800
C	NYO	- THE ORIGINAL LOCATION VECTOR UNDER CONSIDERATION	00009900
C	NY	- THE LOCATION VECTOR PRESENTLY CONSIDERED BY	00010000
C		DROPPING ONE PLANT AT A TIME FROM THE	00010100
C		ORIGINAL LOCATION VECTOR	00010200
C	NYF	- THE FINAL LOCATION VECTOR CURRENTLY STORED	00010300
C	NPLANT-	VARIABLE USED FOR THE PLANTS IN RANKING THE OPTIMAL	00010400
C		PROFITS EMPLOYING PRIORITY RULE 1	00010500
C	NCENT	- VARIABLE USED FOR THE DEMAND CENTERS IN RANKING THE	00010600
C		OPTIMAL PROFITS EMPLOYING PRIORITY RULE 1	00010700
C	ICENT	- VARIABLE INDICATING WHETHER A DEMAND CENTER HAS	00010800
C		ALREADY BEEN ALLOCATED	00010900
C	IDROP	- VARIABLE INDICATING THE PLANT PRESENTLY DROPPED	00011000
C		FROM THE ORIGINAL LOCATION VECTOR	00011100
C	NDROP	- VARIABLE USED FOR RANKING THE PLANTS FOR DROP	00011200
C		EMPLOYING PRIORITY RULE 2	00011300
C			00011400
C	/BLOCK4/		00011500
C			00011600
C	XSTAR	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR EACH	00011700
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00011800
C	XO	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE ORIGINAL	00011900
C		LOCATION VECTOR UNDER CONSIDERATION	00012000
C	X	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE LOCATION	00012100
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00012200
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00012300
C	XF	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE FINAL	00012400
C		LOCATION VECTOR CURRENTLY STORED	00012500
C	PR	- OPTIMAL PROFITS DETERMINED FOR EACH COMBINATION OF	00012600
C		DEMAND CENTERS AND PROFITS	00012700
C	PROF	- THE RANKED OPTIMAL PROFITS DETERMINED FOR EACH	00012800
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00012900
C		USING PRIORITY RULE 1	00013000
C			00013100
C	/BLOCK5/		00013200
C			00013300
C	NZO	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE ORIGINAL	00013400
C		LOCATION VECTOR UNDER CONSIDERATION	00013500
C	NZ	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE LOCATION	00013600
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00013700
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00013800
C	NZF	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE FINAL	00013900
C		LOCATION VECTOR CURRENTLY STORED	00014000
C			00014100
C	/BLOCK6/		00014200


```

C                                                    00014300
C      A      - CAPACITY OF EACH PLANT                00014400
C      F      - FIXED COST OF EACH PLANT             00014500
C      ADROP- THE REMAINING CAPACITY OF EACH PLANT AFTER HAVING
C              ALLOCATED A SUBSET OF DEMAND CENTERS  00014600
C                                                    00014700
C                                                    00014800
C*****00014900
C                                                    00015000
C*** OTHER VARIABLE DEFINITIONS                    00015100
C                                                    00015200
C      A1     - THE LOWER LIMIT OF THE UNIFORM DISTRIBUTION OF
C              DEMAND DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS             00015300
C                                                    00015400
C      A2     - THE UPPER LIMIT OF THE UNIFORM DISTRIBUTION OF
C              DEMAND DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS             00015500
C                                                    00015600
C      MU     - MEAN OF THE UNIFORM DISTRIBUTION OF DEMAND
C              DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS             00015700
C                                                    00015800
C      MU     - MEAN OF THE UNIFORM DISTRIBUTION OF DEMAND
C              DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS             00015900
C                                                    00016000
C      R      - PER UNIT VARIABLE COST OF THE PRODUCT SUPPLIED FOR
C              EACH COMBINATION OF DEMAND CENTERS AND PLANTS 00016100
C                                                    00016200
C      PC     - PER UNIT PRICE OF THE PRODUCT RECEIVED FOR EACH
C              COMBINATION OF DEMAND CENTERS AND PLANTS 00016300
C                                                    00016400
C                                                    00016500
C                                                    00016600
C*****00016700
C      INTEGER RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
C      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
C      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
C      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,
C      *FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROP,ADROP,
C      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
C      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
C      *MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
C      *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
C      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
C      DATA RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS/2HRO,2HMA,
C      *2HFI,2HED,1HO,3HPRD,2HFT,3HEQU,3HINE,1HZ,1HY,1H+/
C      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
C      *TPFIN,TPROF
C      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
C      *,KK,JJ
C      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
C      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
C      COMMON/BLOCK5/NZO,NZ,NZF
C      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
C      COMMON/BLOCK7/B,C,D
C      COMMON/BLOCK8/IX,IY,YFL
C*****00018900
C      NO=12
C      NOT=14
C      MPT=3
C      NCT=4
C      CONST1=1.0
C      CONST2=0.0
C      MNTOT=MPT*NCT
C      MQ=NCT+MPT
C      MR=MPT*NCT+MPT
C*****00019900
C      GENERATE VALUES RANDOMLY FOR F(MM)          00020000
C      SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C      DISTRIBUTION                                  00020100
C*****00020200
C      B=200.
C      C=300.
C      IX=13
C      WRITE(NO,4025)
C      4025 FORMAT(//5X,'THE PLANT FIXED COSTS ARE',//)
C      WRITE(NO,5248)
C      5248 FORMAT(5X,'FIXED COST'.6X,'PLANT'//)
C      DO 1001 MM=1,MPT
C      CALL UNFRM
C*****00021100
C*****00021200
C*****00021300

```

```

F(MM)=D
WRITE(NO,5249) D,MM
5249 FORMAT(1X,F14.6,7X,I2//)
IX=IY
1001 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR A(MM)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=400.
      C=450.
      WRITE(NO,4026)
4026 FORMAT(/5X,'THE PLANT CAPACITIES ARE',//)
      WRITE(NO,5250)
5250 FORMAT(5X,'CAPACITY',6X,'PLANT'//)
      DO 1002 MM=1,MPT
      CALL UNFRM
      A(MM)=D
      WRITE(NO,5251) D,MM
5251 FORMAT(1X,F14.6,5X,I2//)
      IX=IY
      1002 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR R(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=0.5
      C=2.5
      WRITE(NO,4027)
4027 FORMAT(/5X,'THE PER UNIT VARIABLE COSTS ARE'//)
      WRITE(NO,5252)
5252 FORMAT(1X,'PER UNIT VARIABLE COST',5X,'PLANT',5X,'DEMAND CENTER',
*/)
      DO 1003 MM=1,MPT
      DO 1004 LL=1,NCT
      CALL UNFRM
      R(MM,LL)=D
      WRITE(NO,5253) D,MM,LL
5253 FORMAT(5X,F14.6,9X,I2,13X,I2//)
      1004 IX=IY
      1003 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR THE LOWER AND UPPER LIMITS OF THE
C UNIFORM DISTRIBUTION DESCRIPTIVE OF THE DEMAND
C*****
      WRITE(NO,4028)
4028 FORMAT(/1X,'THE PER UNIT PRICE AND THE MEAN OF THE DEMAND DISTRIBUTION
FOR THE PRODUCT IS'//)
      WRITE(NO,5254)
5254 FORMAT(4X,'PER UNIT PRICE',4X,'MEAN DEMAND',11X,'LOWER',13X,
*'UPPER',13X,'PLANT',5X,'DEMAND CENTER'//)
      B1=50.
      C1=100.
      B2=150.
      C2=300.
      DO 7005 MM=1,MPT
      DO 7006 LL=1,NCT
      B=B1
      C=C1
      CALL UNFRM
      A1(MM,LL)=D
      IX=IY
      B=B2
      C=C2
      CALL UNFRM
      A2(MM,LL)=D
      MU(MM,LL)=(A1(MM,LL)+A2(MM,LL))/2.
      PC(MM,LL)=600./MU(MM,LL)
      WRITE(NO,5255) PC(MM,LL),MU(MM,LL),A1(MM,LL),A2(MM,LL),MM,LL
00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00023000
00023100
00023200
00023300
00023400
00023500
00023600
00023700
00023800
00023900
00024000
00024100
00024200
00024300
00024400
00024500
00024600
00024700
00024800
00024900
00025000
00025100
00025200
00025300
00025400
00025500
00025600
00025700
00025800
00025900
00026000
00026100
00026200
00026300
00026400
00026500
00026600
00026700
00026800
00026900
00027000
00027100
00027200
00027300
00027400
00027500
00027600
00027700
00027800
00027900
00028000
00028100
00028200
00028300
00028400

```



```

15   PROFM=PROF(MN)                                00042700
      NPLAM=NPLANT(MN)                             00042800
      NCENTM=NCENT(MN)                             00042900
      DO 20 I=MN,MNTOT                             00043000
          IF(PROFM-PROF(I)) 25,25,20              00043100
25   PROFM=PROF(I)                                00043200
      NPLAM=NPLANT(I)                             00043300
      NCENTM=NCENT(I)                             00043400
      IMAX=I                                       00043500
20   CONTINUE                                     00043600
      PROF(IMAX)=PROF(MN)                         00043700
      NPLANT(IMAX)=NPLANT(MN)                    00043800
      NCENT(IMAX)=NCENT(MN)                      00043900
      PROF(MN)=PROFM                             00044000
      NPLANT(MN)=NPLAM                          00044100
      NCENT(MN)=NCENTM                          00044200
      MN=MN+1                                    00044300
      IF(MN-MNTOT) 15,15,35                      00044400
35   WRITE(NO,4030)                               00044500
4030 FORMAT(5X,'THE SORTED PROFITS AND THE RESPECTIVE PLANTS AND THE DE00044600
      *MAND CENTERS ARE'//)                       00044700
      WRITE(NO,4034)                              00044800
4034 FORMAT(5X,'RANKED PROFIT',10X,'PLANT',10X,'DEMAND CENTER'//) 00044900
      DO 5033 I=1,MNTOT                          00045000
5033  WRITE(NO,4035) PROF(I),NPLANT(I),NCENT(I)  00045100
4035  FORMAT(3X,F14.6,12X,I2,17X,I2//)          00045200
C*****                                           00045300
C SORT THE PLANTS IN A DECENDING ORDER OF FIXED COSTS 00045400
C*****                                           00045500
      DO 333 I=1,MPT                              00045600
          FDROP(I)=F(I)                          00045700
          ADROP(I)=A(I)                         00045800
333   NDROP(I)=I                                 00045900
          MJ=1                                   00046000
37   FDROPM=FDROP(MJ)                          00046100
          ADROPM=ADROP(MJ)                    00046200
          NDROPM=NDROP(MJ)                   00046300
      DO 40 I=MJ,MPT                             00046400
          IF(FDROPM-FDROP(I)) 45,45,40         00046500
45   FDROPM=FDROP(I)                          00046600
          ADROPM=ADROP(I)                    00046700
          NDROPM=NDROP(I)                   00046800
          IMAX=I                              00046900
40   CONTINUE                                  00047000
          FDROP(IMAX)=FDROP(MJ)              00047100
          ADROP(IMAX)=ADROP(MJ)             00047200
          NDROP(IMAX)=NDROP(MJ)            00047300
          FDROP(MJ)=FDROPM                 00047400
          ADROP(MJ)=ADROPM                00047500
          NDROP(MJ)=NDROPM                00047600
          IDROP(MJ)=1                      00047700
          MJ=MJ+1                          00047800
          IF(MJ-MPT) 37,37,50              00047900
50   WRITE(NO,4031)                            00048000
4031 FORMAT(5X,'THE SORTED FIXED COSTS AND OTHER RESPECTIVE PARAMETERS 00048100
      *ARE',//)                                00048200
          WRITE(NO,4032)                    00048300
4032 FORMAT(1X,'RANKED PLANTS FOR DROP',10X,'FIXED COST',11X,'CAPACITY' 00048400
      *,11X,'FIXED COST',10X,'CAPACITY',12X,'IDROP VALUES'//) 00048500
          DO 5034 I=1,MPT                   00048600
5034  WRITE(NO,4033) NDROP(I),FDROP(I),ADROP(I),F(NDROP(I)), 00048700
      *A(NDROP(I)),IDROP(I)                00048800
4033  FORMAT(10X,I2,17X,F14.6,6X,F14.6,6X,F14.6,6X,F14.6,10X,I7//) 00048900
C*****                                           00049000
C START WITH ALL PLANTS OPEN                    00049100
C SET ALL NY(I) EQUAL TO 1                     00049200
C*****                                           00049300
      DO 55 K=1,MPT                             00049400
55   NY(NDROP(K))=1                          00049500
C*****                                           00049600
C INITIALIZE ALL PARAMETERS                    00049700

```

```

C*****
    TOFIX=0.0
    TFFIX=0.0
    TRORG=0.0
    TRFIN=0.0
    TPRG=0.0
    TPFIN=0.0
    IVECT1=1
    IVECT2=1
    NORG=0
    NCON=0
    DO 610 MM=1,MPT
        NYO(NDROP(MM))=0
        NYF(NDROP(MM))=0
        AF(NDROP(MM))=0.0
        FF(NDROP(MM))=0.0
    DO 611 LL=1,NCT
        XO(NDROP(MM),LL)=0.0
        XF(NDROP(MM),LL)=0.0
        X(NDROP(MM),LL)=0.0
        NZO(NDROP(MM),LL)=0
        NZF(NDROP(MM),LL)=0
611    NZ(NDROP(MM),LL)=0
610    CONTINUE
    DO 477 LL=1,NCT
477    ICENT(LL)=0
C*****
C CALL THE SUBROUTINE GLOBE TO PERFORM THE GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
    CALL GLOBE
    IF(IVECT1=0) 325,323,325
323    WRITE(NO,103)
103    FORMAT(1X,'FOR THE HEURISTICS EMPLOYED NO FEASIBLE SOLUTION CAN BE
*FOUND DUE TO CAPACITY LIMITATIONS'//)
999    STOP
5074    WRITE(NO,5075)
5075    FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS - STOP'/'*****
*****'//)
    GO TO 999
325    IF(NORG=0) 327,328,327
328    IVECT2=1
C*****
C CONSIDER THE DROP OF EACH PLANT
C*****
327    DO 520 KK=1,MPT
        IF(IDROP(KK)-1) 520,530,520
530    NY(NDROP(KK))=0
        FDROP(KK)=0.0
        ADROP(KK)=0.0
    CALL GLOBE
    IF(IVECT1=0) 535,540,535
540    IVECT1=1
        IDROP(KK)=0
        NCON=NCON+1
        IF(NCON=MPT) 560,435,435
535    IF(IVECT2=0) 550,555,550
555    IVECT2=1
        IF(TPRG-TPROF) 560,565,565
565    DIF=TPRG-TPROF
        IF(DIF-F(NDROP(KK))) 560,560,575
575    IDROP(KK)=0
        NCON=NCON+1
        IF(NCON=MPT) 560,435,435
550    IF(TPRG-TPROF) 585,696,696
696    IDN=1
        GO TO 693
585    IF(TPFIN-TPROF) 590,693,693
590    TFFIX=TFIX
        TRFIN=TREV
        TPFIN=TPROF

```

```

      NN=KK
      DO 592 MM=1,MPT
        NYF(NDROP(MM))=NY(NDROP(MM))
        IF(NYF(NDROP(MM))-1) 783,784,783
784    AF(NDROP(MM))=A(NDROP(MM))
        FF(NDROP(MM))=F(NDROP(MM))
        GO TO 997
783    AF(NDROP(MM))=O.O
        FF(NDROP(MM))=O.O
997    DO 593 LL=1,NCT
        XF(NDROP(MM),LL)=X(NDROP(MM),LL)
593    NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)
592    CONTINUE
        WRITE(NO,8889)
8889    FORMAT(//5X,'THE INTERMEDIATE FINAL SOLUTION HAS BEEN FOUND'//)
        WRITE(NO,2927)
2927    FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
      *'CAPACITY',/1X,'====',8X,'=====',5X,'=====',
      *5X,'=====',//)
        DO 2928 MM=1,MPT
2928    WRITE(NO,2929) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
      *AF(NDROP(MM))
2929    FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)
        WRITE(NO,2931)
2931    FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
      *,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
      *,12X,'=====',//)
        DO 2422 LL=1,NCT
        DO 2423 MM=1,MPT
          IF(NZF(NDROP(MM),LL)-1) 2423,2424,2423
2424    WRITE(NO,2425) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2423    CONTINUE
2422    CONTINUE
        WRITE(NO,3379) TPFIN
3379    FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)
2425    FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
693    DO 694 MM=1,MPT
        IF(NY(NDROP(MM))-1) 694,699,694
699    ADROP(MM)=A(NDROP(MM))
        DO 695 LL=1,NCT
        X(NDROP(MM),LL)=O.O
695    NZ(NDROP(MM),LL)=O
694    CONTINUE
        IF(IDN-1) 560,565,560
560    NY(NDROP(KK))=1
        FDROP(KK)=F(NDROP(KK))
        ADROP(KK)=A(NDROP(KK))
        IF(IDN-1) 520,562,520
562    IDN=O
520    CONTINUE
C*****
C SET ORIGINAL PARAMETERS EQUAL TO THE FINAL PARAMETERS AND
C THE INTERMEDIATE PARAMETERS AS APPROPRIATE
C*****
      TDFIX=TFFIX
      TRORG=TRFIN
      TPORG=TPFIN
      DO 801 MM=1,MPT
        NYO(NDROP(MM))=NYF(NDROP(MM))
        NY(NDROP(MM))=NYF(NDROP(MM))
        IF(NN-MM) 803,802,803
802    IDROP(MM)=O
        NCON=NCON+1
        IF(NCON-MPT) 803,435,435
803    IF(NY(NDROP(MM))-1) 804,805,804
805    ADROP(MM)=A(NDROP(MM))
        FDROP(MM)=F(NDROP(MM))
        GO TO 806
804    ADROP(MM)=O.O
        FDROP(MM)=O.O
806    DO 807 LL=1,NCT

```

```

      XO(NDROP(MM),LL)=XF(NDROP(MM),LL)
807  NZO(NDROP(MM),LL)=NZF(NDROP(MM),LL)
801  CONTINUE
      DO 741 LL=1,NCT
841  ICENT(LL)=0
      GO TO 327
C*****
C PRINT ALL FINAL VALUES
C*****
435  IF(NDRG=0) 493,5074,493
493  WRITE(NDROP(MM),LL) TPFIN
967  FORMAT(D20.10)
      WRITE(NDROP(MM),LL)
8883 FORMAT(/5X,'THE FINAL SOLUTION HAS BEEN FOUND'//)
      WRITE(NDROP(MM),LL)
2001 FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'=====',8X,'=====',5X,'=====',
* 5X,'=====',//)
      DO 2002 MM=1,MPT
2002 WRITE(NDROP(MM),LL) NYF(NDROP(MM)),FF(NDROP(MM)),
* AF(NDROP(MM))
2003 FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)
      WRITE(NDROP(MM),LL)
2004 FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
* 'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
* 12X,'=====',//)
      DO 2005 LL=1,NCT
      DO 2006 MM=1,MPT
          IF(NZF(NDROP(MM),LL)-1) 2006,2023,2006
2023 WRITE(NDROP(MM),LL) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2006 CONTINUE
2005 CONTINUE
2007 FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
      WRITE(NDROP(MM),LL) TPFIN
      STOP
      END
C*****
C SUBROUTINE TO PERFORM GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
      SUBROUTINE GLOBE
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
* NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
* IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
      REAL*8 P,S,B,C,D,Q,YFL,G1,C2,C3,C4,CONST1,CONST2,FAC1,
* FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROPM,ADROPM,
* SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
* TPPTL,TOFIX,TFIX,TFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
* MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
* XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
* A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
* TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NDRG,NCON,NALOC,MPT,NCT,MNTOT,ND
* ,KK,JJ
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
      TCAP=0.0
      TRGLBL=0.0
      DO 70 LL=1,NCT
          PRMAX=0.0
          CAPMIN=1.0D20
      DO 75 MM=1,MPT
          IF(NY(NDROP(MM))-1) 75.80,75
80  IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 75.85,85
85  IF(CAPMIN-XSTAR(NDROP(MM),LL)) 90.90,95
95  CAPMIN=XSTAR(NDROP(MM),LL)
90  IF(PRMAX-PR(NDROP(MM),LL)) 100.75,75
100 PRMAX=PR(NDROP(MM),LL)

```



```

75  CONTINUE                                00071100
    TCAP=TCAP+CAPMIN                        00071200
    TRGLBL=TRGLBL+PRMAX                    00071300
70  CONTINUE                                00071400
    SUMC=0.0                                00071500
    TFIX=0.0                                00071600
    DO 105 MM=1,MPT                          00071700
      IF(NY(NDROP(MM))-1) 105,110,105      00071800
110  SUMC=SUMC+ADROP(MM)                    00071900
      TFIX=TFIX+FDROP(MM)                  00072000
105  CONTINUE                                00072100
    TPGLBL=TRGLBL-TFIX                      00072200
    IF(SUMC-TCAP) 115,120,120              00072300
115  IVECT1=0                                00072400
    WRITE(NO,5888)                          00072500
5888  FORMAT(5X,'THE GLOBAL TEST FOR FEASIBLE CAPACITY FAILED'//) 00072600
    GO TO 133                                00072700
120  IF(TPORG-TPGLBL) 125,117,117          00072800
117  IF(NORG=0) 118,119,118                00072900
118  TPROF=TPGLBL                          00073000
119  IVECT2=0                                00073100
    WRITE(NO,5889)                          00073200
5889  FORMAT(5X,'THE GLOBAL TEST FOR OPTIMAL PROFITS FAILED'//) 00073300
    GO TO 133                                00073400
125  WRITE(NO,5890)                          00073500
5890  FORMAT(5X,'PASSED BOTH GLOBAL TESTS FOR FEASIBLE CAPACITY AND OPTI00073600
      *MAL PROFITS'//)                      00073700
C*****                                     00073800
C CALL THE SUBROUTINE ALOC TO PERFORM FEASIBLE ALLOCATIONS 00073900
C*****                                     00074000
    CALL ALOC                                00074100
133  RETURN                                  00074200
    END                                       00074300
C*****                                     00074400
C SUBROUTINE TO PERFORM FEASIBLE ALLOCATIONS 00074500
C*****                                     00074600
    SUBROUTINE ALOC                          00074700
    INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00074800
    *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00074900
    *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00075000
    REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1, 00075100
    *FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROPM,ADROPM, 00075200
    *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00075300
    *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00075400
    *MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00075500
    *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00075600
    *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00075700
    COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00075800
    *TPFIN,TPROF 00075900
    COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00076000
    *,KK,JJ 00076100
    COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00076200
    COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00076300
    COMMON/BLOCK5/NZO,NZ,NZF 00076400
    COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00076500
    NALOC=0 00076600
    NTEST=1 00076700
    TREV=0.0 00076800
    DO 135 JJ=1,MNTOT 00076900
      IF(ICENT(NCENT(JJ))-1) 7290,135,7290 00077000
7290  IF(NY(NPLANT(JJ))-1) 135,707,135 00077100
707  DO 708 IB=1,MPT 00077200
      IF(NPLANT(JJ)-NDROP(IB)) 708,140,708 00077300
708  CONTINUE 00077400
140  IF(ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ))) 135,150,150 00077500
150  X(NPLANT(JJ),NCENT(JJ))=XSTAR(NPLANT(JJ),NCENT(JJ)) 00077600
      NZ(NPLANT(JJ),NCENT(JJ))=1 00077700
      ICENT(NCENT(JJ))=1 00077800
      ADROP(IB)=ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ)) 00077900
      TREV=TREV+PROF(JJ) 00078000
      NALOC=NALOC+1 00078100

```



```

151 DO 738 LL=1,NCT                                00085300
738 ICENT(LL)=O                                    00085400
      RETURN                                        00085500
      END                                           00085600
C*****                                           00085700
C SUBROUTINE TO PERFORM PARTIAL TESTS FOR          00085800
C OPTIMAL PROFITS AND FEASIBLE CAPACITY           00085900
C*****                                           00086000
      SUBROUTINE PARTL                              00086100
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00086200
      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00086300
      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00086400
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1, 00086500
      *FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROPM,ADROPM, 00086600
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00086700
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00086800
      *MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00086900
      *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00087000
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00087100
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00087200
      *TPFIN,TPROF 00087300
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00087400
      *,KK,JJ 00087500
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00087600
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00087700
      COMMON/BLOCK5/NZO,NZ,NZF 00087800
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00087900
      RCAP=0.0 00088000
      TRPTL=0.0 00088100
      DO 160 LL=1,NCT 00088200
      PRMAX=0.0 00088300
      CAPMIN=1.0D20 00088400
      IF(ICENT(LL)-1) 736,160,736 00088500
736 DO 165 MM=1,MPT 00088600
      IF(NY(NDROP(MM))-1) 165,175,165 00088700
      IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 165,180,180 00088800
      IF(CAPMIN-XSTAR(NDROP(MM),LL)) 185,185,190 00088900
      CAPMIN=XSTAR(NDROP(MM),LL) 00089000
      IF(PRMAX-PR(NDROP(MM),LL)) 195,165,165 00089100
      PRMAX=PR(NDROP(MM),LL) 00089200
      CONTINUE 00089300
      RCAP=RCAP+CAPMIN 00089400
      TRPTL=TRPTL+PRMAX 00089500
      CONTINUE 00089600
      TRPTL=TREV+TRPTL 00089700
      SUMR=0.0 00089800
      DO 200 MM=1,MPT 00089900
      IF(NY(NDROP(MM))-1) 200,205,200 00090000
      SUMR=SUMR+ADROP(MM) 00090100
      CONTINUE 00090200
      TPPTL=TRPTL-TFIX 00090300
      IF(SUMR-RCAP) 210,215,215 00090400
      IVECT1=0 00090500
      WRITE(NO,5891) 00090600
      5891 FORMAT(5X,'THE PARTIAL TEST FOR FEASIBLE CAPACITY FAILED'//) 00090700
      GO TO 220 00090800
      IF(TPORG-TPPTL) 220,225,225 00090900
      IF(NORG-O) 213,221,213 00091000
      TPROF=TPPTL 00091100
      IVECT2=0 00091200
      WRITE(NO,5892) 00091300
      5892 FORMAT(5X,'THE PARTIAL TEST FOR OPTIMAL PROFITS FAILED'//) 00091400
      GO TO 220 00091500
      220 RETURN 00091600
      END 00091700
C*****                                           00091800
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES 00091900
C BETWEEN THE LOWER LIMIT B AND UPPER LIMIT C      00092000
C*****                                           00092100
      SUBROUTINE UNFRM                              00092200
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00092300

```



```

C*****0000100
C                                0000200
C    THIS PROGRAM PERFORMS THE HEURISTIC ALGORITHM FOR PLANT 00000300
C    LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE      00000400
C    STOCHASTIC DEMAND                                     00000500
C                                                         00000600
C    ALSO IT SETS UP THE INPUT DATA DECK FOR EXTREME POINT 00000700
C    RANKING TECHNIQUE                                    00000800
C                                                         00000900
C    THE PLANTS ARE ASSUMED UNCAPACITATED                 00001000
C    AND A TYPE 1 NORMALLY DISTRIBUTED DEMAND IS ASSUMED 00001100
C                                                         00001200
C    THIS ALGORITHM TAKES INTO ACCOUNT OF THE FACT THAT   00001300
C    ALL DEMAND CENTERS NEED TO BE ALLOCATED AND THAT     00001400
C    EACH DEMAND CENTER MUST RECEIVE THE SUPPLY           00001500
C    FROM AT MOST ONE PLANT                               00001600
C                                                         00001700
C    WRITTEN BY LOGENDRAN RASARATNAM                     00001800
C    SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT      00001900
C    OKLAHOMA STATE UNIVERSITY                           00002000
C                                                         00002100
C    DISSERTATION ADVISER: DR. M. PALMER TERRELL         00002200
C                                                         00002300
C    VERSION 1 -- AUGUST, 1984                            00002400
C*****00002500
C                                00002600
C*****00002700
C                                00002800
C    C*** GENERAL STRUCTURE AND INPUT REQUIREMENTS:      00002900
C    (MAIN PROGRAM DRIVES THE SUBROUTINES UNFRM AND GLOBE) 00003000
C    (UNFRM DRIVES SUBROUTINE RANDU)                     00003100
C    (GLOBE DRIVES SUBROUTINE ALOC)                       00003200
C    (ALOC DRIVES SUBROUTINE PARTL)                      00003300
C                                                         00003400
C*****00003500
C                                00003600
C    SUBROUTINE                FUNCTION                  00003700
C    -----                -----                    00003800
C    RANDU                      TO GENERATE A UNIFORMLY DISTRIBUTED 00003900
C                                RANDOM NUMBER BETWEEN 0 AND 1      00004000
C    UNFRM                      TO CONVERT THE RANDOM NUMBER GENERATED 00004100
C                                BY RANDU BETWEEN APPROPRIATE LIMITS 00004200
C                                ESTABLISHED FOR THE UNIFORM        00004300
C                                DISTRIBUTION TO BE USED            00004400
C    GLOBE                      PERFORMS THE GLOBAL TESTS FOR OPTIMAL 00004500
C                                PROFITS AND FEASIBLE CAPACITY      00004600
C    ALOC                      PERFORMS THE FEASIBLE ALLOCATIONS FOR 00004700
C                                FOR EACH OF THE DEMAND CENTERS     00004800
C                                DICTATED BY PRIORITY RULE 1       00004900
C    PARTL                      PERFORMS THE PARTIAL TESTS FOR OPTIMAL 00005000
C                                PROFITS AND FEASIBLE CAPACITY      00005100
C                                00005200
C*****00005300
C                                00005400
C    C*** EXTERNAL FUNCTIONS REQUIRED                      00005500
C    (1) REGULAR SYSTEM SUPPLIED FORTRAN FUNCTIONS      00005600
C    (2) TWO IMSL SUBROUTINES                            00005700
C    MDNOR -- CUMULATIVE PROBABILITY FUNCTION OF        00005800
C    STANDARD NORMAL                                    00005900
C    MDNRIS -- INVERSE FUNCTION OF MDNOR                00006000
C                                                         00006100
C*****00006200
C                                00006300
C    C*** COMMON BLOCK VARIABLE DEFINITIONS             00006400
C    ONLY THOSE VARIABLES THAT REQUIRE EXPLANATION ARE LISTED 00006500
C                                                         00006600
C    /BLOCK1/                                           00006700
C                                                         00006800
C    TOFIX - TOTAL FIXED COST FOR THE ORIGINAL LOCATION VECTOR 00006900
C    UNDER CONSIDERATION                               00007000
C    TFIX -- TOTAL FIXED COST FOR THE LOCATION VECTOR PRESENTLY 00007100

```

C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007200
C		FROM THE ORIGINAL LOCATION VECTOR	00007300
C	TFFIX	- TOTAL FIXED COST FOR THE FINAL LOCATION VECTOR	00007400
C		CURRENTLY STORED	00007500
C	TRORG	- TOTAL REVENUE FOR THE ORIGINAL LOCATION VECTOR	00007600
C		UNDER CONSIDERATION	00007700
C	TREV	- TOTAL REVENUE FOR THE LOCATION VECTOR PRESENTLY	00007800
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007900
C		FROM THE ORIGINAL LOCATION VECTOR	00008000
C	TRFIN	- TOTAL REVENUE FOR THE FINAL LOCATION VECTOR	00008100
C		CURRENTLY STORED	00008200
C	TPORG	- TOTAL PROFIT FOR THE ORIGINAL LOCATION VECTOR	00008300
C		UNDER CONSIDERATION	00008400
C	TPROF	- TOTAL PROFIT FOR THE LOCATION VECTOR PRESENTLY	00008500
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008600
C		FROM THE ORIGINAL LOCATION VECTOR	00008700
C	TPFIN	- TOTAL PROFIT FOR THE FINAL LOCATION VECTOR	00008800
C		CURRENTLY STORED	00008900
C			00009000
C	/BLOCK2/		00009100
C			00009200
C	I Vect1-	INDICATOR VARIABLE FOR THE TEST ON FEASIBLE CAPACITY	00009300
C	I Vect2-	INDICATOR VARIABLE FOR THE TEST ON OPTIMAL PROFITS	00009400
C	NORG	- VARIABLE INDICATING THAT THE ORIGINAL FEASIBLE	00009500
C		SOLUTION HAS BEEN FOUND	00009600
C	NCON	- TOTAL NUMBER OF PLANTS DROPPED SO FAR	00009700
C	NALOC	- TOTAL NUMBER OF DEMAND CENTERS ALLOCATED SO FAR	00009800
C	MPT	- TOTAL NUMBER OF PLANTS FOR THE PROBLEM SOLVED	00009900
C	NCT	- TOTAL NUMBER OF DEMAND CENTERS FOR THE PROBLEM SOLVED	00010000
C			00010100
C	/BLOCK3/		00010200
C			00010300
C	NYD	- THE ORIGINAL LOCATION VECTOR UNDER CONSIDERATION	00010400
C	NY	- THE LOCATION VECTOR PRESENTLY CONSIDERED BY	00010500
C		DROPPING ONE PLANT AT A TIME FROM THE	00010600
C		ORIGINAL LOCATION VECTOR	00010700
C	NYF	- THE FINAL LOCATION VECTOR CURRENTLY STORED	00010800
C	NPLANT-	VARIABLE USED FOR THE PLANTS IN RANKING THE OPTIMAL	00010900
C		PROFITS EMPLOYING PRIORITY RULE 1	00011000
C	NCENT	- VARIABLE USED FOR THE DEMAND CENTERS IN RANKING THE	00011100
C		OPTIMAL PROFITS EMPLOYING PRIORITY RULE 1	00011200
C	ICENT	- VARIABLE INDICATING WHETHER A DEMAND CENTER HAS	00011300
C		ALREADY BEEN ALLOCATED	00011400
C	IDROP	- VARIABLE INDICATING THE PLANT PRESENTLY DROPPED	00011500
C		FROM THE ORIGINAL LOCATION VECTOR	00011600
C	NDROP	- VARIABLE USED FOR RANKING THE PLANTS FOR DROP	00011700
C		EMPLOYING PRIORITY RULE 2	00011800
C			00011900
C	/BLOCK4/		00012000
C			00012100
C	XSTAR	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR EACH	00012200
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00012300
C	XO	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE ORIGINAL	00012400
C		LOCATION VECTOR UNDER CONSIDERATION	00012500
C	X	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE LOCATION	00012600
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00012700
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00012800
C	XF	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE FINAL	00012900
C		LOCATION VECTOR CURRENTLY STORED	00013000
C	PR	- OPTIMAL PROFITS DETERMINED FOR EACH COMBINATION OF	00013100
C		DEMAND CENTERS AND PROFITS	00013200
C	PROF	- THE RANKED OPTIMAL PROFITS DETERMINED FOR EACH	00013300
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00013400
C		USING PRIORITY RULE 1	00013500
C			00013600
C	/BLOCK5/		00013700
C			00013800
C	NZO	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE ORIGINAL	00013900
C		LOCATION VECTOR UNDER CONSIDERATION	00014000
C	NZ	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE LOCATION	00014100
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00014200

```

C           AT A TIME FROM THE ORIGINAL LOCATION VECTOR           00014300
C     NZF - COMPONENTS OF THE ALLOCATION MATRIX FOR THE FINAL     00014400
C           LOCATION VECTOR CURRENTLY STORED                     00014500
C                                                                 00014600
C /BLOCK6/                                                       00014700
C                                                                 00014800
C     A - CAPACITY OF EACH PLANT                                00014900
C     F - FIXED COST OF EACH PLANT                             00015000
C     ADROP- THE REMAINING CAPACITY OF EACH PLANT AFTER HAVING 00015100
C           ALLOCATED A SUBSET OF DEMAND CENTERS                00015200
C                                                                 00015300
C*****00015400
C                                                                 00015500
C*** OTHER VARIABLE DEFINITIONS                                00015600
C                                                                 00015700
C     MU - MEAN OF THE NORMAL DISTRIBUTION OF DEMAND           00015800
C           DETERMINED FOR EACH COMBINATION OF                 00015900
C           DEMAND CENTERS AND PLANTS                          00016000
C     SIGMA- STANDARD DEVIATION OF THE NORMAL DISTRIBUTION OF 00016100
C           DEMAND DETERMINED FOR EACH COMBINATION OF         00016200
C           DEMAND CENTERS AND PLANTS                          00016300
C     R - PER UNIT VARIABLE COST OF THE PRODUCT SUPPLIED FOR 00016400
C           EACH COMBINATION OF DEMAND CENTERS AND PLANTS     00016500
C     PC - PER UNIT PRICE OF THE PRODUCT RECEIVED FOR EACH    00016600
C           COMBINATION OF DEMAND CENTERS AND PLANTS          00016700
C                                                                 00016800
C*****00016900
C     INTEGER RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,        00017000
C     *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00017100
C     *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)    00017200
C     REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00017300
C     *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00017400
C     *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,   00017500
C     *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00017600
C     *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00017700
C     *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),    00017800
C     *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00017900
C     DATA RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS/2HR0,2HMA, 00018000
C     *2HFI,2HE0,1HO,3HPR0,2HFT,3HEQU,3HINE,1HZ,1HY,1H+/    00018100
C     COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00018200
C     *TPFIN,TPROF                                           00018300
C     COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO, 00018400
C     *KK,JJ                                                 00018500
C     COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00018600
C     COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF                   00018700
C     COMMON/BLOCK5/NZO,NZ,NZF                              00018800
C     COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS          00018900
C     COMMON/BLOCK7/B,C,D                                   00019000
C     COMMON/BLOCK8/IX,IY,YFL                               00019100
C*****00019200
C     NO=12                                                  00019300
C     NOT=14                                                 00019400
C     MPT=3                                                  00019500
C     NCT=4                                                  00019600
C     CONST1=1.0                                            00019700
C     CONST2=0.0                                            00019800
C     MNTOT=MPT*NCT                                         00019900
C     MQ=NCT+MPT                                           00020000
C     MR=MPT*NCT+MPT                                       00020100
C*****00020200
C     GENERATE VALUES RANDOMLY FOR F(MM)                  00020300
C     SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM 00020400
C     DISTRIBUTION                                          00020500
C*****00020600
C     B=200.                                                00020700
C     C=300.                                                00020800
C     IX=13                                                 00020900
C     WRITE(NO,4025)                                        00021000
C     4025 FORMAT(//5X,'THE PLANT FIXED COSTS ARE',//)    00021100
C     WRITE(NO,5248)                                        00021200
C     5248 FORMAT(5X,'FIXED COST',5X,'PLANT'//)           00021300

```

```

DO 1001 MM=1,MPT
CALL UNFRM
F(MM)=D
WRITE(NO,5249) D,MM
5249 FORMAT(1X,F14.6,7X,I2//)
IX=IY
1001 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR A(MM)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=20000.
      C=25000.
      WRITE(NO,4026)
4026 FORMAT(//5X,'THE PLANT CAPACITIES ARE'//)
      WRITE(NO,5250)
5250 FORMAT(5X,'CAPACITY',5X,'PLANT'//)
DO 1002 MM=1,MPT
CALL UNFRM
A(MM)=D
WRITE(NO,5251) D,MM
5251 FORMAT(1X,F14.6,5X,I2//)
IX=IY
1002 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR R(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=0.5
      C=2.5
      WRITE(NO,4027)
4027 FORMAT(//5X,'THE PER UNIT VARIABLE COSTS ARE'//)
      WRITE(NO,5252)
5252 FORMAT(1X,'PER UNIT VARIABLE COST',5X,'PLANT',5X,'DEMAND CENTER',
*/)
DO 1003 MM=1,MPT
DO 1004 LL=1,NCT
CALL UNFRM
R(MM,LL)=D
WRITE(NO,5253) D,MM,LL
5253 FORMAT(5X,F14.6,9X,I2,13X,I2//)
1004 IX=IY
1003 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR PC(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=3.
      C=6.
      WRITE(NO,4028)
4028 FORMAT(//1X,'THE PER UNIT PRICE AND THE MEAN AND STANDARD DEVIATIO
*N OF THE DEMAND FOR THE PRODUCTS ARE'//)
      WRITE(NO,5254)
5254 FORMAT(4X,'PER UNIT PRICE',6X,'MEAN DEMAND',5X,'STANDARD DEVIATION
*',5X,'PLANT',5X,'DEMAND CENTER'//)
DO 1005 MM=1,MPT
DO 1006 LL=1,NCT
CALL UNFRM
PC(MM,LL)=D
MU(MM,LL)=600./PC(MM,LL)
SIGMA(MM,LL)=10.0
WRITE(NO,5255) D,MU(MM,LL),SIGMA(MM,LL),MM,LL
5255 FORMAT(1X,F14.6,5X,F14.6,5X,F14.6,11X,I2,11X,I2//)
1006 IX=IY
1005 CONTINUE
C*****
C COMPUTE THE VALUES OF THE OPTIMAL SUPPLY QUANTITIES XSTAR(MM,LL)
C AND THE OPTIMAL PROFITS PR(MM,LL)

```



```

C*****
DO 900 MM=1,MPT
DO 901 LL=1,NCT
P=1.-R(MM,LL)/PC(MM,LL)
CALL MDNRIS(P,S,IER)
XSTAR(MM,LL)=S*SIGMA(MM,LL)+MU(MM,LL)
IF(XSTAR(MM,LL)-O.) 902,902,903
903 C1=PC(MM,LL)*MU(MM,LL)
TWOPI=2.*3.141593
FAC1=SQRT(TWOPI)
FAC2=1./FAC1
FAC3=XSTAR(MM,LL)-MU(MM,LL)
FAC4=FAC3/SIGMA(MM,LL)
FAC5=- (FAC4**2)/2.
FAC6=EXP(FAC5)
FAC7=-PC(MM,LL)*SIGMA(MM,LL)
C2=FAC7*FAC2*FAC6
S=FAC4
CALL MDNOR(S,P)
FAC8=1.-P
C3=PC(MM,LL)*FAC3*FAC8
C4=-R(MM,LL)*XSTAR(MM,LL)
PR(MM,LL)=C1+C2+C3+C4
IF(PR(MM,LL)-O.) 902,902,901
902 XSTAR(MM,LL)=O.
PR(MM,LL)=O.
901 CONTINUE
900 CONTINUE
WRITE(NO,4029)
4029 FORMAT(/5X,'THE OPTIMAL SUPPLY QUANTITIES AND THE PROFITS ARE',//00031400
*)
WRITE(NO,5256)
5256 FORMAT(1X,'OPTIMAL SUPPLY QUANTITY',8X,'PROFIT',8X,'PLANT',5X,
*'DEMAND CENTER'//)
DO 3003 MM=1,MPT
DO 3004 LL=1,NCT
3004 WRITE(NO,5257) XSTAR(MM,LL),PR(MM,LL),MM,LL
5257 FORMAT(4X,F14.6,9X,F14.6,6X,I2,13X,I2//)
3003 CONTINUE
C*****
C GENERATE THE INPUT DATA DECK FOR THE
C EXTREME POINT RANKING TECHNIQUE
C*****
WRITE(NOT,137) MQ,MR
137 FORMAT(2I5)
WRITE(NOT,138) RD
138 FORMAT(1X,A2)
WRITE(NOT,139) ZERO,PRO,FT
139 FORMAT(11X,A1,1X,A3,A2)
DO 141 LL=1,NCT
IF(LL-9) 142,142,143
142 WRITE(NOT,144) ZERO,EQU,ZERO,LL
144 FORMAT(11X,A1,1X,A3,A1,I1)
GO TO 141
143 WRITE(NOT,145) ZERO,EQU,LL
145 FORMAT(11X,A1,1X,A3,I2)
141 CONTINUE
DO 146 MM=1,MPT
IF(MM-9) 147,147,148
147 WRITE(NOT,144) POS,INE,ZERO,MM
GO TO 146
148 WRITE(NOT,145) POS,INE,MM
146 CONTINUE
WRITE(NOT,149) MA
149 FORMAT(1X,A2)
DO 150 LL=1,NCT
DO 151 MM=1,MPT
IF(MM-9) 152,152,153
152 IF(LL-9) 154,154,155
154 WRITE(NOT,733) Z,ZERO,MM,ZERO,LL,PRO,FT,PR(MM,LL)
733 FORMAT(7X,A1,A1,I1,A1,I1,1X,A3,A2,1X,D20.10)
00028500
00028600
00028700
00028800
00028900
00029000
00029100
00029200
00029300
00029400
00029500
00029600
00029700
00029800
00029900
00030000
00030100
00030200
00030300
00030400
00030500
00030600
00030700
00030800
00030900
00031000
00031100
00031200
00031300
00031400
00031500
00031600
00031700
00031800
00031900
00032000
00032100
00032200
00032300
00032400
00032500
00032600
00032700
00032800
00032900
00033000
00033100
00033200
00033300
00033400
00033500
00033600
00033700
00033800
00033900
00034000
00034100
00034200
00034300
00034400
00034500
00034600
00034700
00034800
00034900
00035000
00035100
00035200
00035300
00035400
00035500

```

```

WRITE(NOT,734) Z,ZERO,MM,ZERO,LL,EQU,ZERO,LL,CONST1      00035600
734  FORMAT(7X,A1,A1,I1,A1,I1,1X,A3,A1,I1,1X,D20.10)      00035700
WRITE(NOT,734) Z,ZERO,MM,ZERO,LL,INE,ZERO,MM,XSTAR(MM,LL) 00035800
      GO TO 151                                           00035900
155  WRITE(NOT,736) Z,ZERO,MM,LL,PRO,FT,PR(MM,LL)        00036000
736  FORMAT(7X,A1,A1,I1,I2,1X,A3,A2,1X,D20.10)          00036100
WRITE(NOT,737) Z,ZERO,MM,LL,EQU,LL,CONST1                00036200
737  FORMAT(7X,A1,A1,I1,I2,1X,A3,I2,1X,D20.10)          00036300
WRITE(NOT,738) Z,ZERO,MM,LL,INE,ZERO,MM,XSTAR(MM,LL)    00036400
738  FORMAT(7X,A1,A1,I1,I2,1X,A3,A1,I1,1X,D20.10)      00036500
      GO TO 151                                           00036600
153  IF(LL-9) 157,157,158                                00036700
157  WRITE(NOT,739) Z,MM,ZERO,LL,PRO,FT,PR(MM,LL)        00036800
739  FORMAT(7X,A1,I2,A1,I1,1X,A3,A2,1X,D20.10)          00036900
WRITE(NOT,740) Z,MM,ZERO,LL,EQU,ZERO,LL,CONST1          00037000
740  FORMAT(7X,A1,I2,A1,I1,1X,A3,A1,I1,1X,D20.10)      00037100
WRITE(NOT,756) Z,MM,ZERO,LL,INE,MM,XSTAR(MM,LL)         00037200
756  FORMAT(7X,A1,I2,A1,I1,1X,A3,I2,1X,D20.10)          00037300
      GO TO 151                                           00037400
158  WRITE(NOT,742) Z,MM,LL,PRO,FT,PR(MM,LL)            00037500
742  FORMAT(7X,A1,I2,I2,1X,A3,A2,1X,D20.10)            00037600
WRITE(NOT,743) Z,MM,LL,EQU,LL,CONST1                    00037700
743  FORMAT(7X,A1,I2,I2,1X,A3,I2,1X,D20.10)            00037800
WRITE(NOT,743) Z,MM,LL,INE,MM,XSTAR(MM,LL)              00037900
151  CONTINUE                                           00038000
150  CONTINUE                                           00038100
      DO 159 MM=1,MPT                                     00038200
          FMS(MM)=-F(MM)                                  00038300
          AMS(MM)=-A(MM)                                  00038400
          IF(MM-9) 160,160,161                            00038500
160  WRITE(NOT,745) Y,ZERO,ZERO,ZERO,MM,PRO,FT,FMS(MM)   00038600
WRITE(NOT,746) Y,ZERO,ZERO,ZERO,MM,INE,ZERO,MM,AMS(MM)  00038700
      GO TO 159                                           00038800
161  WRITE(NOT,747) Y,ZERO,ZERO,MM,PRO,FT,FMS(MM)       00038900
WRITE(NOT,748) Y,ZERO,ZERO,MM,INE,MM,AMS(MM)            00039000
745  FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A2,1X,D20.10)      00039100
746  FORMAT(7X,A1,A1,A1,A1,A1,I1,1X,A3,A1,I1,1X,D20.10) 00039200
747  FORMAT(7X,A1,A1,A1,I2,1X,A3,A2,1X,D20.10)          00039300
748  FORMAT(7X,A1,A1,A1,I2,1X,A3,I2,1X,D20.10)          00039400
159  CONTINUE                                           00039500
WRITE(NOT,749) FI                                         00039600
749  FORMAT(1X,A2)                                         00039700
      DO 162 LL=1,NCT                                     00039800
          IF(LL-9) 163,163,164                            00039900
163  WRITE(NOT,755) EQU,ZERO,LL,CONST1                    00040000
      GO TO 162                                           00040100
164  WRITE(NOT,750) EQU,LL,CONST1                         00040200
755  FORMAT(13X,A3,A1,I1,1X,D20.10)                      00040300
750  FORMAT(13X,A3,I2,1X,D20.10)                          00040400
162  CONTINUE                                           00040500
      DO 165 MM=1,MPT                                     00040600
          IF(MM-9) 166,166,167                            00040700
166  WRITE(NOT,751) INE,ZERO,MM,CONST2                    00040800
      GO TO 165                                           00040900
167  WRITE(NOT,752) INE,MM,CONST2                         00041000
751  FORMAT(13X,A3,A1,I1,1X,D20.10)                      00041100
752  FORMAT(13X,A3,I2,1X,D20.10)                          00041200
165  CONTINUE                                           00041300
WRITE(NOT,753) EO                                         00041400
753  FORMAT(1X,A2)                                         00041500
C*****
C SORT THE PROFITS IN A DECENDING ORDER
C*****
      K=0
      KONST=0
      DO 7 I=1,MPT
          DO 10 J=1,NCT
              K=KONST+J
              PROF(K)=PR(I,J)
              NPLANT(K)=I
              NCENT(K)=J

```

```

10 CONTINUE                                00042700
    KONST=I*NCT                            00042800
7 CONTINUE                                00042900
    MN=1                                    00043000
15 PROFM=PROF(MN)                          00043100
    NPLAM=NPLANT(MN)                      00043200
    NCENTM=NCENT(MN)                      00043300
DO 20 I=MN,MNTOT                          00043400
    IF (PROFM-PROF(I)) 25,25,20           00043500
25 PROFM=PROF(I)                          00043600
    NPLAM=NPLANT(I)                      00043700
    NCENTM=NCENT(I)                      00043800
    IMAX=I                                 00043900
20 CONTINUE                                00044000
    PROF(IMAX)=PROF(MN)                   00044100
    NPLANT(IMAX)=NPLANT(MN)               00044200
    NCENT(IMAX)=NCENT(MN)                 00044300
    PROF(MN)=PROFM                        00044400
    NPLANT(MN)=NPLAM                     00044500
    NCENT(MN)=NCENTM                     00044600
    MN=MN+1                               00044700
    IF (MN-MNTOT) 15,15,35                00044800
35 WRITE(NO,4030)                          00044900
4030 FORMAT(5X,'THE SORTED PROFITS AND THE RESPECTIVE PLANTS AND THE DE00045000
*MAND CENTERS ARE'//)                    00045100
    WRITE(NO,4034)                        00045200
4034 FORMAT(5X,'RANKED PROFIT',10X,'PLANT',10X,'DEMAND CENTER'//) 00045300
    DO 5033 I=1,MNTOT                    00045400
5033 WRITE(NO,4035) PROF(I),NPLANT(I),NCENT(I) 00045500
4035 FORMAT(3X,F14.6,12X,I2,17X,I2//)    00045600
C*****                                00045700
C SORT THE PLANTS IN A DECENDING ORDER OF FIXED COSTS 00045800
C*****                                00045900
    DO 333 I=1,MPT                        00046000
        FDROP(I)=F(I)                     00046100
        ADROP(I)=A(I)                     00046200
333 NDROP(I)=I                            00046300
    MJ=1                                   00046400
37 FDROPM=FDROP(MJ)                       00046500
    ADROPM=ADROP(MJ)                     00046600
    NDROPM=NDROP(MJ)                     00046700
    DO 40 I=MJ,MPT                        00046800
        IF (FDROPM-FDROP(I)) 45,45,40    00046900
45 FDROPM=FDROP(I)                       00047000
    ADROPM=ADROP(I)                     00047100
    NDROPM=NDROP(I)                     00047200
    IMAX=I                                00047300
40 CONTINUE                                00047400
    FDROP(IMAX)=FDROP(MJ)                 00047500
    ADROP(IMAX)=ADROP(MJ)                 00047600
    NDROP(IMAX)=NDROP(MJ)                 00047700
    FDROP(MJ)=FDROPM                     00047800
    ADROP(MJ)=ADROPM                     00047900
    NDROP(MJ)=NDROPM                     00048000
    IDROP(MJ)=1                           00048100
    MJ=MJ+1                               00048200
    IF (MJ-MPT) 37,37,50                  00048300
50 WRITE(NO,4031)                          00048400
4031 FORMAT(5X,'THE SORTED FIXED COSTS AND OTHER RESPECTIVE PARAMETERS 00048500
*ARE',//)                                00048600
    WRITE(NO,4032)                        00048700
4032 FORMAT(1X,'RANKED PLANTS FOR DROP',10X,'FIXED COST',11X,'CAPACITY'00048800
*,11X,'FIXED COST',10X,'CAPACITY',12X,'IDROP VALUES'//) 00048900
    DO 5034 I=1,MPT                      00049000
5034 WRITE(NO,4033) NDROP(I),FDROP(I),ADROP(I),F(NDROP(I)), 00049100
    *A(NDROP(I)),IDROP(I)                00049200
4033 FORMAT(10X,I2,17X,F14.6,6X,F14.6,6X,F14.6,6X,F14.6,10X,I7//) 00049300
C*****                                00049400
C START WITH ALL PLANTS OPEN             00049500
C SET ALL NY(I) EQUAL TO 1               00049600
C*****                                00049700

```

```

DO 55 K=1,MPT                                00049800
55 NY(NDROP(K))=1                             00049900
C*****                                     00050000
C INITIALIZE ALL PARAMETERS                   00050100
C*****                                     00050200
      TOFIX=0.0                               00050300
      TFFIX=0.0                               00050400
      TRORG=0.0                               00050500
      TRFIN=0.0                               00050600
      TPORG=0.0                               00050700
      TPFIN=0.0                               00050800
      IVECT1=1                                00050900
      IVECT2=1                                00051000
      NORG=0                                  00051100
      NCON=0                                  00051200
DO 610 MM=1,MPT                              00051300
      NYD(NDROP(MM))=O                       00051400
      NYF(NDROP(MM))=O                       00051500
      AF(NDROP(MM))=O.O                     00051600
      FF(NDROP(MM))=O.O                     00051700
DO 611 LL=1,NCT                              00051800
      XO(NDROP(MM),LL)=O.O                  00051900
      XF(NDROP(MM),LL)=O.O                  00052000
      X(NDROP(MM),LL)=O.O                   00052100
      NZO(NDROP(MM),LL)=O                   00052200
      NZF(NDROP(MM),LL)=O                   00052300
611 NZ(NDROP(MM),LL)=O                       00052400
610 CONTINUE                                 00052500
      DO 477 LL=1,NCT                        00052600
477 ICENT(LL)=O                               00052700
C*****                                     00052800
C CALL THE SUBROUTINE GLOBE TO PERFORM THE GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY      00052900
C*****                                     00053000
      CALL GLOBE                             00053100
      IF(IVECT1-O) 325,323,325                00053200
323 WRITE(NO,103)                             00053300
103 FORMAT(1X,'FOR THE HEURISTICS EMPLOYED NO FEASIBLE SOLUTION CAN BE
*FOUND DUE TO CAPACITY LIMITATIONS'//)      00053400
999 STOP                                       00053500
5074 WRITE(NO,5075)                           00053600
5075 FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS - STOP'//
*****'//)                                  00053700
      GO TO 999                               00053800
325 IF(NORG-O) 327,328,327                    00053900
328 IVECT2=1                                  00054000
C*****                                     00054100
C CONSIDER THE DROP OF EACH PLANT            00054200
C*****                                     00054300
327 DO 520 KK=1,MPT                           00054400
      IF(IDROP(KK)-1) 520,530,520            00054500
530 NY(NDROP(KK))=O                           00054600
      FDROP(KK)=O.O                          00054700
      ADROP(KK)=O.O                          00054800
      CALL GLOBE                             00054900
      IF(IVECT1-O) 535,540,535                00055000
540 IVECT1=1                                  00055100
      IDROP(KK)=O                            00055200
      NCON=NCON+1                            00055300
      IF(NCON-MPT) 560,435,435                00055400
535 IF(IVECT2-O) 550,555,550                  00055500
555 IVECT2=1                                  00055600
      IF(TPORG-TPROF) 560,565,565            00055700
565 DIF=TPORG-TPROF                           00055800
      IF(DIF-F(NDROP(KK))) 560,560,575      00055900
575 IDROP(KK)=O                              00056000
      NCON=NCON+1                            00056100
      IF(NCON-MPT) 560,435,435                00056200
550 IF(TPORG-TPROF) 585,696,696              00056300
696 IDN=1                                      00056400
      GO TO 693                              00056500

```

```

585     IF(TPFIN-TPROF) 590,693,693                00056900
590     TFFIX=TFIX                                00057000
        TRFIN=TREV                                00057100
        TPFIN=TPROF                                00057200
        NN=KK                                       00057300
        DO 592 MM=1,MPT                             00057400
            NYF(NDROP(MM))=NY(NDROP(MM))           00057500
            IF(NYF(NDROP(MM))-1) 783,784,783       00057600
784     AF(NDROP(MM))=A(NDROP(MM))                00057700
            FF(NDROP(MM))=F(NDROP(MM))             00057800
            GO TO 997                                00057900
783     AF(NDROP(MM))=0.0                          00058000
            FF(NDROP(MM))=0.0                      00058100
997     DO 593 LL=1,NCT                             00058200
            XF(NDROP(MM),LL)=X(NDROP(MM),LL)       00058300
593     NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)         00058400
592     CONTINUE                                    00058500
        WRITE(NO,8889)                              00058600
8889    FORMAT(//5X,'THE INTERMEDIATE FINAL SOLUTION HAS BEEN FOUND'//) 00058700
        WRITE(NO,2927)                              00058800
2927    FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X, 00058900
        *,'CAPACITY',/1X,'=====',8X,'=====',5X,'=====', 00059000
        *5X,'=====',//)                            00059100
        DO 2928 MM=1,MPT                             00059200
2928    WRITE(NO,2929) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)), 00059300
        *AF(NDROP(MM))                              00059400
2929    FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)    00059500
        WRITE(NO,2931)                              00059600
2931    FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X 00059700
        *,'PROFIT',/1X,'=====',5X,'=====',8X,'===== 00059800
        *,12X,'=====',//)                          00059900
        DO 2422 LL=1,NCT                             00060000
        DO 2423 MM=1,MPT                             00060100
            IF(NZF(NDROP(MM),LL)-1) 2423,2424,2423 00060200
2424    WRITE(NO,2425) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00060300
2423    CONTINUE                                    00060400
2422    CONTINUE                                    00060500
        WRITE(NO,3379) TPFIN                         00060600
3379    FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//) 00060700
2425    FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//) 00060800
693     DO 694 MM=1,MPT                             00060900
            IF(NY(NDROP(MM))-1) 694,699,694       00061000
699     ADROP(MM)=A(NDROP(MM))                    00061100
            DO 695 LL=1,NCT                         00061200
                X(NDROP(MM),LL)=0.0                00061300
695     NZ(NDROP(MM),LL)=0                        00061400
694     CONTINUE                                    00061500
            IF(IDN-1) 560,565,560                  00061600
560     NY(NDROP(KK))=1                            00061700
            FDROP(KK)=F(NDROP(KK))                 00061800
            ADROP(KK)=A(NDROP(KK))                 00061900
            IF(IDN-1) 520,562,520                  00062000
562     IDN=0                                       00062100
520     CONTINUE                                    00062200
C*****                                           00062300
C SET ORIGINAL PARAMETERS EQUAL TO THE FINAL PARAMETERS AND 00062400
C THE INTERMEDIATE PARAMETERS AS APPROPRIATE 00062500
C*****                                           00062600
        TOFIX=TFFIX                                00062700
        TRORG=TRFIN                                00062800
        TPORG=TPFIN                                00062900
        DO 801 MM=1,MPT                             00063000
            NYD(NDROP(MM))=NYF(NDROP(MM))           00063100
            NY(NDROP(MM))=NYF(NDROP(MM))           00063200
            IF(NN-MM) 803,802,803                  00063300
802     IDROP(MM)=0                                00063400
            NCON=NCON+1                             00063500
            IF(NCON-MPT) 803,435,435               00063600
803     IF(NY(NDROP(MM))-1) 804,805,804           00063700
805     ADROP(MM)=A(NDROP(MM))                    00063800
            FDROP(MM)=F(NDROP(MM))                 00063900

```

```

      GO TO 806
804  ADROP(MM)=0.0
      FDROP(MM)=0.0
806  DO 807 LL=1,NCT
      XO(NDROP(MM),LL)=XF(NDROP(MM),LL)
807  NZD(NDROP(MM),LL)=NZF(NDROP(MM),LL)
801  CONTINUE
      DO 741 LL=1,NCT
741  ICENT(LL)=0
      GO TO 327
C*****
C PRINT ALL FINAL VALUES
C*****
435  IF(NORG=0) 493,5074,493
493  WRITE(NOT,967) TPFIN
967  FORMAT(D20.10)
      WRITE(NO,8883)
8883 FORMAT(/5X,'THE FINAL SOLUTION HAS BEEN FOUND'//)
      WRITE(NO,2001)
2001 FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'====',8X,'=====',5X,'=====',
* 5X,'=====',//)
      DO 2002 MM=1,MPT
2002 WRITE(NO,2003) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
* AF(NDROP(MM))
2003 FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)
      WRITE(NO,2004)
2004 FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
* ,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
* ,12X,'=====',//)
      DO 2005 LL=1,NCT
      DO 2006 MM=1,MPT
      IF(NZF(NDROP(MM),LL)-1) 2006,2023,2006
2023 WRITE(NO,2007) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2006 CONTINUE
2005 CONTINUE
2007 FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
      WRITE(NO,3379) TPFIN
      STOP
      END
C*****
C SUBROUTINE TO PERFORM GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
      SUBROUTINE GLOBE
      INTEGER RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
* NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
* IDROP(40),NDROP(40),NZD(40,50),NZ(40,50),NZF(40,50)
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
* FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM,
* SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
* TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
* SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
* XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
* A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
* TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO,
* KK,JJ
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
      TCAP=0.0
      TRGLBL=0.0
      DO 70 LL=1,NCT
      PRMAX=0.0
      CAPMIN=1.0D20
      DO 75 MM=1,MPT
      IF(NY(NDROP(MM))-1) 75,80,75
80  IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 75,85,85

```

```

85     IF(CAPMIN-XSTAR(NDROP(MM),LL)) 90,90,95          00071100
95     CAPMIN=XSTAR(NDROP(MM),LL)                      00071200
90     IF(PRMAX-PR(NDROP(MM),LL)) 100,75,75           00071300
100    PRMAX=PR(NDROP(MM),LL)                          00071400
75     CONTINUE                                        00071500
      TCAP=TCAP+CAPMIN                                00071600
      TRGLBL=TRGLBL+PRMAX                             00071700
70     CONTINUE                                        00071800
      SUMC=0.0                                         00071900
      TFIX=0.0                                         00072000
      DO 105 MM=1,MPT                                  00072100
        IF(NY(NDROP(MM))-1) 105,110,105              00072200
110    SUMC=SUMC+ADROP(MM)                             00072300
        TFIX=TFIX+FDROP(MM)                           00072400
105    CONTINUE                                        00072500
      TPGLBL=TRGLBL-TFIX                               00072600
      IF(SUMC-TCAP) 115,120,120                       00072700
115    IVECT1=0                                         00072800
      WRITE(NO,5888)                                    00072900
5888   FORMAT(5X,'THE GLOBAL TEST FOR FEASIBLE CAPACITY FAILED'///) 00073000
      GO TO 133                                         00073100
120    IF(TPORG-TPGLBL) 125,117,117                  00073200
117    IF(NORG=0) 118,119,118                        00073300
118    TPROF=TPGLBL                                    00073400
119    IVECT2=0                                         00073500
      WRITE(NO,5889)                                    00073600
5889   FORMAT(5X,'THE GLOBAL TEST FOR OPTIMAL PROFITS FAILED'///) 00073700
      GO TO 133                                         00073800
125   WRITE(NO,5890)                                    00073900
5890   FORMAT(5X,'PASSED BOTH GLOBAL TESTS FOR FEASIBLE CAPACITY AND OPTI 00074000
      *MAL PROFITS'///)                                00074100
C*****                                                00074200
C CALL THE SUBROUTINE ALOC TO PERFORM FEASIBLE ALLOCATIONS 00074300
C*****                                                00074400
      CALL ALOC                                        00074500
133   RETURN                                           00074600
      END                                             00074700
C*****                                                00074800
C SUBROUTINE TO PERFORM FEASIBLE ALLOCATIONS             00074900
C*****                                                00075000
      SUBROUTINE ALOC                                  00075100
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00075200
      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00075300
      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00075400
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00075500
      *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00075600
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00075700
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00075800
      *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00075900
      *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00076000
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00076100
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00076200
      *TPFIN,TPROF                                     00076300
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO, 00076400
      *KK,JJ                                           00076500
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00076600
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF             00076700
      COMMON/BLOCK5/NZO,NZ,NZF                       00076800
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00076900
      NALOC=0                                         00077000
      NTEST=1                                         00077100
      TREV=0.0                                         00077200
      DO 135 JJ=1,MNTOT                                00077300
        IF(ICENT(NCENT(JJ))-1) 7290,135,7290          00077400
7290   IF(NY(NPLANT(JJ))-1) 135,707,135              00077500
707   DO 708 IB=1,MPT                                  00077600
        IF(NPLANT(JJ)-NDROP(IB)) 708,140,708          00077700
708   CONTINUE                                        00077800
140   IF(ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ))) 135,150,150 00077900
150   X(NPLANT(JJ),NCENT(JJ))=XSTAR(NPLANT(JJ),NCENT(JJ)) 00078000
        NZ(NPLANT(JJ),NCENT(JJ))=1                   00078100

```

```

          ICENT(NCENT(JJ))=1                                00078200
          ADROP(IB)=ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ)) 00078300
          TREV=TREV+PROF(JJ)                                00078400
          NALOC=NALOC+1                                    00078500
          IF(NALOC-NCT) 152,157,157                        00078600
152      IF(AMOD(FLOAT(NALOC),FLOAT(NTEST))-O.O) 135,155,135 00078700
C*****
C CALL THE SUBROUTINE PARTL TO PERFORM PARTIAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
155      CALL PARTL
          IF(IVECT1-O) 132,137,132                          00079200
132      IF(IVECT2-O) 135,137,135                          00079300
135      CONTINUE
          IF(NALOC-NCT) 149,157,157                        00079400
157      TPROF=TREV-TFIX                                    00079500
          WRITE(NO,6237) TPROF                              00079600
6237     FORMAT(/5X,'THE TOTAL PROFIT FOR THIS CONFIGURATION IS= ',
          *F14.6//)
          IF(NORG-O) 151,139,151                            00079700
139      IF(TPORG-TPROF) 141,137,137                      00079800
141      TOFIX=TFIX                                        00079900
          TFFIX=TFIX                                        00080000
          TRORG=TREV                                       00080100
          TRFIN=TREV                                       00080200
          TPORG=TPROF                                       00080300
          TPFIN=TPROF                                       00080400
          DO 142 MM=1,MPT                                    00080500
            NYO(NDROP(MM))=NY(NDROP(MM))                   00080600
            NYF(NDROP(MM))=NY(NDROP(MM))                   00080700
            IF(NY(NDROP(MM))-1) 142,146,142                 00080800
146      ADROP(MM)=A(NDROP(MM))                             00080900
            AF(NDROP(MM))=A(NDROP(MM))                     00081000
            FF(NDROP(MM))=F(NDROP(MM))                     00081100
          DO 143 LL=1,NCT                                    00081200
            XO(NDROP(MM),LL)=X(NDROP(MM),LL)              00081300
            XF(NDROP(MM),LL)=X(NDROP(MM),LL)              00081400
            NZO(NDROP(MM),LL)=NZ(NDROP(MM),LL)            00081500
            NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)            00081600
            X(NDROP(MM),LL)=O.O                            00081700
143      NZ(NDROP(MM),LL)=O                                00081800
142      CONTINUE
          NORG=1                                           00081900
          WRITE(NO,5893)                                    00082000
5893     FORMAT(5X,'THE ORIGINAL FEASIBLE SOLUTION HAS BEEN FOUND'//) 00082100
4232     WRITE(NO,3022)                                    00082200
3022     FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
          *'CAPACITY',/1X,'=====',8X,'=====',5X,'=====',
          *5X,'=====',//)
          DO 3023 MM=1,MPT                                  00082300
3023     WRITE(NO,3024) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
          *AF(NDROP(MM))                                    00082400
3024     FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)          00082500
          WRITE(NO,3025)                                    00082600
3025     FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X,
          *,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
          *,12X,'=====',//)
          DO 3026 LL=1,NCT                                  00082700
          DO 3027 MM=1,MPT                                  00082800
            IF(NZF(NDROP(MM),LL)-1) 3027,3051,3027        00082900
3051     WRITE(NO,3028) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00083000
3027     CONTINUE
3026     CONTINUE
3028     FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)        00083100
          WRITE(NO,3380) TPFIN                              00083200
3380     FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)      00083300
          GO TO 151                                         00083400
149      IVECT1=O                                          00083500
137      DO 153 MM=1,MPT                                    00083600
            IF(NY(NDROP(MM))-1) 153,710,153                00083700
710      ADROP(MM)=A(NDROP(MM))                            00083800

```



```

DO 154 LL=1,NCT                                00085300
  X(NDROP(MM),LL)=0.0                          00085400
154 NZ(NDROP(MM),LL)=0                        00085500
153 CONTINUE                                   00085600
151 DO 738 LL=1,NCT                            00085700
738 ICENT(LL)=0                                00085800
  RETURN                                        00085900
  END                                            00086000
C*****                                         00086100
C SUBROUTINE TO PERFORM PARTIAL TESTS FOR      00086200
C OPTIMAL PROFITS AND FEASIBLE CAPACITY       00086300
C*****                                         00086400
  SUBROUTINE PARTL                              00086500
    INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00086600
    *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00086700
    *IDROP(40),NDROP(40),NZO(40.50),NZ(40.50),NZF(40.50) 00086800
    REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00086900
    *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROP,ADROP, 00087000
    *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00087100
    *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00087200
    *SIGMA(40.50),MU(40.50),R(40.50),PC(40.50),XSTAR(40.50),DIF, 00087300
    *XD(40.50),X(40.50),XF(40.50),PR(40.50),PROF(2000), 00087400
    *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00087500
    COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00087600
    *TPFIN,TPROF 00087700
    COMMON/BLOCK2/IVECT1,IVECT2,NDRG,NCON,NALOC,MPT,NCT,MNTOT,NO 00087800
    *,KK,JJ 00087900
    COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00088000
    COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00088100
    COMMON/BLOCK5/NZO,NZ,NZF 00088200
    COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00088300
    RCAP=0.0 00088400
    TRPTL=0.0 00088500
    DO 160 LL=1,NCT 00088600
      PRMAX=0.0 00088700
      CAPMIN=1.0D20 00088800
      IF(ICENT(LL)-1) 736,160,736 00088900
736 DO 165 MM=1,MPT 00089000
      IF(NY(NDROP(MM))-1) 165,175,165 00089100
175 IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 165,180,180 00089200
180 IF(CAPMIN-XSTAR(NDROP(MM),LL)) 185,185,190 00089300
190 CAPMIN=XSTAR(NDROP(MM),LL) 00089400
185 IF(PRMAX-PR(NDROP(MM),LL)) 195,165,165 00089500
195 PRMAX=PR(NDROP(MM),LL) 00089600
165 CONTINUE 00089700
      RCAP=RCAP+CAPMIN 00089800
      TRPTL=TRPTL+PRMAX 00089900
160 CONTINUE 00090000
      TRPTL=TREV+TRPTL 00090100
      SUMR=0.0 00090200
      DO 200 MM=1,MPT 00090300
        IF(NY(NDROP(MM))-1) 200,205,200 00090400
205 SUMR=SUMR+ADROP(MM) 00090500
200 CONTINUE 00090600
        TPPTL=TRPTL-TFIX 00090700
        IF(SUMR-RCAP) 210,215,215 00090800
210 IVECT1=0 00090900
        WRITE(NO,5891) 00091000
5891 FORMAT(5X,'THE PARTIAL TEST FOR FEASIBLE CAPACITY FAILED'///) 00091100
        GO TO 220 00091200
215 IF(TPORG-TPPTL) 220,225,225 00091300
225 IF(NDRG-O) 213,221,213 00091400
213 TPROF=TPPTL 00091500
221 IVECT2=0 00091600
        WRITE(NO,5892) 00091700
5892 FORMAT(5X,'THE PARTIAL TEST FOR OPTIMAL PROFITS FAILED'///) 00091800
        GO TO 220 00091900
220 RETURN 00092000
      END 00092100
C*****                                         00092200
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES 00092300

```

```

C BETWEEN THE LOWER LIMIT B AND UPPER LIMIT C
C*****
SUBROUTINE UNFRM
  INTEGER RO, MA, FI, EO, SLSH, ZERO, PRO, FT, EQU, INE, Z, Y, POS,
  *NY(40), NYD(40), NYF(40), NPLANT(2000), NCENT(2000), ICENT(50),
  *IDROP(40), NDROP(40), NZD(40, 50), NZ(40, 50), NZF(40, 50)
  REAL*8 P, S, B, C, D, Q, YFL, C1, C2, C3, C4, CONST1, CONST2, FAC1, FAC2,
  *FAC3, FAC4, FAC5, FAC6, FAC7, FAC8, TWOPI, PROFM, FDROPM, ADROPM,
  *SUMC, SUMR, CAPMIN, PRMAX, TCAP, RCAP, TRGLBL, TPGLBL, TRPTL,
  *TPPTL, TOFIX, TFIX, TFFIX, TRORG, TREV, TRFIN, TPORG, TPROF, TPFIN,
  *SIGMA(40, 50), MU(40, 50), R(40, 50), PC(40, 50), XSTAR(40, 50), DIF,
  *XD(40, 50), X(40, 50), XF(40, 50), PR(40, 50), PROF(2000),
  *A(40), F(40), ADROP(40), FDROP(40), AF(40), FF(40), FMS(40), AMS(40)
  COMMON/BLOCK7/B, C, D
  COMMON/BLOCK8/IX, IY, YFL
  CALL RANDU
  Q=YFL
  D=B+(C-B)*Q
  RETURN
END
C*****
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES
C BETWEEN 0 AND 1
C*****
SUBROUTINE RANDU
  INTEGER RO, MA, FI, EO, SLSH, ZERO, PRO, FT, EQU, INE, Z, Y, POS,
  *NY(40), NYD(40), NYF(40), NPLANT(2000), NCENT(2000), ICENT(50),
  *IDROP(40), NDROP(40), NZD(40, 50), NZ(40, 50), NZF(40, 50)
  REAL*8 P, S, B, C, D, Q, YFL, C1, C2, C3, C4, CONST1, CONST2, FAC1, FAC2,
  *FAC3, FAC4, FAC5, FAC6, FAC7, FAC8, TWOPI, PROFM, FDROPM, ADROPM,
  *SUMC, SUMR, CAPMIN, PRMAX, TCAP, RCAP, TRGLBL, TPGLBL, TRPTL,
  *TPPTL, TOFIX, TFIX, TFFIX, TRORG, TREV, TRFIN, TPORG, TPROF, TPFIN,
  *SIGMA(40, 50), MU(40, 50), R(40, 50), PC(40, 50), XSTAR(40, 50), DIF,
  *XD(40, 50), X(40, 50), XF(40, 50), PR(40, 50), PROF(2000),
  *A(40), F(40), ADROP(40), FDROP(40), AF(40), FF(40), FMS(40), AMS(40)
  COMMON/BLOCK8/IX, IY, YFL
  IY=IX*65539
  IF(IY) 5, 6, 6
5  IY=IY+2147483647+1
6  YFL=IY
  YFL=YFL*0.4656613E-9
  RETURN
END
00092400
00092500
00092600
00092700
00092800
00092900
00093000
00093100
00093200
00093300
00093400
00093500
00093600
00093700
00093800
00093900
00094000
00094100
00094200
00094300
00094400
00094500
00094600
00094700
00094800
00094900
00095000
00095100
00095200
00095300
00095400
00095500
00095600
00095700
00095800
00095900
00096000
00096100
00096200
00096300
00096400
00096500
00096600

```

```

C*****0000100
C
C      THIS PROGRAM PERFORMS THE HEURISTIC ALGORITHM FOR PLANT      0000200
C      LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE             0000300
C      STOCHASTIC DEMAND                                           0000400
C                                                                    0000500
C      ALSO IT SETS UP THE INPUT DATA DECK FOR EXTREME POINT     0000600
C      RANKING TECHNIQUE                                           0000700
C                                                                    0000800
C                                                                    0000900
C      THE PLANTS ARE ASSUMED UNCAPACITATED                       00001000
C      AND A TYPE 2 NORMALLY DISTRIBUTED DEMAND IS ASSUMED        00001100
C                                                                    00001200
C      THIS ALGORITHM TAKES INTO ACCOUNT OF THE FACT THAT         00001300
C      ALL DEMAND CENTERS NEED TO BE ALLOCATED AND THAT           00001400
C      EACH DEMAND CENTER MUST RECEIVE THE SUPPLY                  00001500
C      FROM AT MOST ONE PLANT                                     00001600
C                                                                    00001700
C                                                                    00001800
C      WRITTEN BY LOGENDRAN RASARATNAM                            00001900
C      SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT            00002000
C      OKLAHOMA STATE UNIVERSITY                                  00002100
C                                                                    00002200
C      DISSERTATION ADVISER: DR. M. PALMER TERRELL               00002300
C                                                                    00002400
C      VERSION 1 -- AUGUST, 1984                                   00002500
C*****00002600
C*****00002700
C                                                                    00002800
C*** GENERAL STRUCTURE AND INPUT REQUIREMENTS:                   00002900
C (MAIN PROGRAM DRIVES THE SUBROUTINES UNFRM AND GLOBE)          00003000
C (UNFRM DRIVES SUBROUTINE RANDU)                                00003100
C (GLOBE DRIVES SUBROUTINE ALOC)                                 00003200
C (ALOC DRIVES SUBROUTINE PARTL)                                 00003300
C                                                                    00003400
C*****00003500
C                                                                    00003600
C      SUBROUTINE          FUNCTION                                00003700
C      -----          -----
C      RANDU              TO GENERATE A UNIFORMLY DISTRIBUTED     00003800
C                          RANDOM NUMBER BETWEEN 0 AND 1          00003900
C      UNFRM              TO CONVERT THE RANDOM NUMBER GENERATED  00004000
C                          BY RANDU BETWEEN APPROPRIATE LIMITS    00004100
C                          ESTABLISHED FOR THE UNIFORM             00004200
C                          DISTRIBUTION TO BE USED                 00004300
C      GLOBE              PERFORMS THE GLOBAL TESTS FOR OPTIMAL    00004400
C                          PROFITS AND FEASIBLE CAPACITY          00004500
C      ALOC               PERFORMS THE FEASIBLE ALLOCATIONS FOR    00004600
C                          FOR EACH OF THE DEMAND CENTERS         00004700
C                          DICTATED BY PRIORITY RULE 1           00004800
C      PARTL             PERFORMS THE PARTIAL TESTS FOR OPTIMAL    00004900
C                          PROFITS AND FEASIBLE CAPACITY          00005000
C                                                                    00005100
C                                                                    00005200
C*****00005300
C                                                                    00005400
C*** EXTERNAL FUNCTIONS REQUIRED                                   00005500
C (1) REGULAR SYSTEM SUPPLIED FORTRAN FUNCTIONS                 00005600
C (2) TWO IMSL SUBROUTINES                                       00005700
C     MDNOR -- CUMULATIVE PROBABILITY FUNCTION OF                00005800
C             STANDARD NORMAL                                    00005900
C     MDNRIS -- INVERSE FUNCTION OF MDNOR                        00006000
C                                                                    00006100
C*****00006200
C                                                                    00006300
C*** COMMON BLOCK VARIABLE DEFINITIONS                           00006400
C ONLY THOSE VARIABLES THAT REQUIRE EXPLANATION ARE LISTED     00006500
C                                                                    00006600
C /BLOCK1/                                                       00006700
C                                                                    00006800
C     TOFIX - TOTAL FIXED COST FOR THE ORIGINAL LOCATION VECTOR  00006900
C             UNDER CONSIDERATION                               00007000
C     TFIX  - TOTAL FIXED COST FOR THE LOCATION VECTOR PRESENTLY 00007100

```

C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007200
C		FROM THE ORIGINAL LOCATION VECTOR	00007300
C	TFFIX	- TOTAL FIXED COST FOR THE FINAL LOCATION VECTOR	00007400
C		CURRENTLY STORED	00007500
C	TRORG	- TOTAL REVENUE FOR THE ORIGINAL LOCATION VECTOR	00007600
C		UNDER CONSIDERATION	00007700
C	TREV	- TOTAL REVENUE FOR THE LOCATION VECTOR PRESENTLY	00007800
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007900
C		FROM THE ORIGINAL LOCATION VECTOR	00008000
C	TRFIN	- TOTAL REVENUE FOR THE FINAL LOCATION VECTOR	00008100
C		CURRENTLY STORED	00008200
C	TPORG	- TOTAL PROFIT FOR THE ORIGINAL LOCATION VECTOR	00008300
C		UNDER CONSIDERATION	00008400
C	TPROF	- TOTAL PROFIT FOR THE LOCATION VECTOR PRESENTLY	00008500
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008600
C		FROM THE ORIGINAL LOCATION VECTOR	00008700
C	TPFIN	- TOTAL PROFIT FOR THE FINAL LOCATION VECTOR	00008800
C		CURRENTLY STORED	00008900
C			00009000
C	/BLOCK2/		00009100
C			00009200
C	IVECT1-	INDICATOR VARIABLE FOR THE TEST ON FEASIBLE CAPACITY	00009300
C	IVECT2-	INDICATOR VARIABLE FOR THE TEST ON OPTIMAL PROFITS	00009400
C	NORG	- VARIABLE INDICATING THAT THE ORIGINAL FEASIBLE	00009500
C		SOLUTION HAS BEEN FOUND	00009600
C	NCON	- TOTAL NUMBER OF PLANTS DROPPED SO FAR	00009700
C	NALOC	- TOTAL NUMBER OF DEMAND CENTERS ALLOCATED SO FAR	00009800
C	MPT	- TOTAL NUMBER OF PLANTS FOR THE PROBLEM SOLVED	00009900
C	NCT	- TOTAL NUMBER OF DEMAND CENTERS FOR THE PROBLEM SOLVED	00010000
C			00010100
C	/BLOCK3/		00010200
C			00010300
C	NYO	- THE ORIGINAL LOCATION VECTOR UNDER CONSIDERATION	00010400
C	NY	- THE LOCATION VECTOR PRESENTLY CONSIDERED BY	00010500
C		DROPPING ONE PLANT AT A TIME FROM THE	00010600
C		ORIGINAL LOCATION VECTOR	00010700
C	NYF	- THE FINAL LOCATION VECTOR CURRENTLY STORED	00010800
C	NPLANT-	VARIABLE USED FOR THE PLANTS IN RANKING THE OPTIMAL	00010900
C		PROFITS EMPLOYING PRIORITY RULE 1	00011000
C	NCENT	- VARIABLE USED FOR THE DEMAND CENTERS IN RANKING THE	00011100
C		OPTIMAL PROFITS EMPLOYING PRIORITY RULE 1	00011200
C	ICENT	- VARIABLE INDICATING WHETHER A DEMAND CENTER HAS	00011300
C		ALREADY BEEN ALLOCATED	00011400
C	IDROP	- VARIABLE INDICATING THE PLANT PRESENTLY DROPPED	00011500
C		FROM THE ORIGINAL LOCATION VECTOR	00011600
C	NDROP	- VARIABLE USED FOR RANKING THE PLANTS FOR DROP	00011700
C		EMPLOYING PRIORITY RULE 2	00011800
C			00011900
C	/BLOCK4/		00012000
C			00012100
C	XSTAR	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR EACH	00012200
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00012300
C	XO	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE ORIGINAL	00012400
C		LOCATION VECTOR UNDER CONSIDERATION	00012500
C	X	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE LOCATION	00012600
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00012700
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00012800
C	XF	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE FINAL	00012900
C		LOCATION VECTOR CURRENTLY STORED	00013000
C	PR	- OPTIMAL PROFITS DETERMINED FOR EACH COMBINATION OF	00013100
C		DEMAND CENTERS AND PROFITS	00013200
C	PROF	- THE RANKED OPTIMAL PROFITS DETERMINED FOR EACH	00013300
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00013400
C		USING PRIORITY RULE 1	00013500
C			00013600
C	/BLOCK5/		00013700
C			00013800
C	NZO	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE ORIGINAL	00013900
C		LOCATION VECTOR UNDER CONSIDERATION	00014000
C	NZ	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE LOCATION	00014100
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00014200

```

C           AT A TIME FROM THE ORIGINAL LOCATION VECTOR           00014300
C     NZF - COMPONENTS OF THE ALLOCATION MATRIX FOR THE FINAL     00014400
C           LOCATION VECTOR CURRENTLY STORED                     00014500
C                                                         00014600
C /BLOCK6/                                                     00014700
C                                                         00014800
C     A - CAPACITY OF EACH PLANT                               00014900
C     F - FIXED COST OF EACH PLANT                            00015000
C     ADROP- THE REMAINING CAPACITY OF EACH PLANT AFTER HAVING 00015100
C           ALLOCATED A SUBSET OF DEMAND CENTERS              00015200
C                                                         00015300
C*****00015400
C                                                         00015500
C*** OTHER VARIABLE DEFINITIONS                               00015600
C                                                         00015700
C     MU - MEAN OF THE NORMAL DISTRIBUTION OF DEMAND          00015800
C           DETERMINED FOR EACH COMBINATION OF                00015900
C           DEMAND CENTERS AND PLANTS                        00016000
C     SIGMA- STANDARD DEVIATION OF THE NORMAL DISTRIBUTION OF 00016100
C           DEMAND DETERMINED FOR EACH COMBINATION OF        00016200
C           DEMAND CENTERS AND PLANTS                        00016300
C     R - PER UNIT VARIABLE COST OF THE PRODUCT SUPPLIED FOR 00016400
C           EACH COMBINATION OF DEMAND CENTERS AND PLANTS    00016500
C     PC - PER UNIT PRICE OF THE PRODUCT RECEIVED FOR EACH   00016600
C           COMBINATION OF DEMAND CENTERS AND PLANTS         00016700
C                                                         00016800
C*****00016900
C     INTEGER RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,       00017000
C     *NY(40),NYG(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00017100
C     *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50);NZF(40,50)    00017200
C     REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2, 00017300
C     *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM, 00017400
C     *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00017500
C     *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00017600
C     *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00017700
C     *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00017800
C     *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00017900
C     DATA RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS/2HR0,2HMA, 00018000
C     *2HFI,2HEQ,1HO,3HPR0,2HFT,3HEQU,3HINE,1HZ,1HY,1H+/ 00018100
C     COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00018200
C     *TPFIN,TPROF 00018300
C     COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00018400
C     *,KK,JJ 00018500
C     COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00018600
C     COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00018700
C     COMMON/BLOCK5/NZO,NZ,NZF 00018800
C     COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00018900
C     COMMON/BLOCK7/B,C,D 00019000
C     COMMON/BLOCK8/IX,IY,YFL 00019100
C*****00019200
C     NO=12 00019300
C     NOT=14 00019400
C     MPT=3 00019500
C     NCT=4 00019600
C     CONST1=1.0 00019700
C     CONST2=0.0 00019800
C     MNTOT=MPT*NCT 00019900
C     MQ=NCT+MPT 00020000
C     MR=MPT*NCT+MPT 00020100
C*****00020200
C GENERATE VALUES RANDOMLY FOR F(MM) 00020300
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM 00020400
C DISTRIBUTION 00020500
C*****00020600
C     B=200. 00020700
C     C=300. 00020800
C     IX=13 00020900
C     WRITE(NO,4025) 00021000
C     4025 FORMAT(//5X,'THE PLANT FIXED COSTS ARE',//) 00021100
C     WRITE(NO,5248) 00021200
C     5248 FORMAT(5X,'FIXED COST',5X,'PLANT'//) 00021300

```

```

DO 1001 MM=1,MPT
CALL UNFRM
F(MM)=D
WRITE(NO,5249) D,MM
5249 FORMAT(1X,F14.6,7X,I2//)
IX=IY
1001 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR A(MM)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=20000.
      C=25000.
      WRITE(NO,4026)
4026 FORMAT(//5X,'THE PLANT CAPACITIES ARE',//)
      WRITE(NO,5250)
5250 FORMAT(5X,'CAPACITY',5X,'PLANT'//)
DO 1002 MM=1,MPT
CALL UNFRM
A(MM)=D
WRITE(NO,5251) D,MM
5251 FORMAT(1X,F14.6,5X,I2//)
IX=IY
1002 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR R(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=0.5
      C=2.5
      WRITE(NO,4027)
4027 FORMAT(//5X,'THE PER UNIT VARIABLE COSTS ARE'//)
      WRITE(NO,5252)
5252 FORMAT(1X,'PER UNIT VARIABLE COST',5X,'PLANT',5X,'DEMAND CENTER',/
*/)
DO 1003 MM=1,MPT
DO 1004 LL=1,NCT
CALL UNFRM
R(MM,LL)=D
WRITE(NO,5253) D,MM,LL
5253 FORMAT(5X,F14.6,9X,I2,13X,I2//)
1004 IX=IY
1003 CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR PC(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B1=3.
      C1=6.
      B2=7.
      C2=13.
      WRITE(NO,4028)
4028 FORMAT(//1X,'THE PER UNIT PRICE AND THE MEAN AND STANDARD DEVIATIO
*N OF THE DEMAND FOR THE PRODUCTS ARE'//)
      WRITE(NO,5254)
5254 FORMAT(4X,'PER UNIT PRICE',6X,'MEAN DEMAND',5X,'STANDARD DEVIATION
*',5X,'PLANT',5X,'DEMAND CENTER'//)
DO 1005 MM=1,MPT
DO 1006 LL=1,NCT
B=B1
C=C1
CALL UNFRM
PC(MM,LL)=D
MU(MM,LL)=600./PC(MM,LL)
IX=IY
B=B2
C=C2
CALL UNFRM

```

```

00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00023000
00023100
00023200
00023300
00023400
00023500
00023600
00023700
00023800
00023900
00024000
00024100
00024200
00024300
00024400
00024500
00024600
00024700
00024800
00024900
00025000
00025100
00025200
00025300
00025400
00025500
00025600
00025700
00025800
00025900
00026000
00026100
00026200
00026300
00026400
00026500
00026600
00026700
00026800
00026900
00027000
00027100
00027200
00027300
00027400
00027500
00027600
00027700
00027800
00027900
00028000
00028100
00028200
00028300
00028400

```

```

        SIGMA(MM,LL)=D                                00028500
        WRITE(NO,5255) D,MU(MM,LL),SIGMA(MM,LL),MM,LL 00028600
5255  FORMAT(1X,F14.6,5X,F14.6,5X,F14.6,11X,I2,13X,I2//) 00028700
1006  IX=IY                                           00028800
1005  CONTINUE                                         00028900
C*****                                               00029000
C COMPUTE THE VALUES OF THE OPTIMAL SUPPLY QUANTITIES XSTAR(MM,LL) 00029100
C AND THE OPTIMAL PROFITS PR(MM,LL)                  00029200
C*****                                               00029300
        DO 900 MM=1,MPT                                00029400
        DO 901 LL=1,NCT                                00029500
            P=1.-R(MM,LL)/PC(MM,LL)                   00029600
            CALL MDNRIS(P,S,IER)                       00029700
            XSTAR(MM,LL)=S*SIGMA(MM,LL)+MU(MM,LL)     00029800
            IF(XSTAR(MM,LL)-O.) 902,902,903           00029900
903    C1=PC(MM,LL)*MU(MM,LL)                         00030000
            TWOPI=2.*3.141593                          00030100
            FAC1=SQRT(TWOPI)                           00030200
            FAC2=1./FAC1                               00030300
            FAC3=XSTAR(MM,LL)-MU(MM,LL)               00030400
            FAC4=FAC3/SIGMA(MM,LL)                    00030500
            FAC5=-(FAC4**2)/2.                         00030600
            FAC6=EXP(FAC5)                             00030700
            FAC7=-PC(MM,LL)*SIGMA(MM,LL)              00030800
            C2=FAC7*FAC2*FAC6                         00030900
            S=FAC4                                      00031000
            CALL MDNOR(S,P)                            00031100
            FAC8=1.-P                                  00031200
            C3=PC(MM,LL)*FAC3*FAC8                    00031300
            C4=-R(MM,LL)*XSTAR(MM,LL)                 00031400
            PR(MM,LL)=C1+C2+C3+C4                     00031500
            IF(PR(MM,LL)-O.) 902,902,901             00031600
902    XSTAR(MM,LL)=O.                                00031700
            PR(MM,LL)=O.                               00031800
901    CONTINUE                                       00031900
900    CONTINUE                                       00032000
        WRITE(NO,4029)                                00032100
4029  FORMAT(/5X,'THE OPTIMAL SUPPLY QUANTITIES AND THE PROFITS ARE',// 00032200
*)
        WRITE(NO,5256)                                00032300
5256  FORMAT(1X,'OPTIMAL SUPPLY QUANTITY',8X,'PROFIT',8X,'PLANT',5X, 00032400
* 'DEMAND CENTER'//)
        DO 3003 MM=1,MPT                               00032500
        DO 3004 LL=1,NCT                               00032600
3004  WRITE(NO,5257) XSTAR(MM,LL),PR(MM,LL),MM,LL   00032700
5257  FORMAT(4X,F14.6,9X,F14.6,6X,I2,13X,I2//)     00032800
3003  CONTINUE                                       00032900
C*****                                               00033000
C GENERATE THE DATA DECK FOR THE                     00033100
C EXTREME POINT RANKING TECHNIQUE                   00033200
C*****                                               00033300
        WRITE(NOT,137) MQ,MR                          00033400
137   FORMAT(2I5)                                    00033500
        WRITE(NOT,138) RO                             00033600
138   FORMAT(1X,A2)                                  00033700
        WRITE(NOT,139) ZERO,PRO,FT                   00033800
139   FORMAT(11X,A1,1X,A3,A2)                       00033900
        DO 141 LL=1,NCT                               00034000
            IF(LL-9) 142,142,143                      00034100
142   WRITE(NOT,144) ZERO,EQU,ZERO,LL                00034200
144   FORMAT(11X,A1,1X,A3,A1,I1)                    00034300
            GO TO 141                                  00034400
143   WRITE(NOT,145) ZERO,EQU,LL                     00034500
145   FORMAT(11X,A1,1X,A3,I2)                       00034600
141   CONTINUE                                       00034700
        DO 146 MM=1,MPT                                00034800
            IF(MM-9) 147,147,148                      00034900
147   WRITE(NOT,144) POS,INE,ZERO,MM                 00035000
            GO TO 146                                  00035100
148   WRITE(NOT,145) POS,INE,MM                      00035200
146   CONTINUE                                       00035300
148   WRITE(NOT,145) POS,INE,MM                      00035400
146   CONTINUE                                       00035500

```

```

WRITE(NOT, 149) MA                                00035600
149 FORMAT(1X, A2)                                00035700
DO 150 LL=1, NCT                                  00035800
DO 151 MM=1, MPT                                  00035900
    IF(MM-9) 152, 152, 153                         00036000
152 IF(LL-9) 154, 154, 155                         00036100
154 WRITE(NOT, 733) Z, ZERO, MM, ZERO, LL, PRO, FT, PR(MM, LL) 00036200
733 FORMAT(7X, A1, A1, I1, A1, I1, 1X, A3, A2, 1X, D20. 10) 00036300
WRITE(NOT, 734) Z, ZERO, MM, ZERO, LL, EQU, ZERO, LL, CONST1 00036400
734 FORMAT(7X, A1, A1, I1, A1, I1, 1X, A3, A1, I1, 1X, D20. 10) 00036500
WRITE(NOT, 734) Z, ZERO, MM, ZERO, LL, INE, ZERO, MM, XSTAR(MM, LL) 00036600
GO TO 151                                          00036700
155 WRITE(NOT, 736) Z, ZERO, MM, LL, PRO, FT, PR(MM, LL) 00036800
736 FORMAT(7X, A1, A1, I1, I2, 1X, A3, A2, 1X, D20. 10) 00036900
WRITE(NOT, 737) Z, ZERO, MM, LL, EQU, LL, CONST1 00037000
737 FORMAT(7X, A1, A1, I1, I2, 1X, A3, I2, 1X, D20. 10) 00037100
WRITE(NOT, 738) Z, ZERO, MM, LL, INE, ZERO, MM, XSTAR(MM, LL) 00037200
738 FORMAT(7X, A1, A1, I1, I2, 1X, A3, A1, I1, 1X, D20. 10) 00037300
GO TO 151                                          00037400
153 IF(LL-9) 157, 157, 158                         00037500
157 WRITE(NOT, 739) Z, MM, ZERO, LL, PRO, FT, PR(MM, LL) 00037600
739 FORMAT(7X, A1, I2, A1, I1, 1X, A3, A2, 1X, D20. 10) 00037700
WRITE(NOT, 740) Z, MM, ZERO, LL, EQU, ZERO, LL, CONST1 00037800
740 FORMAT(7X, A1, I2, A1, I1, 1X, A3, A1, I1, 1X, D20. 10) 00037900
WRITE(NOT, 756) Z, MM, ZERO, LL, INE, MM, XSTAR(MM, LL) 00038000
756 FORMAT(7X, A1, I2, A1, I1, 1X, A3, I2, 1X, D20. 10) 00038100
GO TO 151                                          00038200
158 WRITE(NOT, 742) Z, MM, LL, PRO, FT, PR(MM, LL) 00038300
742 FORMAT(7X, A1, I2, I2, 1X, A3, A2, 1X, D20. 10) 00038400
WRITE(NOT, 743) Z, MM, LL, EQU, LL, CONST1 00038500
743 FORMAT(7X, A1, I2, I2, 1X, A3, I2, 1X, D20. 10) 00038600
WRITE(NOT, 743) Z, MM, LL, INE, MM, XSTAR(MM, LL) 00038700
151 CONTINUE                                       00038800
150 CONTINUE                                       00038900
DO 159 MM=1, MPT                                  00039000
    FMS(MM)=-F(MM)                                  00039100
    AMS(MM)=-A(MM)                                  00039200
    IF(MM-9) 160, 160, 161                         00039300
160 WRITE(NOT, 745) Y, ZERO, ZERO, ZERO, MM, PRO, FT, FMS(MM) 00039400
WRITE(NOT, 746) Y, ZERO, ZERO, ZERO, MM, INE, ZERO, MM, AMS(MM) 00039500
GO TO 159                                          00039600
161 WRITE(NOT, 747) Y, ZERO, ZERO, MM, PRO, FT, FMS(MM) 00039700
WRITE(NOT, 748) Y, ZERO, ZERO, MM, INE, MM, AMS(MM) 00039800
745 FORMAT(7X, A1, A1, A1, A1, I1, 1X, A3, A2, 1X, D20. 10) 00039900
746 FORMAT(7X, A1, A1, A1, A1, I1, 1X, A3, A1, I1, 1X, D20. 10) 00040000
747 FORMAT(7X, A1, A1, A1, I2, 1X, A3, A2, 1X, D20. 10) 00040100
748 FORMAT(7X, A1, A1, A1, I2, 1X, A3, I2, 1X, D20. 10) 00040200
159 CONTINUE                                       00040300
WRITE(NOT, 749) FI                                  00040400
749 FORMAT(1X, A2)                                  00040500
DO 162 LL=1, NCT                                  00040600
    IF(LL-9) 163, 163, 164                         00040700
163 WRITE(NOT, 755) EQU, ZERO, LL, CONST1          00040800
GO TO 162                                          00040900
164 WRITE(NOT, 750) EQU, LL, CONST1                00041000
755 FORMAT(13X, A3, A1, I1, 1X, D20. 10)          00041100
750 FORMAT(13X, A3, I2, 1X, D20. 10)              00041200
162 CONTINUE                                       00041300
DO 165 MM=1, MPT                                  00041400
    IF(MM-9) 166, 166, 167                         00041500
166 WRITE(NOT, 751) INE, ZERO, MM, CONST2          00041600
GO TO 165                                          00041700
167 WRITE(NOT, 752) INE, MM, CONST2                00041800
751 FORMAT(13X, A3, A1, I1, 1X, D20. 10)          00041900
752 FORMAT(13X, A3, I2, 1X, D20. 10)              00042000
165 CONTINUE                                       00042100
WRITE(NOT, 753) EO                                  00042200
753 FORMAT(1X, A2)                                  00042300
C *****                                          00042400
C SORT THE PROFITS IN A DECENDING ORDER          00042500
C *****                                          00042600

```



```

K=0
KONST=0
DO 7 I=1,MPT
DO 10 J=1,NCT
  K=KONST+J
  PROF(K)=PR(I,J)
  NPLANT(K)=I
  NCENT(K)=J
10 CONTINUE
  KONST=I*NCT
7 CONTINUE
  MN=1
15 PROFM=PROF(MN)
  NPLAM=NPLANT(MN)
  NCENTM=NCENT(MN)
DO 20 I=MN,MNTOT
  IF(PROFM-PROF(I)) 25,25,20
25 PROFM=PROF(I)
  NPLAM=NPLANT(I)
  NCENTM=NCENT(I)
  IMAX=I
20 CONTINUE
  PROF(IMAX)=PROF(MN)
  NPLANT(IMAX)=NPLANT(MN)
  NCENT(IMAX)=NCENT(MN)
  PROF(MN)=PROFM
  NPLANT(MN)=NPLAM
  NCENT(MN)=NCENTM
  MN=MN+1
  IF(MN-MNTOT) 15,15,35
35 WRITE(NO,4030)
4030 FORMAT(5X,'THE SORTED PROFITS AND THE RESPECTIVE PLANTS AND THE DE
*MAND CENTERS ARE'//)
WRITE(NO,4034)
4034 FORMAT(5X,'RANKED PROFIT',10X,'PLANT',10X,'DEMAND CENTER'//)
DO 5033 I=1,MNTOT
5033 WRITE(NO,4035) PROF(I),NPLANT(I),NCENT(I)
4035 FORMAT(3X,F14.6,12X,I2,17X,I2//)
C*****
C SORT THE PLANTS IN A DECENDING ORDER OF FIXED COSTS
C*****
DO 333 I=1,MPT
  FDROP(I)=F(I)
  ADROP(I)=A(I)
333 NDROP(I)=I
  MJ=1
37 FDROPM=FDROP(MJ)
  ADROPM=ADROP(MJ)
  NDROPM=NDROP(MJ)
DO 40 I=MJ,MPT
  IF(FDROPM-FDROP(I)) 45,45,40
45 FDROPM=FDROP(I)
  ADROPM=ADROP(I)
  NDROPM=NDROP(I)
  IMAX=I
40 CONTINUE
  FDROP(IMAX)=FDROP(MJ)
  ADROP(IMAX)=ADROP(MJ)
  NDROP(IMAX)=NDROP(MJ)
  FDROP(MJ)=FDROPM
  ADROP(MJ)=ADROPM
  NDROP(MJ)=NDROPM
  IDROP(MJ)=1
  MJ=MJ+1
  IF(MJ-MPT) 37,37,50
50 WRITE(NO,4031)
4031 FORMAT(5X,'THE SORTED FIXED COSTS AND OTHER RESPECTIVE PARAMETERS
*ARE',//)
WRITE(NO,4032)
4032 FORMAT(1X,'RANKED PLANTS FOR DROP',10X,'FIXED COST',11X,'CAPACITY'
*,11X,'FIXED COST',10X,'CAPACITY',12X,'IDROP VALUES'//)

```

```

DO 5034 I=1,MPT
5034 WRITE(NO,4033) NDROP(I),FDROP(I),ADROP(I),F(NDROP(I)),
*A(NDROP(I)),IDROP(I)
4033 FORMAT(10X,I2,17X,F14.6,6X,F14.6,6X,F14.6,6X,F14.6,10X,I7//)
C*****
C START WITH ALL PLANTS OPEN
C SET ALL NY(I) EQUAL TO 1
C*****
DO 55 K=1,MPT
55 NY(NDROP(K))=1
C*****
C INITIALIZE ALL PARAMETERS
C*****
TOFIX=0.0
TFFIX=0.0
TRORG=0.0
TRFIN=0.0
TPORG=0.0
TPFIN=0.0
IVECT1=1
IVECT2=1
NORG=0
NCON=0
DO 610 MM=1,MPT
NYO(NDROP(MM))=0
NYF(NDROP(MM))=0
AF(NDROP(MM))=0.0
FF(NDROP(MM))=0.0
DO 611 LL=1,NCT
XO(NDROP(MM),LL)=0.0
XF(NDROP(MM),LL)=0.0
X(NDROP(MM),LL)=0.0
NZO(NDROP(MM),LL)=0
NZF(NDROP(MM),LL)=0
611 NZ(NDROP(MM),LL)=0
610 CONTINUE
DO 477 LL=1,NCT
477 ICENT(LL)=0
C*****
C CALL THE SUBROUTINE GLOBE TO PERFORM THE GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
CALL GLOBE
IF(IVECT1=0) 325,323,325
323 WRITE(NO,103)
103 FORMAT(1X,'FOR THE HEURISTICS EMPLOYED NO FEASIBLE SOLUTION CAN BE
*FOUND DUE TO CAPACITY LIMITATIONS'//)
999 STOP
5074 WRITE(NO,5075)
5075 FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS - STOP'//*****
*****'//)
GO TO 999
325 IF(NORG=0) 327,328,327
328 IVECT2=1
C*****
C CONSIDER THE DROP OF EACH PLANT
C*****
327 DO 520 KK=1,MPT
IF(IDROP(KK)-1) 520,530,520
530 NY(NDROP(KK))=0
FDROP(KK)=0.0
ADROP(KK)=0.0
CALL GLOBE
IF(IVECT1=0) 535,540,535
540 IVECT1=1
IDROP(KK)=0
NCON=NCON+1
IF(NCON=MPT) 560,435,435
535 IF(IVECT2=0) 550,555,550
555 IVECT2=1
IF(TPORG=TPROF) 560,565,565
00049800
00049900
00050000
00050100
00050200
00050300
00050400
00050500
00050600
00050700
00050800
00050900
00051000
00051100
00051200
00051300
00051400
00051500
00051600
00051700
00051800
00051900
00052000
00052100
00052200
00052300
00052400
00052500
00052600
00052700
00052800
00052900
00053000
00053100
00053200
00053300
00053400
00053500
00053600
00053700
00053800
00053900
00054000
00054100
00054200
00054300
00054400
00054500
00054600
00054700
00054800
00054900
00055000
00055100
00055200
00055300
00055400
00055500
00055600
00055700
00055800
00055900
00056000
00056100
00056200
00056300
00056400
00056500
00056600
00056700
00056800

```

```

565   DIF=TPORG-TPROF .                                00056900
      IF(DIF-F(NDROP(KK))) 560,560,575                00057000
575   IDROP(KK)=0                                       00057100
      NCON=NCON+1                                       00057200
      IF(NCON-MPT) 560,435,435                         00057300
550   IF(TPORG-TPROF) 585,696,696                    00057400
696   IDN=1                                             00057500
      GO TO 693                                         00057600
585   IF(TPFIN-TPROF) 590,693,693                    00057700
590   TFFIX=TFIX                                       00057800
      TRFIN=TREV                                       00057900
      TPFIN=TPROF                                       00058000
      NN=KK                                             00058100
      DO 592 MM=1,MPT                                   00058200
        NYF(NDROP(MM))=NY(NDROP(MM))                  00058300
        IF(NYF(NDROP(MM))-1) 783,784,783              00058400
784   AF(NDROP(MM))=A(NDROP(MM))                      00058500
        FF(NDROP(MM))=F(NDROP(MM))                   00058600
        GO TO 997                                       00058700
783   AF(NDROP(MM))=0.0                                00058800
        FF(NDROP(MM))=0.0                              00058900
997   DO 593 LL=1,NCT                                  00059000
        XF(NDROP(MM),LL)=X(NDROP(MM),LL)             00059100
593   NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)             00059200
592   CONTINUE                                         00059300
      WRITE(NO,8889)                                    00059400
8889  FORMAT(//5X,'THE INTERMEDIATE FINAL SOLUTION HAS BEEN FOUND'//) 00059500
      WRITE(NO,2927)                                    00059600
2927  FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
      *'/CAPACITY',/1X,'=====',8X,'=====','5X,'=====','
      *5X,'=====','//)                               00059900
      DO 2928 MM=1,MPT                                  00060000
2928  WRITE(NO,2929) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
      *AF(NDROP(MM))                                    00060100
      *AF(NDROP(MM))                                    00060200
2929  FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)          00060300
      WRITE(NO,2931)                                    00060400
2931  FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
      *,'/PROFIT',/1X,'=====','5X,'=====','8X,'====='
      *,12X,'=====','//)                             00060600
      DO 2422 LL=1,NCT                                  00060700
      DO 2423 MM=1,MPT                                  00060800
      IF(NZF(NDROP(MM),LL)-1) 2423,2424,2423          00060900
2424  WRITE(NO,2425) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00061000
2423  CONTINUE                                         00061100
2422  CONTINUE                                         00061200
      WRITE(NO,3379) TPFIN                              00061300
3379  FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)      00061400
2425  FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)        00061500
693   DO 694 MM=1,MPT                                  00061600
      IF(NY(NDROP(MM))-1) 694,699,694                  00061700
699   ADROP(MM)=A(NDROP(MM))                           00061800
      DO 695 LL=1,NCT                                  00061900
      X(NDROP(MM),LL)=0.0                              00062000
695   NZ(NDROP(MM),LL)=0                              00062100
694   CONTINUE                                         00062200
      IF(IDN-1) 560,565,560                             00062300
560   NY(NDROP(KK))=1                                  00062400
      FDROP(KK)=F(NDROP(KK))                           00062500
      ADROP(KK)=A(NDROP(KK))                           00062600
      IF(IDN-1) 520,562,520                             00062700
562   IDN=0                                             00062800
520   CONTINUE                                         00062900
C*****                                                00063000
C SET ORIGINAL PARAMETERS EQUAL TO THE FINAL PARAMETERS AND 00063100
C THE INTERMEDIATE PARAMETERS AS APPROPRIATE           00063200
C*****                                                00063300
      TOFIX=TFFIX                                       00063400
      TRORG=TRFIN                                       00063500
      TPORG=TPFIN                                       00063600
      DO 801 MM=1,MPT                                   00063700
      NYO(NDROP(MM))=NYF(NDROP(MM))                   00063800
      NYO(NDROP(MM))=NYF(NDROP(MM))                   00063900

```

```

      NY(NDROP(MM))=NYF(NDROP(MM))
      IF(NN-MM) 803,802,803
802  IDROP(MM)=0
      NCON=NCON+1
      IF(NCON-MPT) 803,435,435
803  IF(NY(NDROP(MM))-1) 804,805,804
805  ADROP(MM)=A(NDROP(MM))
      FDROP(MM)=F(NDROP(MM))
      GO TO 806
804  ADROP(MM)=0.0
      FDROP(MM)=0.0
806  DO 807 LL=1,NCT
      XO(NDROP(MM),LL)=XF(NDROP(MM),LL)
807  NZO(NDROP(MM),LL)=NZF(NDROP(MM),LL)
801  CONTINUE
      DO 741 LL=1,NCT
741  ICENT(LL)=0
      GO TO 327
C*****
C PRINT ALL FINAL VALUES
C*****
435  IF(NORG-O) 493,5074,493
493  WRITE(NOT,967) TPFIN
967  FORMAT(D20.10)
      WRITE(NO,8883)
8883 FORMAT(/5X,'THE FINAL SOLUTION HAS BEEN FOUND'//)
      WRITE(NO,2001)
2001 FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'====',8X,'=====',5X,'=====',
* 5X,'=====',//)
      DO 2002 MM=1,MPT
2002 WRITE(NO,2003) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
* AF(NDROP(MM))
2003 FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)
      WRITE(NO,2004)
2004 FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
* ,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
* ,12X,'=====',//)
      DO 2005 LL=1,NCT
      DO 2006 MM=1,MPT
      IF(NZF(NDROP(MM),LL)-1) 2006,2023,2006
2023 WRITE(NO,2007) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2006 CONTINUE
2005 CONTINUE
2007 FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
      WRITE(NO,3379) TPFIN
      STOP
      END
C*****
C SUBROUTINE TO PERFORM GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
      SUBROUTINE GLOBE
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
* NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
* IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
* FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM,
* SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
* TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
* SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
* XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
* A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
* TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
* ,KK,JJ
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS

```

```

TCAP=0.0
TRGLBL=0.0
DO 70 LL=1,NCT
  PRMAX=0.0
  CAPMIN=1.0D20
DO 75 MM=1,MPT
  IF(NY(NDROP(MM))-1) 75,80,75
80  IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 75,85,85
85  IF(CAPMIN-XSTAR(NDROP(MM),LL)) 90,90,95
95  CAPMIN=XSTAR(NDROP(MM),LL)
90  IF(PRMAX-PR(NDROP(MM),LL)) 100,75,75
100 PRMAX=PR(NDROP(MM),LL)
75  CONTINUE
    TCAP=TCAP+CAPMIN
    TRGLBL=TRGLBL+PRMAX
70  CONTINUE
    SUMC=0.0
    TFIX=0.0
DO 105 MM=1,MPT
  IF(NY(NDROP(MM))-1) 105,110,105
110  SUMC=SUMC+ADROP(MM)
    TFIX=TFIX+FDROP(MM)
105  CONTINUE
    TPGLBL=TRGLBL-TFIX
    IF(SUMC-TCAP) 115,120,120
115  IVECT1=0
    WRITE(NO,5888)
5888  FORMAT(5X,'THE GLOBAL TEST FOR FEASIBLE CAPACITY FAILED'//)
    GO TO 133
120  IF(TPORG-TPGLBL) 125,117,117
117  IF(NORG=0) 118,119,118
118  TPROF=TPGLBL
119  IVECT2=0
    WRITE(NO,5889)
5889  FORMAT(5X,'THE GLOBAL TEST FOR OPTIMAL PROFITS FAILED'//)
    GO TO 133
125  WRITE(NO,5890)
5890  FORMAT(5X,'PASSED BOTH GLOBAL TESTS FOR FEASIBLE CAPACITY AND OPTI
*MAL PROFITS'//)
C*****
C CALL THE SUBROUTINE ALOC TO PERFORM FEASIBLE ALLOCATIONS
C*****
  CALL ALOC
133  RETURN
END
C*****
C SUBROUTINE TO PERFORM FEASIBLE ALLOCATIONS
C*****
  SUBROUTINE ALOC
    INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
    *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
    *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
    REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
    *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM,
    *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
    *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
    *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
    *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
    *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
    COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
    *TPFIN,TPROF
    COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
    *,KK,JJ
    COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
    COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
    COMMON/BLOCK5/NZO,NZ,NZF
    COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
    NALOC=0
    NTEST=1
    TREV=0.0
DO 135 JJ=1,MNTOT

```

```

          IF(ICENT(NCENT(JJ))-1) 7290,135,7290          00078200
7290      IF(NY(NPLANT(JJ))-1) 135,707,135            00078300
707      DO 708 IB=1,MPT                               00078400
          IF(NPLANT(JJ)-NDROP(IB)) 708,140,708        00078500
708      CONTINUE                                     00078600
140      IF(ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ))) 135,150,150 00078700
150      X(NPLANT(JJ),NCENT(JJ))=XSTAR(NPLANT(JJ),NCENT(JJ)) 00078800
          NZ(NPLANT(JJ),NCENT(JJ))=1                 00078900
          ICENT(NCENT(JJ))=1                          00079000
          ADROP(IB)=ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ)) 00079100
          TREV=TREV+PROF(JJ)                          00079200
          NALOC=NALOC+1                               00079300
          IF(NALOC-NCT) 152,157,157                   00079400
152      IF(AMOD(FLOAT(NALOC),FLOAT(NTEST))-O.O) 135,155,135 00079500
C*****
C CALL THE SUBROUTINE PARTL TO PERFORM PARTIAL TESTS FOR 00079600
C OPTIMAL PROFITS AND FEASIBLE CAPACITY                00079700
C*****
155      CALL PARTL                                    00079800
          IF(IVECT1-O) 132,137,132                    00079900
132      IF(IVECT2-O) 135,137,135                    00080000
135      CONTINUE                                     00080100
          IF(NALOC-NCT) 149,157,157                   00080200
157      TPROF=TREV-TFIX                              00080300
          WRITE(NO,6237) TPROF                         00080400
6237      FORMAT(/5X,'THE TOTAL PROFIT FOR THIS CONFIGURATION IS= ', 00080500
          *F14.6//)                                    00080600
          IF(NDRG-O) 151,139,151                       00080700
139      IF(TPORG-TPROF) 141,137,137                 00080800
141      TOFIX=TFIX                                   00080900
          TFFIX=TFIX                                   00081000
          TRORG=TREV                                   00081100
          TRFIN=TREV                                  00081200
          TPORG=TPROF                                  00081300
          TPFIN=TPROF                                  00081400
          DO 142 MM=1,MPT                               00081500
              NYO(NDROP(MM))=NY(NDROP(MM))            00081600
              NYF(NDROP(MM))=NY(NDROP(MM))            00081700
              IF(NY(NDROP(MM))-1) 142,146,142          00081800
146      ADROP(MM)=A(NDROP(MM))                       00081900
              AF(NDROP(MM))=A(NDROP(MM))              00082000
              FF(NDROP(MM))=F(NDROP(MM))              00082100
          DO 143 LL=1,NCT                               00082200
              XO(NDROP(MM),LL)=X(NDROP(MM),LL)        00082300
              XF(NDROP(MM),LL)=X(NDROP(MM),LL)        00082400
              NZO(NDROP(MM),LL)=NZ(NDROP(MM),LL)      00082500
              NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)      00082600
              X(NDROP(MM),LL)=O.O                      00082700
143      NZ(NDROP(MM),LL)=O                          00082800
142      CONTINUE                                     00082900
          NDRG=1                                       00083000
          WRITE(NO,5893)                               00083100
5893      FORMAT(5X,'THE ORIGINAL FEASIBLE SOLUTION HAS BEEN FOUND'//) 00083200
4232      WRITE(NO,3022)                               00083300
3022      FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X, 00083400
          *'CAPACITY',/1X,'=====',8X,'=====',5X,'=====', 00083500
          *5X,'=====',//)                             00083600
          DO 3023 MM=1,MPT                             00083700
3023      WRITE(NO,3024) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)), 00083800
          *AF(NDROP(MM))                               00083900
3024      FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)      00084000
          WRITE(NO,3025)                               00084100
3025      FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X 00084200
          *,'PROFIT',/1X,'=====',5X,'=====',8X,'=====' 00084300
          *,12X,'=====',//)                          00084400
          DO 3026 LL=1,NCT                             00084500
          DO 3027 MM=1,MPT                             00084600
              IF(NZF(NDROP(MM),LL)-1) 3027,3051,3027 00084700
3051      WRITE(NO,3028) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL) 00084800
3027      CONTINUE                                     00084900
3026      CONTINUE                                     00085000

```

```

3028 FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
      WRITE(NO,3380) TPFIN
3380 FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)
      GO TO 151
149   IVECT1=0
137   DO 153 MM=1,MPT
      IF(NY(NDROP(MM))-1) 153,710,153
710   ADROP(MM)=A(NDROP(MM))
      DO 154 LL=1,NCT
        X(NDROP(MM),LL)=0.0
154   NZ(NDROP(MM),LL)=0
153   CONTINUE
151   DO 738 LL=1,NCT
738   ICENT(LL)=0
      RETURN
      END
C*****
C SUBROUTINE TO PERFORM PARTIAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
      SUBROUTINE PARTL
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,FAC2,
      *FAC3,FAC4,FAC5,FAC6,FAC7,FAC8,TWOPI,PROFM,FDROPM,ADROPM,
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
      *SIGMA(40,50),MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
      *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
      *TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
      *,KK,JJ
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
      RCAP=0.0
      TRPTL=0.0
      DO 160 LL=1,NCT
        PRMAX=0.0
        CAPMIN=1.0D20
        IF(ICENT(LL)-1) 736,160,736
736   DO 165 MM=1,MPT
        IF(NY(NDROP(MM))-1) 165,175,165
175   IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 165,180,180
180   IF(CAPMIN-XSTAR(NDROP(MM),LL)) 185,185,190
190   CAPMIN=XSTAR(NDROP(MM),LL)
185   IF(PRMAX-PR(NDROP(MM),LL)) 195,165,165
195   PRMAX=PR(NDROP(MM),LL)
165   CONTINUE
        RCAP=RCAP+CAPMIN
        TRPTL=TRPTL+PRMAX
160   CONTINUE
        TRPTL=TREV+TRPTL
        SUMR=0.0
        DO 200 MM=1,MPT
          IF(NY(NDROP(MM))-1) 200,205,200
205   SUMR=SUMR+ADROP(MM)
200   CONTINUE
        TPPTL=TRPTL-TFIX
        IF(SUMR-RCAP) 210,215,215
210   IVECT1=0
        WRITE(NO,5891)
5891  FORMAT(5X,'THE PARTIAL TEST FOR FEASIBLE CAPACITY FAILED'//)
        GO TO 220
215   IF(TPORG-TPPTL) 220,225,225
225   IF(NORG-O) 213,221,213
213   TPROF=TPPTL

```



```

C*****0000100
C      0000200
C      THIS PROGRAM PERFORMS THE HEURISTIC ALGORITHM FOR PLANT
C      LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE
C      STOCHASTIC DEMAND
C      0000300
C      0000400
C      0000500
C      0000600
C      ALSO IT SETS UP THE INPUT DATA DECK FOR EXTREME POINT
C      RANKING TECHNIQUE
C      0000700
C      0000800
C      0000900
C      THE PLANTS ARE ASSUMED UNCAPACITATED
C      AND A UNIFORMLY DISTRIBUTED DEMAND IS ASSUMED
C      0001000
C      0001100
C      0001200
C      THIS ALGORITHM TAKES INTO ACCOUNT OF THE FACT THAT
C      ALL DEMAND CENTERS NEED TO BE ALLOCATED AND THAT
C      EACH DEMAND CENTER MUST RECEIVE THE SUPPLY
C      FROM AT MOST ONE PLANT
C      0001300
C      0001400
C      0001500
C      0001600
C      0001700
C      WRITTEN BY LOGENDRAN RASARATNAM
C      SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT
C      OKLAHOMA STATE UNIVERSITY
C      0001800
C      0001900
C      0002000
C      0002100
C      DISSERTATION ADVISER: DR. M. PALMER TERRELL
C      0002200
C      0002300
C      VERSION 1 -- AUGUST, 1984
C      0002400
C*****0002500
C      0002600
C*****0002700
C      0002800
C      0002900
C*** GENERAL STRUCTURE AND INPUT REQUIREMENTS:
C      (MAIN PROGRAM DRIVES THE SUBROUTINES UNFRM AND GLOBE)
C      (UNFRM DRIVES SUBROUTINE RANDU)
C      (GLOBE DRIVES SUBROUTINE ALOC)
C      (ALOC DRIVES SUBROUTINE PARTL)
C      0003000
C      0003100
C      0003200
C      0003300
C      0003400
C*****0003500
C      0003600
C      SUBROUTINE          FUNCTION
C      -----          -----
C      RANDU              TO GENERATE A UNIFORMLY DISTRIBUTED
C                        RANDOM NUMBER BETWEEN 0 AND 1
C      UNFRM              TO CONVERT THE RANDOM NUMBER GENERATED
C                        BY RANDU BETWEEN APPROPRIATE LIMITS
C                        ESTABLISHED FOR THE UNIFORM
C                        DISTRIBUTION TO BE USED
C      GLOBE              PERFORMS THE GLOBAL TESTS FOR OPTIMAL
C                        PROFITS AND FEASIBLE CAPACITY
C      ALOC               PERFORMS THE FEASIBLE ALLOCATIONS FOR
C                        FOR EACH OF THE DEMAND CENTERS
C                        DICTATED BY PRIORITY RULE 1
C      PARTL              PERFORMS THE PARTIAL TESTS FOR OPTIMAL
C                        PROFITS AND FEASIBLE CAPACITY
C      0003700
C      0003800
C      0003900
C      0004000
C      0004100
C      0004200
C      0004300
C      0004400
C      0004500
C      0004600
C      0004700
C      0004800
C      0004900
C      0005000
C      0005100
C      0005200
C*****0005300
C      0005400
C*** NO EXTERNAL FUNCTIONS REQUIRED
C      0005500
C      0005600
C*****0005700
C      0005800
C*** COMMON BLOCK VARIABLE DEFINITIONS
C      ONLY THOSE VARIABLES THAT REQUIRE EXPLANATION ARE LISTED
C      0005900
C      0006000
C      0006100
C /BLOCK1/
C      0006200
C      0006300
C      TOFIX - TOTAL FIXED COST FOR THE ORIGINAL LOCATION VECTOR
C      UNDER CONSIDERATION
C      0006400
C      0006500
C      TFIX - TOTAL FIXED COST FOR THE LOCATION VECTOR PRESENTLY
C      CONSIDERED BY DROPPING ONE PLANT AT A TIME
C      FROM THE ORIGINAL LOCATION VECTOR
C      0006600
C      0006700
C      0006800
C      TFFIX - TOTAL FIXED COST FOR THE FINAL LOCATION VECTOR
C      CURRENTLY STORED
C      0006900
C      0007000
C      TRORG - TOTAL REVENUE FOR THE ORIGINAL LOCATION VECTOR
C      0007100

```

C		UNDER CONSIDERATION	00007200
C	TREV	- TOTAL REVENUE FOR THE LOCATION VECTOR PRESENTLY	00007300
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00007400
C		FROM THE ORIGINAL LOCATION VECTOR	00007500
C	TRFIN	- TOTAL REVENUE FOR THE FINAL LOCATION VECTOR	00007600
C		CURRENTLY STORED	00007700
C	TPORG	- TOTAL PROFIT FOR THE ORIGINAL LOCATION VECTOR	00007800
C		UNDER CONSIDERATION	00007900
C	TPROF	- TOTAL PROFIT FOR THE LOCATION VECTOR PRESENTLY	00008000
C		CONSIDERED BY DROPPING ONE PLANT AT A TIME	00008100
C		FROM THE ORIGINAL LOCATION VECTOR	00008200
C	TPFIN	- TOTAL PROFIT FOR THE FINAL LOCATION VECTOR	00008300
C		CURRENTLY STORED	00008400
C			00008500
C	/BLOCK2/		00008600
C			00008700
C	IVECT1-	INDICATOR VARIABLE FOR THE TEST ON FEASIBLE CAPACITY	00008800
C	IVECT2-	INDICATOR VARIABLE FOR THE TEST ON OPTIMAL PROFITS	00008900
C	NORG	- VARIABLE INDICATING THAT THE ORIGINAL FEASIBLE	00009000
C		SOLUTION HAS BEEN FOUND	00009100
C	NCON	- TOTAL NUMBER OF PLANTS DROPPED SO FAR	00009200
C	NALOC	- TOTAL NUMBER OF DEMAND CENTERS ALLOCATED SO FAR	00009300
C	MPT	- TOTAL NUMBER OF PLANTS FOR THE PROBLEM SOLVED	00009400
C	NCT	- TOTAL NUMBER OF DEMAND CENTERS FOR THE PROBLEM SOLVED	00009500
C			00009600
C	/BLOCK3/		00009700
C			00009800
C	NYO	- THE ORIGINAL LOCATION VECTOR UNDER CONSIDERATION	00009900
C	NY	- THE LOCATION VECTOR PRESENTLY CONSIDERED BY	00010000
C		DROPPING ONE PLANT AT A TIME FROM THE	00010100
C		ORIGINAL LOCATION VECTOR	00010200
C	NYF	- THE FINAL LOCATION VECTOR CURRENTLY STORED	00010300
C	NPLANT-	VARIABLE USED FOR THE PLANTS IN RANKING THE OPTIMAL	00010400
C		PROFITS EMPLOYING PRIORITY RULE 1	00010500
C	NCENT	- VARIABLE USED FOR THE DEMAND CENTERS IN RANKING THE	00010600
C		OPTIMAL PROFITS EMPLOYING PRIORITY RULE 1	00010700
C	ICENT	- VARIABLE INDICATING WHETHER A DEMAND CENTER HAS	00010800
C		ALREADY BEEN ALLOCATED	00010900
C	IDROP	- VARIABLE INDICATING THE PLANT PRESENTLY DROPPED	00011000
C		FROM THE ORIGINAL LOCATION VECTOR	00011100
C	NDROP	- VARIABLE USED FOR RANKING THE PLANTS FOR DROP	00011200
C		EMPLOYING PRIORITY RULE 2	00011300
C			00011400
C	/BLOCK4/		00011500
C			00011600
C	XSTAR	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR EACH	00011700
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00011800
C	XO	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE ORIGINAL	00011900
C		LOCATION VECTOR UNDER CONSIDERATION	00012000
C	X	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE LOCATION	00012100
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00012200
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00012300
C	XF	- OPTIMAL SUPPLY QUANTITIES DETERMINED FOR THE FINAL	00012400
C		LOCATION VECTOR CURRENTLY STORED	00012500
C	PR	- OPTIMAL PROFITS DETERMINED FOR EACH COMBINATION OF	00012600
C		DEMAND CENTERS AND PROFITS	00012700
C	PROF	- THE RANKED OPTIMAL PROFITS DETERMINED FOR EACH	00012800
C		COMBINATION OF DEMAND CENTERS AND PLANTS	00012900
C		USING PRIORITY RULE 1	00013000
C			00013100
C	/BLOCK5/		00013200
C			00013300
C	NZO	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE ORIGINAL	00013400
C		LOCATION VECTOR UNDER CONSIDERATION	00013500
C	NZ	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE LOCATION	00013600
C		VECTOR PRESENTLY CONSIDERED BY DROPPING ONE PLANT	00013700
C		AT A TIME FROM THE ORIGINAL LOCATION VECTOR	00013800
C	NZF	- COMPONENTS OF THE ALLOCATION MATRIX FOR THE FINAL	00013900
C		LOCATION VECTOR CURRENTLY STORED	00014000
C			00014100
C	/BLOCK6/		00014200

```

C                                          00014300
C      A      - CAPACITY OF EACH PLANT      00014400
C      F      - FIXED COST OF EACH PLANT    00014500
C      ADROP- THE REMAINING CAPACITY OF EACH PLANT AFTER HAVING
C              ALLOCATED A SUBSET OF DEMAND CENTERS      00014600
C                                          00014700
C                                          00014800
C*****00014900
C                                          00015000
C*** OTHER VARIABLE DEFINITIONS          00015100
C                                          00015200
C      A1     - THE LOWER LIMIT OF THE UNIFORM DISTRIBUTION OF
C              DEMAND DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS      00015300
C              00015400
C              00015500
C      A2     - THE UPPER LIMIT OF THE UNIFORM DISTRIBUTION OF
C              DEMAND DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS      00015600
C              00015700
C              00015800
C      MU     - MEAN OF THE UNIFORM DISTRIBUTION OF DEMAND
C              DETERMINED FOR EACH COMBINATION OF
C              DEMAND CENTERS AND PLANTS      00015900
C              00016000
C              00016100
C      R      - PER UNIT VARIABLE COST OF THE PRODUCT SUPPLIED FOR
C              EACH COMBINATION OF DEMAND CENTERS AND PLANTS 00016200
C              00016300
C      PC     - PER UNIT PRICE OF THE PRODUCT RECEIVED FOR EACH
C              COMBINATION OF DEMAND CENTERS AND PLANTS      00016400
C              00016500
C              00016600
C*****00016700
C      INTEGER RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS,
C      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
C      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)
C      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,
C      *FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROPM,ADROPM,
C      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
C      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
C      *MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
C      *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
C      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
C      DATA RO,MA,FI,EO,ZERO,PRO,FT,EQU,INE,Z,Y,POS/2HR0,2HMA,
C      *2HFI,2HEO,1HO,3HPRO,2HFT,3HEQU,3HINE,1HZ,1HY,1H+/
C      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
C      *TPFIN,TPROF
C      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
C      *,KK,JJ
C      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
C      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
C      COMMON/BLOCK5/NZO,NZ,NZF
C      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
C      COMMON/BLOCK7/B,C,D
C      COMMON/BLOCK8/IX,IY,YFL
C*****00019000
C      NO=12
C      NOT=14
C      MPT=3
C      NCT=4
C      CONST1=1.0
C      CONST2=0.0
C      MNTOT=MPT*NCT
C      MQ=NCT+MPT
C      MR=MPT*NCT+MPT
C*****00020000
C GENERATE VALUES RANDOMLY FOR F(MM)    00020100
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION                            00020200
C*****00020300
C*****00020400
C      B=200.
C      C=300.
C      IX=13
C      WRITE(NO,4025)
C      4025 FORMAT(//5X,'THE PLANT FIXED COSTS ARE',//)
C      WRITE(NO,5248)
C      5248 FORMAT(5X,'FIXED COST',6X,'PLANT'//)
C      DO 1001 MM=1,MPT
C      CALL UNFRM
C*****00021300

```

```

      F(MM)=D
      WRITE(NO,5249) D,MM
5249  FORMAT(1X,F14.6,7X,I2//)
      IX=IY
1001  CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR A(MM)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=20000.
      C=25000.
      WRITE(NO,4026)
4026  FORMAT(/5X,'THE PLANT CAPACITIES ARE'//)
      WRITE(NO,5250)
5250  FORMAT(5X,'CAPACITY',6X,'PLANT'//)
      DO 1002 MM=1,MPT
      CALL UNFRM
      A(MM)=D
      WRITE(NO,5251) D,MM
5251  FORMAT(1X,F14.6,5X,I2//)
      IX=IY
1002  CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR R(MM,LL)
C SET B AND C EQUAL TO THE UPPER AND LOWER LIMITS OF THE UNIFORM
C DISTRIBUTION
C*****
      B=0.5
      C=2.5
      WRITE(NO,4027)
4027  FORMAT(/5X,'THE PER UNIT VARIABLE COSTS ARE'//)
      WRITE(NO,5252)
5252  FORMAT(1X,'PER UNIT VARIABLE COST',5X,'PLANT',5X,'DEMAND CENTER',/
*/)
      DO 1003 MM=1,MPT
      DO 1004 LL=1,NCT
      CALL UNFRM
      R(MM,LL)=D
      WRITE(NO,5253) D,MM,LL
5253  FORMAT(5X,F14.6,9X,I2,13X,I2//)
1004  IX=IY
1003  CONTINUE
C*****
C GENERATE VALUES RANDOMLY FOR THE LOWER AND UPPER LIMITS OF THE
C UNIFORM DISTRIBUTION DESCRIPTIVE OF THE DEMAND
C*****
      WRITE(NO,4028)
4028  FORMAT(/1X,'THE PER UNIT PRICE AND THE MEAN OF THE DEMAND DISTRIBUTION
*/
FOR THE PRODUCT IS'//)
      WRITE(NO,5254)
5254  FORMAT(4X,'PER UNIT PRICE',4X,'MEAN DEMAND',11X,'LOWER',13X,
*/
'UPPER',13X,'PLANT',5X,'DEMAND CENTER'//)
      B1=50.
      C1=100.
      B2=150.
      C2=300.
      DO 7005 MM=1,MPT
      DO 7006 LL=1,NCT
      B=B1
      C=C1
      CALL UNFRM
      A1(MM,LL)=D
      IX=IY
      B=B2
      C=C2
      CALL UNFRM
      A2(MM,LL)=D
      MU(MM,LL)=(A1(MM,LL)+A2(MM,LL))/2.
      PC(MM,LL)=600./MU(MM,LL)
      WRITE(NO,5255) PC(MM,LL),MU(MM,LL),A1(MM,LL),A2(MM,LL),MM,LL
00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00023000
00023100
00023200
00023300
00023400
00023500
00023600
00023700
00023800
00023900
00024000
00024100
00024200
00024300
00024400
00024500
00024600
00024700
00024800
00024900
00025000
00025100
00025200
00025300
00025400
00025500
00025600
00025700
00025800
00025900
00026000
00026100
00026200
00026300
00026400
00026500
00026600
00026700
00026800
00026900
00027000
00027100
00027200
00027300
00027400
00027500
00027600
00027700
00027800
00027900
00028000
00028100
00028200
00028300
00028400

```



```

155 WRITE(NOT,736) Z,ZERO,MM,LL,PRO,FT,PR(MM,LL) 00035600
736 FORMAT(7X,A1,A1,I1,I2,1X,A3,A2,1X,D20.10) 00035700
WRITE(NOT,737) Z,ZERO,MM,LL,EQU,LL,CONST1 00035800
737 FORMAT(7X,A1,A1,I1,I2,1X,A3,I2,1X,D20.10) 00035900
WRITE(NOT,738) Z,ZERO,MM,LL,INE,ZERO,MM,XSTAR(MM,LL) 00036000
738 FORMAT(7X,A1,A1,I1,I2,1X,A3,A1,I1,1X,D20.10) 00036100
GO TO 151 00036200
153 IF(LL-9) 157,157,158 00036300
157 WRITE(NOT,739) Z,MM,ZERO,LL,PRO,FT,PR(MM,LL) 00036400
739 FORMAT(7X,A1,I2,A1,I1,1X,A3,A2,1X,D20.10) 00036500
WRITE(NOT,740) Z,MM,ZERO,LL,EQU,ZERO,LL,CONST1 00036600
740 FORMAT(7X,A1,I2,A1,I1,1X,A3,A1,I1,1X,D20.10) 00036700
WRITE(NOT,756) Z,MM,ZERO,LL,INE,MM,XSTAR(MM,LL) 00036800
756 FORMAT(7X,A1,I2,A1,I1,1X,A3,I2,1X,D20.10) 00036900
GO TO 151 00037000
158 WRITE(NOT,742) Z,MM,LL,PRO,FT,PR(MM,LL) 00037100
742 FORMAT(7X,A1,I2,I2,1X,A3,A2,1X,D20.10) 00037200
WRITE(NOT,743) Z,MM,LL,EQU,LL,CONST1 00037300
743 FORMAT(7X,A1,I2,I2,1X,A3,I2,1X,D20.10) 00037400
WRITE(NOT,743) Z,MM,LL,INE,MM,XSTAR(MM,LL) 00037500
151 CONTINUE 00037600
150 CONTINUE 00037700
DO 159 MM=1,MPT 00037800
FMS(MM)=-F(MM) 00037900
AMS(MM)=-A(MM) 00038000
IF(MM-9) 160,160,161 00038100
160 WRITE(NOT,745) Y,ZERO,ZERO,ZERO,MM,PRO,FT,FMS(MM) 00038200
WRITE(NOT,746) Y,ZERO,ZERO,ZERO,MM,INE,ZERO,MM,AMS(MM) 00038300
GO TO 159 00038400
161 WRITE(NOT,747) Y,ZERO,ZERO,MM,PRO,FT,FMS(MM) 00038500
WRITE(NOT,748) Y,ZERO,ZERO,MM,INE,MM,AMS(MM) 00038600
745 FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A2,1X,D20.10) 00038700
746 FORMAT(7X,A1,A1,A1,A1,I1,1X,A3,A1,I1,1X,D20.10) 00038800
747 FORMAT(7X,A1,A1,A1,A1,I2,1X,A3,A2,1X,D20.10) 00038900
748 FORMAT(7X,A1,A1,A1,I2,1X,A3,I2,1X,D20.10) 00039000
159 CONTINUE 00039100
WRITE(NOT,749) FI 00039200
749 FORMAT(1X,A2) 00039300
DO 162 LL=1,NCT 00039400
IF(LL-9) 163,163,164 00039500
163 WRITE(NOT,755) EQU,ZERO,LL,CONST1 00039600
GO TO 162 00039700
164 WRITE(NOT,750) EQU,LL,CONST1 00039800
755 FORMAT(13X,A3,A1,I1,1X,D20.10) 00039900
750 FORMAT(13X,A3,I2,1X,D20.10) 00040000
162 CONTINUE 00040100
DO 165 MM=1,MPT 00040200
IF(MM-9) 166,166,167 00040300
166 WRITE(NOT,751) INE,ZERO,MM,CONST2 00040400
GO TO 165 00040500
167 WRITE(NOT,752) INE,MM,CONST2 00040600
751 FORMAT(13X,A3,A1,I1,1X,D20.10) 00040700
752 FORMAT(13X,A3,I2,1X,D20.10) 00040800
165 CONTINUE 00040900
WRITE(NOT,753) EO 00041000
753 FORMAT(1X,A2) 00041100
C***** 00041200
C SORT THE PROFITS IN A DECENDING ORDER 00041300
C***** 00041400
K=0 00041500
KONST=0 00041600
DO 7 I=1,MPT 00041700
DO 10 J=1,NCT 00041800
K=KONST+J 00041900
PROF(K)=PR(I,J) 00042000
NPLANT(K)=I 00042100
NCENT(K)=J 00042200
10 CONTINUE 00042300
KONST=I*NCT 00042400
7 CONTINUE 00042500
MN=1 00042600

```

```

15     PROFM=PROF(MN)                                00042700
      NPLAM=NPLANT(MN)                              00042800
      NCENTM=NCENT(MN)                              00042900
      DO 20 I=MN,MNTOT                               00043000
      IF(PROFM-PROF(I)) 25,25.20                   00043100
25     PROFM=PROF(I)                                00043200
      NPLAM=NPLANT(I)                              00043300
      NCENTM=NCENT(I)                              00043400
      IMAX=I                                         00043500
20     CONTINUE                                     00043600
      PROF(IMAX)=PROF(MN)                          00043700
      NPLANT(IMAX)=NPLANT(MN)                     00043800
      NCENT(IMAX)=NCENT(MN)                       00043900
      PROF(MN)=PROFM                               00044000
      NPLANT(MN)=NPLAM                             00044100
      NCENT(MN)=NCENTM                             00044200
      MN=MN+1                                       00044300
      IF(MN-MNTOT) 15,15.35                        00044400
35     WRITE(NO,4030)                               00044500
4030  FORMAT(5X,'THE SORTED PROFITS AND THE RESPECTIVE PLANTS AND THE DEO
      *MAND CENTERS ARE'//)                        00044700
      WRITE(NO,4034)                               00044800
4034  FORMAT(5X,'RANKED PROFIT',10X,'PLANT',10X,'DEMAND CENTER'//)
      DO 5033 I=1,MNTOT                            00045000
5033  WRITE(NO,4035) PROF(I),NPLANT(I),NCENT(I)   00045100
4035  FORMAT(3X,F14.6,12X,I2,17X,I2//)           00045200
C*****                                           00045300
C SORT THE PLANTS IN A DECENDING ORDER OF FIXED COSTS 00045400
C*****                                           00045500
      DO 333 I=1,MPT
      FDROP(I)=F(I)
      ADROP(I)=A(I)
333   NDROP(I)=I
      MJ=1
37   FDROPM=FDROP(MJ)
      ADROPM=ADROP(MJ)
      NDROPM=NDROP(MJ)
      DO 40 I=MJ,MPT
      IF(FDROPM-FDROP(I)) 45,45.40
45   FDROPM=FDROP(I)
      ADROPM=ADROP(I)
      NDROPM=NDROP(I)
      IMAX=I
40   CONTINUE
      FDROP(IMAX)=FDROP(MJ)
      ADROP(IMAX)=ADROP(MJ)
      NDROP(IMAX)=NDROP(MJ)
      FDROP(MJ)=FDROPM
      ADROP(MJ)=ADROPM
      NDROP(MJ)=NDROPM
      IDROP(MJ)=1
      MJ=MJ+1
      IF(MJ-MPT) 37,37.50
50   WRITE(NO,4031)
4031  FORMAT(5X,'THE SORTED FIXED COSTS AND OTHER RESPECTIVE PARAMETERS
      *ARE'//)
      WRITE(NO,4032)
4032  FORMAT(1X,'RANKED PLANTS FOR DROP',10X,'FIXED COST',11X,'CAPACITY'
      *,11X,'FIXED COST',10X,'CAPACITY',12X,'IDROP VALUES'//)
      DO 5034 I=1,MPT
5034  WRITE(NO,4033) NDROP(I),FDROP(I),ADROP(I),F(NDROP(I)),
      *A(NDROP(I)),IDROP(I)
4033  FORMAT(10X,I2,17X,F14.6,6X,F14.6,6X,F14.6,6X,F14.6,10X,I7//)
C*****                                           00049000
C START WITH ALL PLANTS OPEN                       00049100
C SET ALL NY(I) EQUAL TO 1                         00049200
C*****                                           00049300
      DO 55 K=1,MPT
55   NY(NDROP(K))=1
C*****                                           00049600
C INITIALIZE ALL PARAMETERS                       00049700

```

```

C*****
TOFIX=0.0
TFFIX=0.0
TRORG=0.0
TRFIN=0.0
TPORG=0.0
TPFIN=0.0
IVECT1=1
IVECT2=1
NORG=0
NCON=0
DO 610 MM=1,MPT
  NYO(NDROP(MM))=0
  NYF(NDROP(MM))=0
  AF(NDROP(MM))=0.0
  FF(NDROP(MM))=0.0
DO 611 LL=1,NCT
  XO(NDROP(MM),LL)=0.0
  XF(NDROP(MM),LL)=0.0
  X(NDROP(MM),LL)=0.0
  NZO(NDROP(MM),LL)=0
  NZF(NDROP(MM),LL)=0
611  NZ(NDROP(MM),LL)=0
610  CONTINUE
DO 477 LL=1,NCT
477  ICENT(LL)=0
C*****
C CALL THE SUBROUTINE GLOBE TO PERFORM THE GLOBAL TESTS FOR
C OPTIMAL PROFITS AND FEASIBLE CAPACITY
C*****
CALL GLOBE
  IF(IVECT1=0) 325,323,325
323  WRITE(NO,103)
103  FORMAT(1X,'FOR THE HEURISTICS EMPLOYED NO FEASIBLE SOLUTION CAN BE
*FOUND DUE TO CAPACITY LIMITATIONS'//)
999  STOP
5074 WRITE(NO,5075)
5075 FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS - STOP'//'*****
*****'//)
GO TO 999
325  IF(NORG=0) 327,328,327
328  IVECT2=1
C*****
C CONSIDER THE DROP OF EACH PLANT
C*****
327  DO 520 KK=1,MPT
  IF(IDROP(KK)-1) 520,530,520
530  NY(NDROP(KK))=0
  FDROP(KK)=0.0
  ADROP(KK)=0.0
CALL GLOBE
  IF(IVECT1=0) 535,540,535
540  IVECT1=1
  IDROP(KK)=0
  NCON=NCON+1
  IF(NCON=MPT) 560,435,435
535  IF(IVECT2=0) 550,555,550
555  IVECT2=1
  IF(TPORG-TPROF) 560,565,565
565  DIF=TPORG-TPROF
  IF(DIF-F(NDROP(KK))) 560,560,575
575  IDROP(KK)=0
  NCON=NCON+1
  IF(NCON=MPT) 560,435,435
550  IF(TPORG-TPROF) 585,696,696
696  IDN=1
GO TO 693
585  IF(TPFIN-TPROF) 590,693,693
590  TFFIX=TFIX
  TRFIN=TREV
  TPFIN=TPROF

```



```

NN=KK
DO 592 MM=1,MPT
  NYF(NDROP(MM))=NY(NDROP(MM))
  IF(NYF(NDROP(MM))-1) 783,784,783
784  AF(NDROP(MM))=A(NDROP(MM))
  FF(NDROP(MM))=F(NDROP(MM))
  GO TO 997
783  AF(NDROP(MM))=0.0
  FF(NDROP(MM))=0.0
997  DO 593 LL=1,NCT
  XF(NDROP(MM),LL)=X(NDROP(MM),LL)
593  NZF(NDROP(MM),LL)=NZ(NDROP(MM),LL)
592  CONTINUE
  WRITE(NO,8889)
8889  FORMAT(//5X,'THE INTERMEDIATE FINAL SOLUTION HAS BEEN FOUND'//)
  WRITE(NO,2927)
2927  FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED COST',5X,
* 'CAPACITY',/1X,'=====',8X,'=====',5X,'=====',
*5X,'=====',//)
  DO 2928 MM=1,MPT
2928  WRITE(NO,2929) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
*AF(NDROP(MM))
2929  FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)
  WRITE(NO,2931)
2931  FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
*,'PROFIT',/1X,'=====',5X,'=====',8X,'=====',
*,'12X,'=====',//)
  DO 2422 LL=1,NCT
  DO 2423 MM=1,MPT
  IF(NZF(NDROP(MM),LL)-1) 2423,2424,2423
2424  WRITE(NO,2425) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2423  CONTINUE
2422  CONTINUE
  WRITE(NO,3379) TPFIN
3379  FORMAT(1X,'THE TOTAL PROFIT IS = ',F14.6//)
2425  FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)
693  DO 694 MM=1,MPT
  IF(NY(NDROP(MM))-1) 694,699,694
699  ADROP(MM)=A(NDROP(MM))
  DO 695 LL=1,NCT
  X(NDROP(MM),LL)=0.0
695  NZ(NDROP(MM),LL)=0
694  CONTINUE
  IF(IDN-1) 560,565,560
560  NY(NDROP(KK))=1
  FDROP(KK)=F(NDROP(KK))
  ADROP(KK)=A(NDROP(KK))
  IF(IDN-1) 520,562,520
562  IDN=0
520  CONTINUE
C*****
C SET ORIGINAL PARAMETERS EQUAL TO THE FINAL PARAMETERS AND
C THE INTERMEDIATE PARAMETERS AS APPROPRIATE
C*****
  TOFIX=TFFIX
  TRORG=TRFIN
  TPORG=TPFIN
  DO 801 MM=1,MPT
  NYO(NDROP(MM))=NYF(NDROP(MM))
  NY(NDROP(MM))=NYF(NDROP(MM))
  IF(NN-MM) 803,802,803
802  IDROP(MM)=0
  NCON=NCON+1
  IF(NCON-MPT) 803,435,435
803  IF(NY(NDROP(MM))-1) 804,805,804
805  ADROP(MM)=A(NDROP(MM))
  FDROP(MM)=F(NDROP(MM))
  GO TO 806
804  ADROP(MM)=0.0
  FDROP(MM)=0.0
806  DO 807 LL=1,NCT

```

```

      XO(NDROP(MM),LL)=XF(NDROP(MM),LL)                                00064000
807   NZO(NDROP(MM),LL)=NZF(NDROP(MM),LL)                            00064100
801   CONTINUE                                                         00064200
      DO 741 LL=1,NCT                                                  00064300
841   ICENT(LL)=0                                                      00064400
      GO TO 327                                                         00064500
C*****                                                                00064600
C PRINT ALL FINAL VALUES                                           00064700
C*****                                                                00064800
435   IF(NORG=0) 493,5074,493                                         00064900
493   WRITE(NDROP,967) TPFIN                                          00065000
967   FORMAT(D20.10)                                                  00065100
      WRITE(NDROP,8883)                                                00065200
8883  FORMAT(/5X,'THE FINAL SOLUTION HAS BEEN FOUND'//)              00065300
      WRITE(NDROP,2001)                                                00065400
2001  FORMAT(1X,'PLANT',8X,'O=CLOSED/1=OPEN',5X,'FIXED_COST',5X,
* 'CAPACITY',/1X,'=====',8X,'=====',5X,'=====',
* 5X,'=====')//)                                                  00065500
      DO 2002 MM=1,MPT                                                00065600
2002  WRITE(NDROP,2003) NDROP(MM),NYF(NDROP(MM)),FF(NDROP(MM)),
* AF(NDROP(MM))                                                      00065700
2003  FORMAT(1X,I4,8X,I8,8X,F14.6,2X,F14.6//)                        00065800
      WRITE(NDROP,2004)                                                00065900
2004  FORMAT(1X,'DEMAND CENTER',5X,'SUPPLIED BY PLANT',8X,'QUANTITY',12X
* 'PROFIT',/1X,'=====',5X,'=====',8X,'=====')//)          00066000
      DO 2005 LL=1,NCT                                                00066100
      DO 2006 MM=1,MPT                                                00066200
      IF(NZF(NDROP(MM),LL)-1) 2006,2023,2006                          00066300
2023  WRITE(NDROP,2007) LL,NDROP(MM),XF(NDROP(MM),LL),PR(NDROP(MM),LL)
2006  CONTINUE                                                         00066400
2005  CONTINUE                                                         00066500
2007  FORMAT(7X,I2,18X,I2,10X,F14.6,5X,F14.6//)                      00066600
      WRITE(NDROP,3379) TPFIN                                          00066700
      STOP                                                             00066800
      END                                                             00066900
C*****                                                                00067000
C SUBROUTINE TO PERFORM GLOBAL TESTS FOR OPTIMAL PROFITS            00067100
C AND FEASIBLE CAPACITY                                             00067200
C*****                                                                00067300
      SUBROUTINE GLOBE                                                00067400
      INTEGER RD,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS,          00067500
* NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50),
* IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50)              00067600
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1,
* FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDRPMP,ADROPMP,
* SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL,
* TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN,
* MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF,
* XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000),
* A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40)
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG,
* TPFIN,TPROF
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO
* ,KK,JJ
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF
      COMMON/BLOCK5/NZO,NZ,NZF
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS
      TCAP=0.0
      TRGLBL=0.0
      DO 70 LL=1,NCT
      PRMAX=0.0
      CAPMIN=1.0D20
      DO 75 MM=1,MPT
      IF(NY(NDROP(MM))-1) 75,80,75
80   IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 75,85,85
85   IF(CAPMIN-XSTAR(NDROP(MM),LL)) 90,90,95
95   CAPMIN=XSTAR(NDROP(MM),LL)
90   IF(PRMAX-PR(NDROP(MM),LL)) 100,75,75
100  PRMAX=PR(NDROP(MM),LL)

```

```

75  CONTINUE                                00071100
    TCAP=TCAP+CAPMIN                        00071200
    TRGLBL=TRGLBL+PRMAX                    00071300
70  CONTINUE                                00071400
    SUMC=0.0                                00071500
    TFIX=0.0                                00071600
    DO 105 MM=1,MPT                          00071700
      IF(NY(NDROP(MM))-1) 105,110,105      00071800
110  SUMC=SUMC+ADROP(MM)                    00071900
      TFIX=TFIX+FDROP(MM)                  00072000
105  CONTINUE                                00072100
      TPGLBL=TRGLBL-TFIX                    00072200
      IF(SUMC-TCAP) 115,120,120            00072300
115  IVECT1=0                                00072400
      WRITE(NO,5888)                        00072500
5888  FORMAT(5X,'THE GLOBAL TEST FOR FEASIBLE CAPACITY FAILED'//) 00072600
      GO TO 133                              00072700
120  IF(TPORG-TPGLBL) 125,117,117          00072800
117  IF(NORG-O) 118,119,118                00072900
118  TPROF=TPGLBL                          00073000
119  IVECT2=0                                00073100
      WRITE(NO,5889)                        00073200
5889  FORMAT(5X,'THE GLOBAL TEST FOR OPTIMAL PROFITS FAILED'//) 00073300
      GO TO 133                              00073400
125  WRITE(NO,5890)                          00073500
5890  FORMAT(5X,'PASSED BOTH GLOBAL TESTS FOR FEASIBLE CAPACITY AND OPTI00073600
      *MAL PROFITS'//)                      00073700
C*****                                00073800
C CALL THE SUBROUTINE ALOC TO PERFORM FEASIBLE ALLOCATIONS 00073900
C*****                                00074000
      CALL ALOC                              00074100
133  RETURN                                  00074200
      END                                    00074300
C*****                                00074400
C SUBROUTINE TO PERFORM FEASIBLE ALLOCATIONS 00074500
C*****                                00074600
      SUBROUTINE ALOC                        00074700
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00074800
      *NY(40),NYD(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00074900
      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00075000
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1, 00075100
      *FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROPM,ADROPM, 00075200
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00075300
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00075400
      *MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00075500
      *XO(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00075600
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40),AMS(40) 00075700
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00075800
      *TPFIN,TPROF 00075900
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00076000
      *,KK,JJ 00076100
      COMMON/BLOCK3/NY,NYD,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00076200
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00076300
      COMMON/BLOCK5/NZO,NZ,NZF 00076400
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00076500
      NALOC=0 00076600
      NTEST=1 00076700
      TREV=0.0 00076800
      DO 135 JJ=1,MNTOT 00076900
        IF(ICENT(NCENT(JJ))-1) 7290,135,7290 00077000
7290  IF(NY(NPLANT(JJ))-1) 135,707,135 00077100
707  DO 708 IB=1,MPT 00077200
        IF(NPLANT(JJ)-NDROP(IB)) 708,140,708 00077300
708  CONTINUE 00077400
140  IF(ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ))) 135,150,150 00077500
150  X(NPLANT(JJ),NCENT(JJ))=XSTAR(NPLANT(JJ),NCENT(JJ)) 00077600
      NZ(NPLANT(JJ),NCENT(JJ))=1 00077700
      ICENT(NCENT(JJ))=1 00077800
      ADROP(IB)=ADROP(IB)-XSTAR(NPLANT(JJ),NCENT(JJ)) 00077900
      TREV=TREV+PROF(JJ) 00078000
      NALOC=NALOC+1 00078100

```



```

151 DO 738 LL=1,NCT                                00085300
738 ICENT(LL)=O                                    00085400
      RETURN                                        00085500
      END                                            00085600
C*****                                            00085700
C SUBROUTINE TO PERFORM PARTIAL TESTS FOR OPTIMAL  00085800
C PROFITS AND FEASIBLE CAPACITY                    00085900
C*****                                            00086000
      SUBROUTINE PARTL                               00086100
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00086200
      *NY(40),NYO(40),NYF(40),NPLANT(2000),NCENT(2000),ICENT(50), 00086300
      *IDROP(40),NDROP(40),NZO(40,50),NZ(40,50),NZF(40,50) 00086400
      REAL*8 P,S,B,C,D,Q,YFL,C1,C2,C3,C4,CONST1,CONST2,FAC1, 00086500
      *FAC2,FAC3,FAC4,A1(40,50),A2(40,50),PROFM,FDROP,ADROP, 00086600
      *SUMC,SUMR,CAPMIN,PRMAX,TCAP,RCAP,TRGLBL,TPGLBL,TRPTL, 00086700
      *TPPTL,TOFIX,TFIX,TFFIX,TRORG,TREV,TRFIN,TPORG,TPROF,TPFIN, 00086800
      *MU(40,50),R(40,50),PC(40,50),XSTAR(40,50),DIF, 00086900
      *XD(40,50),X(40,50),XF(40,50),PR(40,50),PROF(2000), 00087000
      *A(40),F(40),ADROP(40),FDROP(40),AF(40),FF(40),FMS(40).AMS(40) 00087100
      COMMON/BLOCK1/TOFIX,TFFIX,TFIX,TRORG,TRFIN,TREV,TPORG, 00087200
      *TPFIN,TPROF 00087300
      COMMON/BLOCK2/IVECT1,IVECT2,NORG,NCON,NALOC,MPT,NCT,MNTOT,NO 00087400
      *,KK,JJ 00087500
      COMMON/BLOCK3/NY,NYO,NYF,NPLANT,NCENT,ICENT,IDROP,NDROP 00087600
      COMMON/BLOCK4/XSTAR,XO,X,XF,PR,PROF 00087700
      COMMON/BLOCK5/NZO,NZ,NZF 00087800
      COMMON/BLOCK6/A,F,ADROP,FDROP,AF,FF,FMS,AMS 00087900
      RCAP=0.0 00088000
      TRPTL=0.0 00088100
      DO 160 LL=1,NCT 00088200
      PRMAX=0.0 00088300
      CAPMIN=1.0D20 00088400
      IF(ICENT(LL)-1) 736,160,736 00088500
736 DO 165 MM=1,MPT 00088600
      IF(NY(NDROP(MM))-1) 165,175,165 00088700
175 IF(ADROP(MM)-XSTAR(NDROP(MM),LL)) 165,180,180 00088800
180 IF(CAPMIN-XSTAR(NDROP(MM),LL)) 185,185,190 00088900
190 CAPMIN=XSTAR(NDROP(MM),LL) 00089000
185 IF(PRMAX-PR(NDROP(MM),LL)) 195,165,165 00089100
195 PRMAX=PR(NDROP(MM),LL) 00089200
165 CONTINUE 00089300
      RCAP=RCAP+CAPMIN 00089400
      TRPTL=TRPTL+PRMAX 00089500
160 CONTINUE 00089600
      TRPTL=TREV+TRPTL 00089700
      SUMR=0.0 00089800
      DO 200 MM=1,MPT 00089900
      IF(NY(NDROP(MM))-1) 200,205,200 00090000
205 SUMR=SUMR+ADROP(MM) 00090100
200 CONTINUE 00090200
      TPPTL=TRPTL-TFIX 00090300
      IF(SUMR-RCAP) 210,215,215 00090400
210 IVECT1=0 00090500
      WRITE(NO,5891) 00090600
5891 FORMAT(5X,'THE PARTIAL TEST FOR FEASIBLE CAPACITY FAILED'//) 00090700
      GO TO 220 00090800
215 IF(TPORG-TPPTL) 220,225,225 00090900
225 IF(NDRG=0) 213,221,213 00091000
213 TPROF=TPPTL 00091100
221 IVECT2=0 00091200
      WRITE(NO,5892) 00091300
5892 FORMAT(5X,'THE PARTIAL TEST FOR OPTIMAL PROFITS FAILED'//) 00091400
      GO TO 220 00091500
220 RETURN 00091600
      END 00091700
C*****                                            00091800
C SUBROUTINE FOR GENERATING THE UNIFORM RANDOM VARIATES 00091900
C BETWEEN THE LOWER LIMIT B AND UPPER LIMIT C      00092000
C*****                                            00092100
      SUBROUTINE UNFRM                               00092200
      INTEGER RO,MA,FI,EO,SLSH,ZERO,PRO,FT,EQU,INE,Z,Y,POS, 00092300

```



```

C*****0000100
C 0000200
C THIS PROGRAM PERFORMS THE ALGORITHMIC STEPS FOR THE REVISED MODEL 0000300
C BASED ON THE EXTREME POINT RANKING TECHNIQUE 0000400
C 0000500
C THE PROBLEM IS SOLVED IN THE MAIN MEMORY 0000600
C 0000700
C THE REVISED MODEL IS A 0-1 INTEGER BINARY LINEAR PROGRAMMING 0000800
C MODEL. AS SUCH, COMPUTATIONS ARE PRIMARILY BASED ON THE 0000900
C PRIMAL SIMPLEX METHOD FOR BOUNDED VARIABLES 0001000
C 0001100
C WRITTEN BY LOGENDRAN RASARATNAM 0001200
C SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT 0001300
C OKLAHOMA STATE UNIVERSITY 0001400
C 0001500
C DISSERTATION ADVISER: DR. M. PALMER TERRELL 0001600
C VERSION 1 -- AUGUST, 1984 0001700
C 0001800
C*****00001900
C 00002000
C* GENERAL STRUCTURE AND INPUT REQUIREMENTS 00002100
C* ONLY A SINGLE MAIN PROGRAM IS USED. NO SUBROUTINES REQUIRED. 00002200
C 00002300
C DESCRIPTION OF INPUT PARAMETERS 00002400
C 00002500
C M1 = MAXIMUM NUMBER OF ROWS IN THE REVISED MODEL, EXCLUDING THE 00002600
C OBJECTIVE ROW 00002700
C N1 = MAXIMUM NUMBER OF STRUCTURAL VARIABLES IN THE REVISED MODEL 00002800
C RNM1 = FIRST FOUR CHARACTERS OF ROW (CONSTRAINT) NAME 00002900
C RNM2 = FIFTH (AND FINAL) CHARACTER OF ROW (CONSTRAINT) NAME 00003000
C CLNM1 = FIRST FOUR CHARACTERS OF COLUMN NAME 00003100
C CLNM2 = FIFTH (AND FINAL) CHARACTER OF COLUMN NAME 00003200
C CDID CARD GROUP HEADER 00003300
C PUNCHED RO FOR ROW IDENTIFICATION HEADER 00003400
C PUNCHED MA FOR MATRIX HEADER 00003500
C PUNCHED FI FOR RHS HEADER 00003600
C PUNCHED EO FOR PROBLEM DELIMITER 00003700
C LGE IDENTIFIES TYPE OF CONSTRAINT 00003800
C + FOR LESS THAN OR EQUAL TO 00003900
C O FOR EQUAL TO 00004000
C VALUE = NUMERICAL VALUE OF MATRIX ELEMENT, OR RHS VALUE 00004100
C ZBAR = THE LOWER BOUND DETERMINED FROM THE HEURISTIC ALGORITHM 00004200
C 00004300
C DESCRIPTION OF VARIABLES 00004400
C 00004500
C IBN1 = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE RANKED 00004600
C EXTREME POINT CURRENTLY CONSIDERED 00004700
C IBN2 = FIFTH CHARACTER OF THE ROW NAME FOR THE RANKED 00004800
C EXTREME POINT CURRENTLY CONSIDERED 00004900
C ITES1 = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE ADJACENT 00005000
C EXTREME POINT CURRENTLY GENERATED 00005100
C ITES2 = FIFTH CHARACTER OF THE ROW NAME FOR THE ADJACENT 00005200
C EXTREME POINT CURRENTLY GENERATED 00005300
C ISTE1 = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE ADJACENT 00005400
C EXTREME POINT STORED 00005500
C ISTE2 = FIFTH CHARACTER OF THE ROW NAME FOR THE ADJACENT 00005600
C EXTREME POINT STORED 00005700
C IBAR1 = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE 00005800
C OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR 00005900
C IBAR2 = FIFTH CHARACTER OF THE ROW NAME FOR THE OPTIMAL/ 00006000
C BEST BINARY SOLUTION FOUND SO FAR 00006100
C O - BASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY 00006200
C IFLAG = { CONSIDERED IS NOT AT ITS UPPER BOUND 00006300
C 1 - BASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY 00006400
C CONSIDERED IS AT ITS UPPER BOUND 00006500
C O - BASIC VARIABLE OF THE ADJACENT EXTREME POINT CURRENTLY 00006600
C IFTES = { GENERATED IS NOT AT ITS UPPER BOUND 00006700
C 1 - BASIC VARIABLE OF THE ADJACENT EXTREME POINT CURRENTLY 00006800
C GENERATED IS AT ITS UPPER BOUND 00006900
C O - BASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED 00007000
C IFSTE = { IS NOT AT ITS UPPER BOUND 00007100

```

C	1 - BASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED	00007200
C	IS AT ITS UPPER BOUND	00007300
C	0 - BASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION	00007400
C	IFBAR = { IS NOT AT ITS UPPER BOUND	00007500
C	1 - BASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION	00007600
C	IS AT ITS UPPER BOUND	00007700
C	BP = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE	00007800
C	RANKED EXTREME POINT CURRENTLY CONSIDERED	00007900
C	BPTES = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE	00008000
C	ADJACENT EXTREME POINT CURRENTLY GENERATED	00008100
C	BPSTE = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE	00008200
C	ADJACENT EXTREME POINT STORED	00008300
C	BPBAR = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE	00008400
C	OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR	00008500
C	RQ = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE RANKED	00008600
C	EXTREME POINT CURRENTLY CONSIDERED	00008700
C	RQTES = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE ADJACENT	00008800
C	EXTREME POINT CURRENTLY GENERATED	00008900
C	RQSTE = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE ADJACENT	00009000
C	EXTREME POINT STORED	00009100
C	RQBAR = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE	00009200
C	OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR	00009300
C	RB = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE	00009400
C	-RANKED EXTREME POINT CURRENTLY CONSIDERED	00009500
C	RBTES = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE	00009600
C	ADJACENT EXTREME POINT CURRENTLY GENERATED	00009700
C	RBSTE = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE	00009800
C	ADJACENT EXTREME POINT STORED	00009900
C	RBBAR = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE	00010000
C	OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR	00010100
C	B = M1*N1 MATRIX REPRESENTING THE RANKED EXTREME POINT	00010200
C	CURRENTLY CONSIDERED	00010300
C	BTES = M1*N1 MATRIX REPRESENTING THE ADJACENT EXTREME POINT	00010400
C	CURRENTLY GENERATED	00010500
C	BSTE = M1*N1 MATRIX REPRESENTING THE ADJACENT EXTREME POINT	00010600
C	STORED	00010700
C	BBAR = M1*N1 MATRIX REPRESENTING THE OPTIMAL/BEST	00010800
C	BINARY SOLUTION FOUND SO FAR	00010900
C	NBN1 = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE RANKED	00011000
C	EXTREME POINT CURRENTLY CONSIDERED	00011100
C	NBN2 = FIFTH CHARACTER OF THE COLUMN NAME FOR THE RANKED	00011200
C	EXTREME POINT CURRENTLY CONSIDERED	00011300
C	NTES1 = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE ADJACENT	00011400
C	EXTREME POINT CURRENTLY GENERATED	00011500
C	NTES2 = FIFTH CHARACTER OF THE COLUMN NAME FOR THE ADJACENT	00011600
C	EXTREME POINT CURRENTLY GENERATED	00011700
C	NSTE1 = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE ADJACENT	00011800
C	EXTREME POINT STORED	00011900
C	NSTE2 = FIFTH CHARACTER OF THE COLUMN NAME FOR THE ADJACENT	00012000
C	EXTREME POINT STORED	00012100
C	NBAR1 = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE	00012200
C	OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR	00012300
C	NBAR2 = FIFTH CHARACTER OF THE COLUMN NAME FOR THE OPTIMAL/	00012400
C	BEST BINARY SOLUTION FOUND SO FAR	00012500
C	0 - NONBASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY	00012600
C	CONSIDERED IS NOT AT ITS UPPER BOUND	00012700
C	1 - NONBASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY	00012800
C	CONSIDERED IS AT ITS UPPER BOUND	00012900
C	0 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT	00013000
C	CURRENTLY GENERATED IS NOT AT ITS UPPER BOUND	00013100
C	1 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT	00013200
C	CURRENTLY GENERATED IS AT ITS UPPER BOUND	00013300
C	0 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED	00013400
C	IS NOT AT ITS UPPER BOUND	00013500
C	1 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED	00013600
C	IS AT ITS UPPER BOUND	00013700
C	0 - NONBASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION	00013800
C	IS NOT AT ITS UPPER BOUND	00013900
C	1 - BASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION	00014000
C	IS AT ITS UPPER BOUND	00014100
C	NBP = OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE	00014200


```

C          OF THE RANKED EXTREME POINT CURRENTLY CONSIDERED          00014300
C NBPTE= OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE    00014400
C          OF THE ADJACENT EXTREME POINT CURRENTLY GENERATED        00014500
C NBPSTE= OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE    00014600
C          OF THE ADJACENT EXTREME POINT STORED                      00014700
C NBPBAR= OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE    00014800
C          OF THE OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR         00014900
C UB      = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE 00015000
C          RANKED EXTREME POINT CURRENTLY CONSIDERED                00015100
C UBTES = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE 00015200
C          ADJACENT EXTREME POINT CURRENTLY GENERATED              00015300
C UBSTE = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE 00015400
C          ADJACENT EXTREME POINT STORED                            00015500
C UBBAR = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE 00015600
C          OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR                00015700
C PI      = SHADOW PRICES FOR THE RANKED EXTREME POINT CURRENTLY     00015800
C          CONSIDERED                                               00015900
C PITES = SHADOW PRICES FOR THE ADJACENT EXTREME POINT CURRENTLY     00016000
C          GENERATED                                                00016100
C PISTE = SHADOW PRICES FOR THE ADJACENT EXTREME POINT STORED        00016200
C PIBAR = SHADOW PRICES FOR THE OPTIMAL/BEST BINARY SOLUTION          00016300
C          FOUND SO FAR                                             00016400
C FN      = OBJECTIVE FUNCTION VALUE FOR THE RANKED EXTREME POINT     00016500
C          CURRENTLY CONSIDERED                                     00016600
C FTES = OBJECTIVE FUNCTION VALUE FOR THE ADJACENT EXTREME POINT     00016700
C          CURRENTLY GENERATED                                      00016800
C FSTE = OBJECTIVE FUNCTION VALUE FOR THE ADJACENT EXTREME POINT     00016900
C          STORED                                                  00017000
C FBAR = OBJECTIVE FUNCTION VALUE FOR THE OPTIMAL/BEST                00017100
C          BINARY SOLUTION FOUND SO FAR                             00017200
C          0 - ADJACENT EXTREME POINT HAS NO POTENTIAL              00017300
C IIND = { FOR BEING CONSIDERED IN THE FUTURE                       00017400
C          1 - ADJACENT EXTREME POINT HAS POTENTIAL                 00017500
C          FOR BEING CONSIDERED IN THE FUTURE                       00017600
C          0 - AN IMPROVED BINARY FEASIBLE SOLUTION HAS NOT BEEN    00017700
C NOPER = { FOUND IN THE CURRENT CYCLE                               00017800
C          1 - AN IMPROVED BINARY FEASIBLE SOLUTION HAS BEEN        00017900
C          FOUND IN THE CURRENT CYCLE                               00018000
C IT      = TOTAL NUMBER OF SIMPLEX ITERATIONS                       00018100
C LSTE = TOTAL NUMBER OF DISTINCT ADJACENT EXTREME POINTS GENERATED 00018200
C          00018300
C *****00018400
C          INTEGER RNM1,RNM2,CLNM1,CLNM2,BLNK,POS,CDID,RO,MA,FI,EO,    00018500
C          *IBN1(18),IBN2(18),ITES1(18),ITES2(18),IBAR1(18),IBAR2(18), 00018600
C          *ISTE1(18,400),ISTE2(18,400),NBN1(88),NBN2(88),NTES1(88), 00018700
C          *NTES2(88),NBAR1(88),NBAR2(88),NSTE1(88,400),NSTE2(88,400), 00018800
C          *IFLAG(18),IFTES(18),IFBAR(18),IFSTE(18,400),NFLAG(88),NFTES(88), 00018900
C          *NFBAR(88),NFSTE(88,400),IIND(400)                        00019000
C          REAL*8 PIVOT,LST,FN,CJBAR,VALUE,BP(18),BPTES(18),BPBAR(18), 00019100
C          *BPSTE(18,400),RO(18),RQTES(18),RQBAR(18),RQSTE(18,400), 00019200
C          *B(18,88),BTES(18,88),BBAR(18,88),BSTE(18,88,400), 00019300
C          *PI(88),PITES(88),PIBAR(88),PISTE(88,400),NBP(88), 00019400
C          *NBPTES(88),NBPBAR(88),NBPSTE(88,400),RB(18),RBTES(18), 00019500
C          *RBBAR(18),RBSTE(18,400),XPI(88),UB(88),UBTES(88), 00019600
C          *UBBAR(88),UBSTE(88,400),FSTE(400),FBAR,FTES,ZBAR,THETA1, 00019700
C          *THETA2,RATIO1,RATIO2,RMIN,SAVE,XL,X                     00019800
C          DATA RO,MA,POS,FI,EO,BLNK/2HRO,2HMA,1H+,2HFI,2HEO,4H---/ 00019900
C *****00020000
C INITIALIZE 00020100
C *****00020200
C          M=0 00020300
C          N=1 00020400
C          NROWS=0 00020500
C          NLE=0 00020600
C          NEQ=0 00020700
C          NEL=0 00020800
C          NRHS=0 00020900
C          NCOLS=0 00021000
C          NI=5 00021100
C          NO=12 00021200
C          READ(NI,1) M1,N1 00021300

```

```

1          FORMAT(2I5)                                00021400
C*****                                              00021500
C CLEAR MATRIX TO ZERO                                00021600
C*****                                              00021700
      DO 12 I=1,M1                                    00021800
      DO 12 J=1,N1                                    00021900
12         B(I,J)=0                                    00022000
C*****                                              00022100
C READ FIRST CARD-SHOULD BE ROWID                    00022200
C*****                                              00022300
      READ(NI,2) CDID                                  00022400
2         FORMAT(1X,A2)                                00022500
          IF(CDID-RO) 3,680,3                          00022600
3         WRITE(NO,3333)                               00022700
3333      FORMAT(//'ROWID CARD MISSING'//)             00022800
3334      STOP                                          00022900
C*****                                              00023000
C READ AND STORE ROWID CARDS INCLUDING DUMMY READ    00023100
C FOR OBJECTIVE ROW NAME                             00023200
C GENERATE POSITIVE AND NEGATIVE SLACKS AS REQUIRED   00023300
C*****                                              00023400
680      READ(NI,681)                                  00023500
681      FORMAT(80X)                                    00023600
101     READ(NI,102) CDID,LGE,RNM1,RNM2               00023700
102     FORMAT(1X,A2,8X,A1,1X,A4,A1)                 00023800
          IF(CDID-MA) 103,104,103                     00023900
103     M=M+1                                          00024000
          NROWS=NROWS+1                                00024100
          IF(LGE-POS) 105,106,105                      00024200
105     IBN1(M)=RNM1                                   00024300
          IBN2(M)=RNM2                                   00024400
          NEQ=NEQ+1                                     00024500
          BP(M)=-1.0                                    00024600
          RB(M)=1.0D20                                  00024700
          GO TO 101                                      00024800
106     IBN1(M)=RNM1                                   00024900
          IBN2(M)=RNM2                                   00025000
          NLE=NLE+1                                     00025100
          BP(M)=0.0                                     00025200
          RB(M)=1.0D20                                  00025300
          GO TO 101                                      00025400
C*****                                              00025500
C READ AND STORE FIRST MATRIX ELEMENT                00025600
C*****                                              00025700
104     READ(NI,195) CDID,CLNM1,CLNM2,RNM1,RNM2,VALUE 00025800
195     FORMAT(1X,A2,4X,A4,A1,1X,A4,A1,1X,D20.10)    00025900
          GO TO 119                                      00026000
109     IF(NBN1(N)-CLNM1) 111,600,111                00026100
600     IF(NBN2(N)-CLNM2) 111,112,111                00026200
112     DO 113 I=1,M                                  00026300
          IF(IBN1(I)-RNM1) 113,602,113                 00026400
602     IF(IBN2(I)-RNM2) 113,116,113                 00026500
113     CONTINUE                                       00026600
          WRITE(NO,8113)                                00026700
8113    FORMAT(//'INCORRECT INGREDIENT CARD'//)       00026800
          STOP                                          00026900
116     B(I,N)=VALUE                                    00027000
C*****                                              00027100
C READ AND STORE MATRIX ELEMENTS                    00027200
C*****                                              00027300
117     READ(NI,195) CDID,CLNM1,CLNM2,RNM1,RNM2,VALUE 00027400
          NEL=NEL+1                                     00027500
          GO TO 109                                      00027600
111     N=N+1                                          00027700
          NEL=NEL-1                                     00027800
          NCOLS=NCOLS+1                                 00027900
          IF(CDID-FI) 119,190,119                      00028000
119     NBN1(N)=CLNM1                                  00028100
          NBN2(N)=CLNM2                                  00028200
          NBP(N)=VALUE                                   00028300
          UB(N)=1.0                                     00028400

```

```

GO TO 117 00028500
C*****00028600
C READ AND STORE RHS ELEMENTS 00028700
C*****00028800
190 DO 191 I=1,M 00028900
191 RQ(I)=0. 00029000
120 READ(NI,121) CDID,RNM1,RNM2,VALUE 00029100
121 FORMAT(1X,A2,10X,A4,A1,1X,D20.10) 00029200
IF(CDID-EO) 122,193,122 00029300
122 DO 124 I=1,M 00029400
IF(IBN1(I)-RNM1) 124,610,124 00029500
610 IF(IBN2(I)-RNM2) 124,125,124 00029600
124 CONTINUE 00029700
WRITE(NO,8124) RNM1,RNM2 00029800
8124 FORMAT(///'NO ROW NAME FOR',A4,A1,///) 00029900
STOP 00030000
125 RQ(I)=VALUE 00030100
NRHS=NRHS+1 00030200
GO TO 120 00030300
193 READ(NI,339) ZBAR 00030400
339 FORMAT(D20.10) 00030500
WRITE(NO,340) ZBAR 00030600
340 FORMAT(1X,'ZBAR= ',D20.10//) 00030700
N=N-1 00030800
WRITE(NO,551) NROWS,NCOLS,NLE,NEQ,NRHS,NEL 00030900
551 FORMAT(1X,'ROWS= ',I3//1X,'COLS= ',I4//1X,'LE ROWS= ',I3//
*1X,'EQ ROWS= ',I3//1X,'RHS = ',I3//1X,'NONZERO MATRIX ELEMENTS=
* ',I6//) 00031100
C*****00031200
C*****00031300
C BLANK OUT ARTIFICIAL NAMES 00031400
C*****00031500
DO 10 I=1,M 00031600
IF(BP(I)+1.0) 10,11,10 00031700
11 IBN1(I)=BLNK 00031800
IBN2(I)=BLNK 00031900
10 CONTINUE 00032000
C*****00032100
C ACCUMULATE COUNT OF INFEASIBILITIES 00032200
C*****00032300
NINF=0 00032400
DO 6000 I=1,M 00032500
IF(BP(I)) 6001,6000,6000 00032600
6001 NINF=NINF+1 00032700
6000 CONTINUE 00032800
C*****00032900
C GENERATE INDICATORS FOR MINIMIZATION OF INFEASIBILITY 00033000
C*****00033100
DO 6101 J=1,N 00033200
XPI(J)=0. 00033300
DO 6101 I=1,M 00033400
IF(BP(I)) 6102,6101,6101 00033500
6102 XPI(J)=XPI(J)-B(I,J) 00033600
6101 CONTINUE 00033700
DO 6002 I=1,M 00033800
6002 BP(I)=0. 00033900
IPHASE=1 00034000
DO 1203 I=1,M 00034100
1203 IFLAG(I)=0 00034200
DO 1204 J=1,N 00034300
1204 NFLAG(J)=0 00034400
C*****00034500
C MAIN ROUTINE 00034600
C*****00034700
IT=0 00034800
C*****00034900
C CALCULATE SHADOW PRICES 00035000
C*****00035100
5232 DO 194 J=1,N 00035200
PI(J)=-NBP(J) 00035300
DO 194 I=1,M 00035400
194 PI(J)=PI(J)+BP(I)*B(I,J) 00035500

```

```

C***** 00035600
C SELECT BEST NONBASIS VECTOR 00035700
C***** 00035800
9101 LST=-1.0D-7 00035900
      KCOL=0 00036000
      GO TO (751,552),IPHASE 00036100
751 IF(NINF) 5433,5433,552 00036200
552 DO 9102 J=1,N 00036300
C***** 00036400
C IGNORE ARTIFICIAL VARIABLES 00036500
C***** 00036600
      IF(NBN1(J)-BLNK+NBN2(J)-BLNK) 651,9102,651 00036700
651 GO TO (6003,6004),IPHASE 00036800
6003 IF(XPI(J)-LST) 6005,9102,9102 00036900
6005 KCOL=J 00037000
      LST=XPI(J) 00037100
      GO TO 9102 00037200
6004 IF(PI(J)-LST) 9103,9102,9102 00037300
9103 KCOL=J 00037400
      LST=PI(J) 00037500
9102 CONTINUE 00037600
      IF(KCOL) 5433,5433,9104 00037700
C***** 00037800
C DETERMINE KEY ROW 00037900
C***** 00038000
9104 KROW1=0 00038100
      KROW2=0 00038200
      CJBAR=LST 00038300
      THETA1=1.0D20 00038400
      THETA2=1.0D20 00038500
      DO 9105 I=1,M 00038600
        IF(B(I,KCOL)) 9106,9105,9107 00038700
9106 RATIO2=(RB(I)-RQ(I))/(-B(I,KCOL)) 00038800
      IF(RATIO2-THETA2) 9108,9105,9105 00038900
9108 THETA2=RATIO2 00039000
      KROW2=I 00039100
      GO TO 9105 00039200
9107 RATIO1=RQ(I)/B(I,KCOL) 00039300
      IF(RATIO1-THETA1) 9109,9105,9105 00039400
9109 THETA1=RATIO1 00039500
      KROW1=I 00039600
9105 CONTINUE 00039700
      RMIN=THETA1 00039800
      IF(RMIN-THETA2) 9110,9110,9111 00039900
9111 RMIN=THETA2 00040000
9110 IF(RMIN-UB(KCOL)) 9112,9112,9113 00040100
9113 RMIN=UB(KCOL) 00040200
9112 IF(RMIN-THETA1) 9114,9115,9114 00040300
9114 IF(RMIN-THETA2) 9116,9117,9116 00040400
C***** 00040500
C RMIN=UB(KCOL).THEREFORE, X(KCOL) REMAINS NONBASIC BUT AT ITS UPPER 00040600
C BOUND. THUS IT MUST BE SUBSTITUTED AS X(KCOL)=UB(KCOL)-X'(KCOL). 00040700
C IF NFLAG(KCOL)=0 *OR* AS X'(KCOL)=UB(KCOL)-X(KCOL) IF NFLAG(KCOL)=1 00040800
C***** 00040900
9116 ND=3 00041000
9223 IF(NFLAG(KCOL)-1) 677,678,677 00041100
677 NFLAG(KCOL)=1 00041200
      GO TO 679 00041300
678 NFLAG(KCOL)=0 00041400
679 DO 9118 I=1,M 00041500
      RQ(I)=RQ(I)-B(I,KCOL) 00041600
9118 B(I,KCOL)=-B(I,KCOL) 00041700
      IT=IT+1 00041800
      GO TO (9119,9120),IPHASE 00041900
9119 PI(KCOL)=-PI(KCOL) 00042000
      XPI(KCOL)=-XPI(KCOL) 00042100
      GO TO 9121 00042200
9120 PI(KCOL)=-PI(KCOL) 00042300
      GO TO 9121 00042400
C***** 00042500
9115 KROW=KROW1 00042600

```

```

          ID=1                                00042700
          ND=1                                00042800
          GO TO 900                            00042900
9117    KROW=KROW2                            00043000
          ID=2                                00043100
          ND=2                                00043200
C*****                                     00043300
C TRANSFORM                                  00043400
C DIVIDE BY PIVOT                            00043500
C*****                                     00043600
900     PIVOT=B(KROW,KCOL)                    00043700
        DO 9128 J=1,N                          00043800
9128    B(KROW,J)=B(KROW,J)/PIVOT             00043900
        RQ(KROW)=RQ(KROW)/PIVOT               00044000
        DO 9129 I=1,M                          00044100
          IF(I-KROW) 9130,9129,9130           00044200
9130    RQ(I)=RQ(I)-RQ(KROW)*B(I,KCOL)        00044300
        DO 4444 J=1,N                          00044400
          IF(J-KCOL) 9131,4444,9131           00044500
9131    B(I,J)=B(I,J)-B(KROW,J)*B(I,KCOL)    00044600
4444    CONTINUE                               00044700
9129    CONTINUE                               00044800
        DO 9132 I=1,M                          00044900
9132    B(I,KCOL)=-B(I,KCOL)/PIVOT           00045000
        B(KROW,KCOL)=1.O/PIVOT                00045100
C*****                                     00045200
C INTERCHANGE BASIS AND NONBASIS VARIABLES  00045300
C*****                                     00045400
        RNM1=NB1(KCOL)                         00045500
        RNM2=NB2(KCOL)                         00045600
        NB1(KCOL)=IB1(KROW)                    00045700
        NB2(KCOL)=IB2(KROW)                    00045800
        IB1(KROW)=RNM1                         00045900
        IB2(KROW)=RNM2                         00046000
        LST=NB(KCOL)                           00046100
        NBP(KCOL)=BP(KROW)                     00046200
        BP(KROW)=LST                           00046300
        J1=NFLAG(KCOL)                         00046400
        NFLAG(KCOL)=IFLAG(KROW)                00046500
        IFLAG(KROW)=J1                         00046600
        XL=UB(KCOL)                             00046700
        UB(KCOL)=RB(KROW)                       00046800
        RB(KROW)=XL                             00046900
        GO TO (801,802),ID                      00047000
801     IT=IT+1                                00047100
802     IF(NB1(KCOL)-BLNK+NB2(KCOL)-BLNK) 9138,9133,9138 00047200
9133    NINF=NINF-1                             00047300
9138    GO TO (9134,9135),IPHASE                00047400
9134    SAVE=PI(KCOL)                           00047500
        DO 9136 J=1,N                          00047600
          PI(J)=PI(J)-SAVE*B(KROW,J)           00047700
9136    XPI(J)=XPI(J)-CJBAR*B(KROW,J)         00047800
          PI(KCOL)=-SAVE/PIVOT                 00047900
          XPI(KCOL)=-CJBAR/PIVOT               00048000
        GO TO (9121,9116),ID                   00048100
9135    DO 9137 J=1,N                          00048200
9137    PI(J)=PI(J)-CJBAR*B(KROW,J)           00048300
          PI(KCOL)=-CJBAR/PIVOT               00048400
        GO TO (9121,9223),ID                   00048500
C*****                                     00048600
C COMPUTE OBJECTIVE FUNCTION                  00048700
C*****                                     00048800
9121    FN=0.                                    00048900
        DO 9122 I=1,M                          00049000
          IF(IFLAG(I)-1) 9123,9124,9123       00049100
9124    FN=FN+BP(I)*(1.-RQ(I))                 00049200
        GO TO 9122                             00049300
9123    FN=FN+BP(I)*RQ(I)                       00049400
9122    CONTINUE                               00049500
        DO 9125 J=1,N                          00049600
          IF(NFLAG(J)-1) 9125,9126,9125       00049700

```

```

9126  FN=FN+NBP(J)                                00049800
9125  CONTINUE                                    00049900
C*****00050000
C CHECK FOR ESSENTIAL ZERO                        00050100
C*****00050200
      DO 6111 I=1,M                                00050300
      DO 6111 J=1,N                                00050400
          X=B(I,J)                                  00050500
          IF(ABS(X)-1.0D-7) 6112,6112,6111         00050600
6112  B(I,J)=0.                                     00050700
6111  CONTINUE                                    00050800
C*****00050900
C LOG ITERATION                                  00051000
C*****00051100
      GO TO (7001,7002,7003),ND                    00051200
7001  WRITE(NO,7004)                               00051300
7004  FORMAT(2X,'OPERATION TYPE 1'///)             00051400
      GO TO 7010                                    00051500
7002  WRITE(NO,7005)                               00051600
7005  FORMAT(2X,'OPERATION TYPE 2'///)            00051700
      GO TO 7010                                    00051800
7003  WRITE(NO,7006)                               00051900
7006  FORMAT(2X,'OPERATION TYPE 3 = NO BASIS CHANGE'///) 00052000
      WRITE(NO,7007)                               00052100
7007  FORMAT(1X,'ITERATION NUMBER',7X,'KEY COLUMN VARIABLE',
      *10X,'OBJECTIVE FUNCTION'///)                00052200
      WRITE(NO,7008) IT,NBN1(KCOL),NBN2(KCOL),FN    00052400
7008  FORMAT(I9,22X,A4,A1,17X,F14.6)              00052500
      GO TO 7009                                    00052600
7010  WRITE(NO,9202)                               00052700
9202  FORMAT(1X,'ITERATION NUMBER',7X,'VAR IN',6X,'VAR OUT',11X,
      *'OBJECTIVE FUNCTION'///)                    00052800
      WRITE(NO,9127) IT,IBN1(KROW),IBN2(KROW),NBN1(KCOL),
      *NBN2(KCOL),FN                               00052900
9127  FORMAT(I9,15X,A4,A1,8X,A4,A1,11X,F14.6)     00053200
7009  WRITE(NO,9500)                               00053300
      ND=0                                          00053400
9500  FORMAT(////)                                00053500
      GO TO 9101                                    00053600
C*****00053700
5433  IF(IPHASE-1) 8000,8000,5434                 00053800
8000  IPHASE=2                                     00053900
      IF(NINF) 8003,8003,8004                     00054000
8004  WRITE(NO,8005)                               00054100
8005  FORMAT(1X,'SOLUTION INFEASIBLE-END OF PHASE 1'////) 00054200
      GO TO 5434                                    00054300
8003  WRITE(NO,8002)                               00054400
8002  FORMAT(1X,'SOLUTION FEASIBLE-END OF PHASE 1'////) 00054500
      GO TO 5232                                    00054600
C*****00054700
C OUTPUT ROUTINE                                 00054800
C*****00054900
5434  WRITE(NO,6701)                               00055000
6701  FORMAT(//'***** THE CONTINUOUS SOLUTION IS *****'///) 00055100
      WRITE(NO,301) IT,FN                          00055200
301  FORMAT(1X,'ITERATION ',I5,5X,'OBJ FN ',F14.6////) 00055300
      WRITE(NO,302)                                00055400
302  FORMAT(3X,'BASIS VAR',5X,'UPPER BOUND? 0=NO/1=YES',15X,'AMOUNT',8X
      *,'UNIT PROFIT',///)                          00055500
      DO 3033 I=1,M                                00055700
3033  WRITE(NO,304) IBN1(I),IBN2(I),IFLAG(I),RQ(I),BP(I) 00055800
      304  FORMAT(7X,A4,A1,17X,I2,19X,F14.6,2X,F14.6//) 00055900
      WRITE(NO,9501)                                00056000
9501  FORMAT(1X,////)                              00056100
      WRITE(NO,305)                                00056200
305  FORMAT(2X,'NB VARIABLE',12X,'REDUCED COST',9X,'NB VARIABLE AT UPPE
      *R BOUND',9X,'UNIT PROFIT'///)                00056400
      DO 309 J=1,N                                  00056500
          IF(NBN1(J)-BLNK+NBN2(J)-BLNK) 311,309,311 00056600
311  IF(NFLAG(J)-1) 312,313,312                   00056700
313  WRITE(NO,314) NBN1(J),NBN2(J),PI(J),NBP(J)    00056800

```

```

GO TO 309
314 FORMAT(5X,A4,A1,8X,F14.6,25X,'YES',17X,F14.6//)
312 WRITE(NO,310) NBN1(J),NBN2(J),PI(J)
310 FORMAT(5X,A4,A1,8X,F14.6,25X,'NO',27X,'O.O'//)
309 CONTINUE
C
C TEST WHETHER THE CONTINUOUS SOLUTION FOUND IS OPTIMAL
C
DO 1004 I=1,M
IF(RB(I)-1.0) 1004,1005,1004
1005 X=RQ(I)
IF(ABS(X)-1.0D-7) 1004,1004,1006
1006 X=RQ(I)-1.00
IF(ABS(X)-1.0D-7) 1004,1004,1007
1004 CONTINUE
WRITE(NO,1008)
1008 FORMAT(1X,'THE CONTINUOUS SOLUTION FOUND IS OPTIMAL'//)
STOP
1007 WRITE(NO,1009)
1009 FORMAT(1X,'THE CONTINUOUS SOLUTION FOUND IS NOT OPTIMAL-CONTINUE WITH THE EXTREME POINT RANKING METHODOLOGY'//)
C
C SET NOPER=0, LSTE=1, KEXT=0, NSTOP=2000, NHALT= 200, NFOUND=0,
C NINT=0, NTEST=0
C
NOPER=0
LSTE=1
KEXT=0
NSTOP=2000
NHALT=200
NFOUND=0
NINT=0
NTEST=0
C
C SET THE FIRST RANKED EXTREME POINT SOLUTION EQUAL TO THE CONTINUOUS
C SOLUTION FOUND
C
KEXT=KEXT+1
C
C SET THE TEST SOLUTION AS WELL AS THE ADJACENT EXTREME POINT
C SOLUTION EQUAL TO THE FIRST RANKED EXTREME POINT SOLUTION
C
FTES=FN
L=LSTE
FSTE(L)=FN
IIND(L)=0
DO 1001 I=1,M
ITES1(I)=IBN1(I)
ITES2(I)=IBN2(I)
ISTE1(I,L)=IBN1(I)
ISTE2(I,L)=IBN2(I)
IFTES(I)=IFLAG(I)
IFSTE(I,L)=IFLAG(I)
BPTES(I)=BP(I)
BPSTE(I,L)=BP(I)
RQTES(I)=RQ(I)
RQSTE(I,L)=RQ(I)
RBTES(I)=RB(I)
RBSTE(I,L)=RB(I)
DO 1002 JR=1,N
BSTE(I,JR,L)=B(I,JR)
1002 BTES(I,JR)=B(I,JR)
1001 CONTINUE
DO 1003 JR=1,N
NTES1(JR)=NBN1(JR)
NTES2(JR)=NBN2(JR)
NSTE1(JR,L)=NBN1(JR)
NSTE2(JR,L)=NBN2(JR)
NFTES(JR)=NFLAG(JR)
NFSTE(JR,L)=NFLAG(JR)
NBPTES(JR)=NBP(JR)

```

```

NBPSTE(JR,L)=NBP(JR) 00064000
UBTES(JR)=UB(JR) 00064100
UBSTE(JR,L)=UB(JR) 00064200
PITES(JR)=PI(JR) 00064300
1003 PISTE(JR,L)=PI(JR) 00064400
LSTE=LSTE+1 00064500
C 00064600
C GENERATE THE EXTREME POINTS ADJACENT TO THE RANKED EXTREME POINT 00064700
C SOLUTION DETERMINED BY INTRODUCING NON BASIC VARIABLES ONE AT A TIME 00064800
C 00064900
889 DO 1010 J=1,N 00065000
C 00065100
C DISREGARD ARTIFICIAL VARIABLES 00065200
C 00065300
IF(NTES1(J)-BLNK+NTES2(J)-BLNK) 771,2979,771 00065400
771 KCOL=J 00065500
C*****00065600
C DETERMINE KEY ROW 00065700
C*****00065800
KROW1=0 00065900
KROW2=0 00066000
CJBAR=PITES(J) 00066100
THETA1=1.OE20 00066200
THETA2=1.OE20 00066300
DO 1011 I=1,M 00066400
IF(BTES(I,KCOL)) 1012,1011,1014 00066500
1012 RATIO2=(RBTES(I)-RQTES(I))/(-BTES(I,KCOL)) 00066600
IF(RATIO2-THETA2) 1015,1011,1011 00066700
1015 THETA2=RATIO2 00066800
KROW2=I 00066900
GO TO 1011 00067000
1014 RATIO1=RQTES(I)/BTES(I,KCOL) 00067100
IF(RATIO1-THETA1) 1016,1011,1011 00067200
1016 THETA1=RATIO1 00067300
KROW1=I 00067400
1011 CONTINUE 00067500
RMIN=THETA1 00067600
IF(RMIN-THETA2) 1017,1017,1018 00067700
1018 RMIN=THETA2 00067800
1017 IF(RMIN-UBTES(KCOL)) 1019,1019,1020 00067900
1020 RMIN=UBTES(KCOL) 00068000
1019 IF(RMIN-THETA1) 1021,1022,1021 00068100
1021 IF(RMIN-THETA2) 1023,1024,1023 00068200
C*****00068300
C RMIN=UB(KCOL).THEREFORE, X(KCOL) REMAINS NONBASIC BUT AT ITS UPPER 00068400
C BOUND. THUS IT MUST BE SUBSTITUTED AS X(KCOL)=UB(KCOL)-X'(KCOL). 00068500
C IF NFLAG(KCOL)=0 *OR* AS X'(KCOL)=UB(KCOL)-X(KCOL) IF NFLAG(KCOL)=1 00068600
C*****00068700
1023 ND=3 00068800
FTES=FTES-RMIN*PITES(KCOL) 00068900
9742 IF(FTES-FN) 9743,9743,2979 00069000
9743 IF(NINT-O) 4013,4012,4013 00069100
4012 X=FTES-ZBAR 00069200
IF(ABS(X)-1.OE-6) 4014,4014,4013 00069300
4014 WRITE(ND,4079) 00069400
4079 FORMAT(1X,'AN ADJACENT EXTREME POINT HAVING THE SAME FUNCTIONAL VA 00069500
*LUE AS THE INITIAL L.B. SOLUTION HAS BEEN FOUND'//) 00069600
NTEST=1 00069700
GO TO(1031,1031,1025),ND 00069800
4013 GO TO(4016,4016,4017),ND 00069900
4016 IF(FTES-ZBAR) 4010,4010,1031 00070000
4017 IF(FTES-ZBAR) 4010,4010,1025 00070100
4010 WRITE(ND,4011) FTES 00070200
4011 FORMAT(1X,'FTES (LESS THAN ZBAR) = ',F14.6//) 00070300
GO TO 2979 00070400
1025 IF(NFTES(KCOL)-1) 1026,1027,1026 00070500
1026 NFTES(KCOL)=1 00070600
GO TO 1028 00070700
1027 NFTES(KCOL)=0 00070800
1028 DO 1029 I=1,M 00070900
RQTES(I)=RQTES(I)-BTES(I,KCOL) 00071000

```



```

1029 BTES(I,KCOL)=-BTES(I,KCOL) 00071100
      IT=IT+1 00071200
      IF(IT-NSTOP) 477,1092,1092 00071300
477 PITES(KCOL)=-PITES(KCOL) 00071400
      GO TO 1041 00071500
C*****00071600
1022 KROW=KROW1 00071700
      ID=1 00071800
      ND=1 00071900
      FTES=FTES-RMIN*PITES(KCOL) 00072000
      GO TO 9742 00072100
1024 KROW=KROW2 00072200
      ID=2 00072300
      ND=2 00072400
      FTES=FTES-RMIN*PITES(KCOL) 00072500
      GO TO 9742 00072600
C*****00072700
C TRANSFORM 00072800
C DEVIDE BY PIVOT 00072900
C*****00073000
1031 PIVOT=BTES(KROW,KCOL) 00073100
      DO 1032 JR=1,N 00073200
1032 BTES(KROW, JR)=BTES(KROW, JR)/PIVOT 00073300
      RQTES(KROW)=RQTES(KROW)/PIVOT 00073400
      DO 1033 I=1,M 00073500
      IF(I-KROW) 1034,1033,1034 00073600
1034 RQTES(I)=RQTES(I)-RQTES(KROW)*BTES(I,KCOL) 00073700
      DO 1035 JR=1,N 00073800
      IF(JR-KCOL) 1036,1035,1036 00073900
1036 BTES(I, JR)=BTES(I, JR)-BTES(KROW, JR)*BTES(I,KCOL) 00074000
1035 CONTINUE 00074100
1033 CONTINUE 00074200
      DO 1037 I=1,M 00074300
1037 BTES(I,KCOL)=-BTES(I,KCOL)/PIVOT 00074400
      BTES(KROW,KCOL)=1.0/PIVOT 00074500
C*****00074600
C INTERCHANGE BASIS AND NONBASIS VARIABLES 00074700
C*****00074800
      RNM1=NTES1(KCOL) 00074900
      RNM2=NTES2(KCOL) 00075000
      NTES1(KCOL)=ITES1(KROW) 00075100
      NTES2(KCOL)=ITES2(KROW) 00075200
      ITES1(KROW)=RNM1 00075300
      ITES2(KROW)=RNM2 00075400
      LST=NBPTES(KCOL) 00075500
      NBPTES(KCOL)=BPTES(KROW) 00075600
      BPTES(KROW)=LST 00075700
      J1=NFTES(KCOL) 00075800
      NFTES(KCOL)=IFTES(KROW) 00075900
      IFTES(KROW)=J1 00076000
      XL=UBTES(KCOL) 00076100
      UBTES(KCOL)=RBTES(KROW) 00076200
      RBTES(KROW)=XL 00076300
      GO TO(1038,1039),ID 00076400
1038 IT=IT+1 00076500
      IF(IT-NSTOP) 1039,1092,1092 00076600
1039 DO 1040 JR=1,N 00076700
1040 PITES(JR)=PITES(JR)-CJBAR*BTES(KROW, JR) 00076800
      PITES(KCOL)=-CJBAR/PIVOT 00076900
      GO TO(1041,1025),ID 00077000
C*****00077100
C CHECK FOR ESSENTIAL ZERO 00077200
C*****00077300
1041 DO 1042 I=1,M 00077400
      DO 1042 JR=1,N 00077500
      X=BTES(I, JR) 00077600
      IF(ABS(X)-1.0D-7) 1043,1043,1042 00077700
1043 BTES(I, JR)=0. 00077800
1042 CONTINUE 00077900
C 00078000
C COMPARE THE BASES WITH ALREADY STORED ADJACENT EXTREME POINTS 00078100

```

```

C
      IF(LSTE-2) 1054,1055,1055
1055  L1=LSTE-1
      DO 1056 L=1,L1
      X=FMSTE(L)-FTES
      IF(ABS(X)-1.0D-7) 984,984,1056
984  DO 1057 I=1,M
      DO 1058 IR=1,M
      IF(ISTE1(I,L)-ITES1(IR)) 1058,1059,1058
1059  IF(ISTE2(I,L)-ITES2(IR)) 1058,1060,1058
1060  IF(IFSTE(I,L)-IFTES(IR)) 1061,1062,1063
1061  X=RQSTE(I,L)-1.0+RQTES(IR)
8008  IF(ABS(X)-1.0D-7) 1057,1057,1056
1062  X=RQSTE(I,L)-RQTES(IR)
      GO TO 8008
1063  X=RQTES(IR)-1.0+RQSTE(I,L)
      GO TO 8008
1058  CONTINUE
      DO 7411 JR=1,N
      IF(NTES1(JR)-BLNK+NTES2(JR)-BLNK) 7412,7411,7412
7412  IF(ISTE1(I,L)-NTES1(JR)) 7411,7413,7411
7413  IF(ISTE2(I,L)-NTES2(JR)) 7411,7414,7411
7414  IF(IFSTE(I,L)-NFTES(JR)) 7415,7416,7415
7415  X=RQSTE(I,L)-1.0
8007  IF(ABS(X)-1.0D-7) 1057,1057,1056
7416  X=RQSTE(I,L)-0.0
      GO TO 8007
7411  CONTINUE
1057  CONTINUE
      WRITE(NO,1065) IT,FTES
1065  FORMAT(1X,'IT= ',I4,5X,'FTES= ',F14.6,5X,'ADJACENT EXTREME POINT
      *S THE SAME AS ONE OF THE STORED POINTS'//)
      GO TO 2979
1056  CONTINUE
      WRITE(NO,1066) IT,FTES
1066  FORMAT(1X,'IT= ',I4,5X,'FTES= ',F14.6,5X,'ADJACENT EXTREME POINT
      *S NOT THE SAME AS ANY OF THE STORED POINTS'//)
C
C TEST FOR INTEGER FEASIBILITY
C
1054  DO 1067 I=1,M
      IF(RBTES(I)-1.0) 1067,1068,1067
1068  X=RQTES(I)
      IF(ABS(X)-1.0D-7) 1067,1067,1069
1069  X=RQTES(I)-1.0
      IF(ABS(X)-1.0D-7) 1067,1067,1070
1067  CONTINUE
      WRITE(NO,1071)
1071  FORMAT(1X,'THE ADJACENT EXTREME POINT GENERATED IS INTEGER FEASIBLE
      *E'//)
      IF(NINT=0) 6169,6170,6169
6170  IF(NTEST-1) 6169,6171,6169
6171  NINT=1
      WRITE(NO,6172)
6172  FORMAT(1X,'THE ADJACENT EXTREME POINT SOLUTION FOUND EQUAL TO THE
      *INITIAL L.B. SOLUTION IS INTEGER FEASIBLE'//)
      ZBAR=FTES
      GO TO 2979
C
C TEST WHETHER AN IMPROVEMENT OVER ZBAR HAS BEEN FOUND
C
6169  IF(ZBAR-FTES) 6173,1070,1070
6173  IF(NINT=0) 1072,6174,1072
6174  NINT=1
C
C UPDATE ZBAR
C
1072  IF(NFOUND=0) 4966,4473,4966
4473  NFOUND=1
4966  ZBAR=FTES
      WRITE(NO,6337) ZBAR

```

```

00078200
00078300
00078400
00078500
00078600
00078700
00078800
00078900
00079000
00079100
00079200
00079300
00079400
00079500
00079600
00079700
00079800
00079900
00080000
00080100
00080200
00080300
00080400
00080500
00080600
00080700
00080800
00080900
00081000
00081100
00081200
00081300
00081400
00081500
00081600
00081700
00081800
00081900
00082000
00082100
00082200
00082300
00082400
00082500
00082600
00082700
00082800
00082900
00083000
00083100
00083200
00083300
00083400
00083500
00083600
00083700
00083800
00083900
00084000
00084100
00084200
00084300
00084400
00084500
00084600
00084700
00084800
00084900
00085000
00085100
00085200

```

```

6337 FORMAT(1X,'AN IMPROVED INTEGER FEASIBLE SOLUTION IS FOUND WITH A V00085300
      *ALUE = ',F14.6//)                                00085400
C                                                         00085500
C STORE THE IMPROVED INTEGER FEASIBLE SOLUTION. MAKE USE OF DIFFERENT 00085600
C ARRAYS FOR THIS PURPOSE                                       00085700
C                                                         00085800
      FBAR=ZBAR                                               00085900
      DO 1073 I=1,M                                           00086000
        IBAR1(I)=ITES1(I)                                     00086100
        IBAR2(I)=ITES2(I)                                     00086200
        IFBAR(I)=IFTES(I)                                     00086300
        BPBAR(I)=BPTES(I)                                     00086400
        RQBAR(I)=RQTES(I)                                     00086500
        RBBAR(I)=RBTES(I)                                     00086600
      DO 1074 JR=1,N                                           00086700
1074   BBAR(I,JR)=BTES(I,JR)                                  00086800
1073   CONTINUE                                               00086900
      DO 1075 JR=1,N                                           00087000
        NBAR1(JR)=NTES1(JR)                                  00087100
        NBAR2(JR)=NTES2(JR)                                  00087200
        NFBAR(JR)=NFTES(JR)                                  00087300
        NBPBAR(JR)=NBPTES(JR)                                00087400
        UBBAR(JR)=UBTES(JR)                                  00087500
1075   PIBAR(JR)=PITES(JR)                                    00087600
C                                                         00087700
C SET NOPER=1                                                 00087800
C                                                         00087900
      NOPER=1                                                 00088000
C                                                         00088100
C STORE THE ADJACENT EXTREME POINT GENERATED IN THE LIST      00088200
C                                                         00088300
1070   L=LSTE                                                 00088400
      FSTE(L)=FTES                                             00088500
      IIND(L)=1                                               00088600
      DO 2071 I=1,M                                           00088700
        ISTE1(I,L)=ITES1(I)                                   00088800
        ISTE2(I,L)=ITES2(I)                                   00088900
        IFSTE(I,L)=IFTES(I)                                   00089000
        BPSTE(I,L)=BPTES(I)                                   00089100
        RQSTE(I,L)=RQTES(I)                                   00089200
        RBSTE(I,L)=RBTES(I)                                   00089300
      DO 2072 JR=1,N                                           00089400
2072   BSTE(I,JR,L)=BTES(I,JR)                                00089500
2071   CONTINUE                                               00089600
      DO 2073 JR=1,N                                           00089700
        NSTE1(JR,L)=NTES1(JR)                                 00089800
        NSTE2(JR,L)=NTES2(JR)                                 00089900
        NFSTE(JR,L)=NFTES(JR)                                 00090000
        NBPSTE(JR,L)=NBPTES(JR)                              00090100
        UBSTE(JR,L)=UBTES(JR)                                 00090200
2073   PISTE(JR,L)=PITES(JR)                                  00090300
      LSTE=LSTE+1                                             00090400
      WRITE(NO,6480) L                                         00090500
6480   FORMAT(1X,'THE ADJACENT EXTREME POINT STORED IS = ',I3//) 00090600
      IF(LSTE-400) 2979,2979,6476                             00090700
6476   IF(NFOUND-O) 6477,6478,6477                           00090800
6478   GO TO 3438                                              00090900
6477   WRITE(NO,6479)                                          00091000
6479   FORMAT(1X,'BEST INTEGER SOLUTION FOUND WITHIN THE FIRST FOUR HUNDRO091100
      *ED STORED ADJACENT EXTREME POINTS IS'//)              00091200
      GO TO 1085                                              00091300
C                                                         00091400
C SET TEST SOLUTION EQUAL TO THE RANKED EXTREME POINT          00091500
C SOLUTION UNDER CONSIDERATION                                00091600
C                                                         00091700
2979   FTES=FN                                               00091800
      DO 2074 I=1,M                                           00091900
        ITES1(I)=IBN1(I)                                       00092000
        ITES2(I)=IBN2(I)                                       00092100
        IFTES(I)=IFLAG(I)                                       00092200
        BPTES(I)=BP(I)                                         00092300

```

```

      RQTES(I)=RQ(I)
      RBTES(I)=RB(I)
      DO 2075 JR=1,N
2075   BTES(I,JR)=B(I,JR)
2074   CONTINUE
      DO 2076 JR=1,N
      NTES1(JR)=NBN1(JR)
      NTES2(JR)=NBN2(JR)
      NFTES(JR)=NFLAG(JR)
      NBPTE(JR)=NBP(JR)
      UBTE(JR)=UB(JR)
2076   PITES(JR)=PI(JR)
1010   CONTINUE
C
C IF AN IMPROVED INTEGER FEASIBLE SOLUTION HAS BEEN FOUND I.E.
C NOPER=1, TEST WHETHER ANY OF THE ADJACENT EXTREME POINTS FOUND
C CAN BE DISREGARDED
C
      IF(NOPER-1) 1080,1081,1080
1081   L=LSTE-1
      DO 1077 I=1,L
      IF(IIND(I)-O) 1078,1077,1078
1078   IF(FSTE(I)-ZBAR) 1079,1079,1077
1079   IIND(I)=O
1077   CONTINUE
      NOPER=O
C
C TEST WHETHER THERE ARE ANY POTENTIAL ADJACENT EXTREME POINTS
C FOR CONSIDERATION IN THE LIST; IF NONE STOP!
C
1080   L=LSTE-1
      DO 1082 I=1,L
      IF(IIND(I)-O) 1083,1082,1083
1082   CONTINUE
C
C NONE OF THE ADJACENT EXTREME POINTS IN THE STORED LIST
C HAVE THE POTENTIAL FOR IMPROVING THE INTEGER FEASIBLE SOLUTION
C CURRENTLY STORED. THUS THE OPTIMAL INTEGER SOLUTION IS FOUND.
C
      WRITE(NO,1084)
1084   FORMAT(1X,'THE OPTIMAL INTEGER SOLUTION HAS BEEN FOUND'//)
      GO TO 3441
C
C DETERMINE THE ADJACENT EXTREME POINT THAT HAS THE LARGEST
C OBJECTIVE FUNCTION VALUE
C
1083   L=LSTE-1
      LST=1.0D-7
      DO 1086 I=1,L
      IF(IIND(I)-O) 1087,1086,1087
1087   IF(FSTE(I)-LST) 1086,1086,1088
1088   LST=FSTE(I)
      ILARGE=I
1086   CONTINUE
C
C DISREGARD THE CHOSEN POINT FROM THE LIST OF ADJACENT EXTREME
C POINTS
C
      IIND(ILARGE)=O
C
C SET THE CHOSEN ADJACENT EXTREME POINT AS THE NEXT RANKED EXTREME
C POINT AS WELL AS THE TEST SOLUTION
C
      L=ILARGE
      FN=FSTE(L)
      FTES=FSTE(L)
      WRITE(NO,4015) FTES
4015   FORMAT(1X,'NEXT RANKED EXTREME POINT HAS A VALUE= ',F14.6//)
      DO 1089 I=1,M
      IBN1(I)=ISTE1(I,L)
      IBN2(I)=ISTE2(I,L)

```

```

00092400
00092500
00092600
00092700
00092800
00092900
00093000
00093100
00093200
00093300
00093400
00093500
00093600
00093700
00093800
00093900
00094000
00094100
00094200
00094300
00094400
00094500
00094600
00094700
00094800
00094900
00095000
00095100
00095200
00095300
00095400
00095500
00095600
00095700
00095800
00095900
00096000
00096100
00096200
00096300
00096400
00096500
00096600
00096700
00096800
00096900
00097000
00097100
00097200
00097300
00097400
00097500
00097600
00097700
00097800
00097900
00098000
00098100
00098200
00098300
00098400
00098500
00098600
00098700
00098800
00098900
00099000
00099100
00099200
00099300
00099400

```

```

      ITES1(I)=ISTE1(I,L)                                00099500
      ITES2(I)=ISTE2(I,L)                                00099600
      IFLAG(I)=IFSTE(I,L)                                00099700
      IFTES(I)=IFSTE(I,L)                                00099800
      BP(I)=BPSTE(I,L)                                    00099900
      BPTES(I)=BPSTE(I,L)                                00100000
      RQ(I)=RQSTE(I,L)                                    00100100
      RQTES(I)=RQSTE(I,L)                                00100200
      RB(I)=RBSTE(I,L)                                    00100300
      RBTES(I)=RBSTE(I,L)                                00100400
DO 1090 JR=1,N                                          00100500
      B(I,JR)=BSTE(I,JR,L)                                00100600
1090 BTES(I,JR)=BSTE(I,JR,L)                            00100700
1089 CONTINUE                                           00100800
DO 1091 JR=1,N                                          00100900
      NBN1(JR)=NSTE1(JR,L)                                00101000
      NBN2(JR)=NSTE2(JR,L)                                00101100
      NTES1(JR)=NSTE1(JR,L)                                00101200
      NTES2(JR)=NSTE2(JR,L)                                00101300
      NFLAG(JR)=NFSTE(JR,L)                                00101400
      NFTES(JR)=NFSTE(JR,L)                                00101500
      NBP(JR)=NBPSTE(JR,L)                                00101600
      NBPTE(JR)=NBPSTE(JR,L)                              00101700
      UB(JR)=UBSTE(JR,L)                                    00101800
      UBTES(JR)=UBSTE(JR,L)                                00101900
      PI(JR)=PISTE(JR,L)                                    00102000
1091 PITES(JR)=PISTE(JR,L)                              00102100
      KEXT=KEXT+1                                          00102200
      WRITE(NO,6488) KEXT                                  00102300
6488 FORMAT(1X,'THE # OF THE RANKED EXTREME POINT IS = ',I3//) 00102400
      IF(KEXT-NHALT) 889,3023,3023                        00102500
3023 IF(NFOUND-O) 3437,3438,3437                        00102600
3438 WRITE(NO,3439)                                       00102700
3439 FORMAT(1X,'NO BETTER INTEGER SOLUTION HAS BEEN FOUND SO FAR'//) 00102800
      GO TO 197                                           00102900
3437 WRITE(NO,3024)                                       00103000
3024 FORMAT(1X,'BEST INTEGER SOLUTION FOUND WITHIN THE FIRST TWO HUNDREO00103100
      *D RANKED EXTREME POINTS IS'//)                    00103200
      GO TO 1085                                          00103300
1092 IF(NFOUND-O) 3440,3438,3440                        00103400
3440 WRITE(NO,1093)                                       00103500
1093 FORMAT(1X,'BEST INTEGER SOLUTION FOUND WITHIN THE FIRST TWO THOUSAO00103600
      *ND ITERATIONS IS'//)                              00103700
      GO TO 1085                                          00103800
C*****00103900
C FINAL OUTPUT ROUTINE                                00104000
C*****00104100
3441 IF(NFOUND-O) 1085,3438,1085                        00104200
1085 WRITE(NO,1094)                                       00104300
1094 FORMAT(1X,'***** THE FINAL RESULTS ARE *****'//) 00104400
      WRITE(NO,1095) IT,FBAR                              00104500
1095 FORMAT(1X,' ITERATION ',I5,5X,' OBJ FN ',F14.6////). 00104600
      WRITE(NO,1096)                                       00104700
1096 FORMAT(3X,'BASIS VAR',5X,' UPPER BOUND? O=NO/1=YES',15X,' AMOUNT',8XO0104800
      *, 'UNIT PROFIT',//)                                00104900
      DO 1097 IK=1,M                                      00105000
1097 WRITE(NO,1098) IBAR1(IK),IBAR2(IK),IFBAR(IK),RQBAR(IK),BPBAR(IK) 00105100
1098 FORMAT(7X,A4,A1,17X,I2,19X,F14.6,2X,F14.6//)        00105200
      WRITE(NO,1099)                                       00105300
1099 FORMAT(1X,////)                                       00105400
      WRITE(NO,1100)                                       00105500
1100 FORMAT(2X,'NB VARIABLE',12X,'REDUCED COST',9X,'NB VARIABLE AT UPPEO0105600
      *R BOUND',9X,'UNIT PROFIT'//)                    00105700
      DO 1101 JF=1,N                                       00105800
          IF(NBAR1(JF)-BLNK+NBAR2(JF)-BLNK) 1102,1101,1102 00105900
1102 IF(NFBAR(JF)-1) 1103,1104,1103                    00106000
1104 WRITE(NO,1105) NBAR1(JF),NBAR2(JF),PIBAR(JF),NBPBAR(JF) 00106100
1105 FORMAT(5X,A4,A1,8X,F14.6,25X,'YES',17X,F14.6//)    00106200
      GO TO 1101                                          00106300
1103 WRITE(NO,1106) NBAR1(JF),NBAR2(JF),PIBAR(JF)      00106400
1106 FORMAT(5X,A4,A1,8X,F14.6,25X,'NO',27X,'O.O'//)    00106500

```

1101 CONTINUE
197 STOP
END

00106600
00106700
00106800

```

C*****00000100
C                                     00000200
C   THIS PROGRAM PERFORMS THE ALGORITHMIC STEPS FOR THE REVISED MODEL 00000300
C     BASED ON THE EXTREME POINT RANKING TECHNIQUE                       00000400
C                                     00000500
C     THE PROBLEM IS SOLVED IN THE MAIN MEMORY AND IN DIRECT           00000600
C     ACCESS FILES CONTAINED IN TEMPORARY DISC STORAGE AREA           00000700
C                                     00000800
C     THE REVISED MODEL IS A 0-1 INTEGER BINARY LINEAR PROGRAMMING 00000900
C     MODEL. AS SUCH, COMPUTATIONS ARE PRIMARILY BASED ON THE          00001000
C     PRIMAL SIMPLEX METHOD FOR BOUNDED VARIABLES                      00001100
C                                     00001200
C           WRITTEN BY LOGENDRAN RASARATNAM                            00001300
C     SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT                 00001400
C           OKLAHOMA STATE UNIVERSITY                                 00001500
C                                     00001600
C     DISSERTATION ADVISER: DR. M. PALMER TERRELL                    00001700
C     VERSION 1 -- AUGUST, 1984                                       00001800
C                                     00001900
C*****00002000
C                                     00002100
C* GENERAL STRUCTURE AND INPUT REQUIREMENTS                          00002200
C* ONLY A SINGLE MAIN PROGRAM IS USED. NO SUBROUTINES REQUIRED.      00002300
C                                     00002400
C DESCRIPTION OF INPUT PARAMETERS                                     00002500
C                                     00002600
C M1      = MAXIMUM NUMBER OF ROWS IN THE REVISED MODEL, EXCLUDING THE 00002700
C           OBJECTIVE ROW                                             00002800
C N1      = MAXIMUM NUMBER OF STRUCTURAL VARIABLES IN THE REVISED MODEL 00002900
C RNM1    = FIRST FOUR CHARACTERS OF ROW (CONSTRAINT) NAME           00003000
C RNM2    = FIFTH (AND FINAL) CHARACTER OF ROW (CONSTRAINT) NAME     00003100
C CLNM1   = FIRST FOUR CHARACTERS OF COLUMN NAME                     00003200
C CLNM2   = FIFTH (AND FINAL) CHARACTER OF COLUMN NAME               00003300
C CDID    = CARD GROUP HEADER                                         00003400
C         PUNCHED RO FOR ROW IDENTIFICATION HEADER                  00003500
C         PUNCHED MA FOR MATRIX HEADER                              00003600
C         PUNCHED FI FOR RHS HEADER                                00003700
C         PUNCHED EO FOR PROBLEM DELIMITER                          00003800
C LGE IDENTIFIES TYPE OF CONSTRAINT                                  00003900
C   + FOR LESS THAN OR EQUAL TO                                     00004000
C   O FOR EQUAL TO                                                 00004100
C VALUE.  = NUMERICAL VALUE OF MATRIX ELEMENT, OR RHS VALUE         00004200
C ZBAR    = THE LOWER BOUND DETERMINED FROM THE HEURISTIC ALGORITHM 00004300
C                                     00004400
C DESCRIPTION OF VARIABLES                                         00004500
C                                     00004600
C IBN1    = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE RANKED     00004700
C           EXTREME POINT CURRENTLY CONSIDERED                       00004800
C IBN2    = FIFTH CHARACTER OF THE ROW NAME FOR THE RANKED           00004900
C           EXTREME POINT CURRENTLY CONSIDERED                       00005000
C ITES1   = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE ADJACENT 00005100
C           EXTREME POINT CURRENTLY GENERATED                       00005200
C ITES2   = FIFTH CHARACTER OF THE ROW NAME FOR THE ADJACENT         00005300
C           EXTREME POINT CURRENTLY GENERATED                       00005400
C ISTE1   = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE ADJACENT 00005500
C           EXTREME POINT STORED                                     00005600
C ISTE2   = FIFTH CHARACTER OF THE ROW NAME FOR THE ADJACENT         00005700
C           EXTREME POINT STORED                                     00005800
C IBAR1   = FIRST FOUR CHARACTERS OF THE ROW NAME FOR THE           00005900
C           OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR               00006000
C IBAR2   = FIFTH CHARACTER OF THE ROW NAME FOR THE OPTIMAL/        00006100
C           BEST BINARY SOLUTION FOUND SO FAR                       00006200
C         O - BASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY 00006300
C IFLAG = {   CONSIDERED IS NOT AT ITS UPPER BOUND                 00006400
C           1 - BASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY 00006500
C             CONSIDERED IS AT ITS UPPER BOUND                     00006600
C           O - BASIC VARIABLE OF THE ADJACENT EXTREME POINT CURRENTLY 00006700
C IFTES = {   GENERATED IS NOT AT ITS UPPER BOUND                 00006800
C           1 - BASIC VARIABLE OF THE ADJACENT EXTREME POINT CURRENTLY 00006900
C             GENERATED IS AT ITS UPPER BOUND                       00007000
C           O - BASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED 00007100

```

```

C IFSTE = {          IS NOT AT ITS UPPER BOUND          00007200
C           1 - BASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED 00007300
C           IS AT ITS UPPER BOUND          00007400
C           0 - BASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION 00007500
C IFBAR = {          IS NOT AT ITS UPPER BOUND          00007600
C           1 - BASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION 00007700
C           IS AT ITS UPPER BOUND          00007800
C BP      = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE 00007900
C           RANKED EXTREME POINT CURRENTLY CONSIDERED          00008000
C BPTES = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE 00008100
C           ADJACENT EXTREME POINT CURRENTLY GENERATED          00008200
C BPSTE = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE 00008300
C           ADJACENT EXTREME POINT STORED          00008400
C BPBAR = OBJECTIVE FUNCTION COEFFICIENT FOR THE BASIC VARIABLE OF THE 00008500
C           OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR          00008600
C RQ      = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE RANKED 00008700
C           EXTREME POINT CURRENTLY CONSIDERED          00008800
C RQTES = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE ADJACENT 00008900
C           EXTREME POINT CURRENTLY GENERATED          00009000
C RQSTE = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE ADJACENT 00009100
C           EXTREME POINT STORED          00009200
C RQBAR = RIGHT HAND SIDE VALUE FOR THE BASIC VARIABLE OF THE 00009300
C           OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR          00009400
C RB      = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE 00009500
C           RANKED EXTREME POINT CURRENTLY CONSIDERED          00009600
C RBTES = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE 00009700
C           ADJACENT EXTREME POINT CURRENTLY GENERATED          00009800
C RBSTE = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE 00009900
C           ADJACENT EXTREME POINT STORED          00010000
C RBBAR = VALUE OF THE UPPER BOUND FOR THE BASIC VARIABLE OF THE 00010100
C           OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR          00010200
C B       = M1*N1 MATRIX REPRESENTING THE RANKED EXTREME POINT 00010300
C           CURRENTLY CONSIDERED          00010400
C BTES = M1*N1 MATRIX REPRESENTING THE ADJACENT EXTREME POINT 00010500
C           CURRENTLY GENERATED          00010600
C BSTE = M1*N1 MATRIX REPRESENTING THE ADJACENT EXTREME POINT 00010700
C           STORED          00010800
C BBAR = M1*N1 MATRIX REPRESENTING THE OPTIMAL/BEST 00010900
C           BINARY SOLUTION FOUND SO FAR          00011000
C NBN1   = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE RANKED 00011100
C           EXTREME POINT CURRENTLY CONSIDERED          00011200
C NBN2   = FIFTH CHARACTER OF THE COLUMN NAME FOR THE RANKED 00011300
C           EXTREME POINT CURRENTLY CONSIDERED          00011400
C NTES1  = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE ADJACENT 00011500
C           EXTREME POINT CURRENTLY GENERATED          00011600
C NTES2  = FIFTH CHARACTER OF THE COLUMN NAME FOR THE ADJACENT 00011700
C           EXTREME POINT CURRENTLY GENERATED          00011800
C NSTE1  = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE ADJACENT 00011900
C           EXTREME POINT STORED          00012000
C NSTE2  = FIFTH CHARACTER OF THE COLUMN NAME FOR THE ADJACENT 00012100
C           EXTREME POINT STORED          00012200
C NBAR1  = FIRST FOUR CHARACTERS OF THE COLUMN NAME FOR THE 00012300
C           OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR          00012400
C NBAR2  = FIFTH CHARACTER OF THE COLUMN NAME FOR THE OPTIMAL/ 00012500
C           BEST BINARY SOLUTION FOUND SO FAR          00012600
C NFLAG = {          0 - NONBASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY 00012700
C           CONSIDERED IS NOT AT ITS UPPER BOUND          00012800
C           1 - NONBASIC VARIABLE OF THE RANKED EXTREME POINT CURRENTLY 00012900
C           CONSIDERED IS AT ITS UPPER BOUND          00013000
C           0 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT 00013100
C           CURRENTLY GENERATED IS NOT AT ITS UPPER BOUND 00013200
C NFTES = {          1 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT 00013300
C           CURRENTLY GENERATED IS AT ITS UPPER BOUND 00013400
C           0 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED 00013500
C           IS NOT AT ITS UPPER BOUND          00013600
C NFSTE = {          1 - NONBASIC VARIABLE OF THE ADJACENT EXTREME POINT STORED 00013700
C           IS AT ITS UPPER BOUND          00013800
C           0 - NONBASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION 00013900
C NFBAR = {          IS NOT AT ITS UPPER BOUND          00014000
C           1 - BASIC VARIABLE OF THE OPTIMAL/BEST BINARY SOLUTION 00014100
C           IS AT ITS UPPER BOUND          00014200

```



```

C NBP   = OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE      00014300
C       OF THE RANKED EXTREME POINT CURRENTLY CONSIDERED                00014400
C NBPTE= OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE      00014500
C       OF THE ADJACENT EXTREME POINT CURRENTLY GENERATED              00014600
C NBPSTE= OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE      00014700
C       OF THE ADJACENT EXTREME POINT STORED                           00014800
C NBPBAR= OBJECTIVE FUNCTION COEFFICIENT FOR THE NONBASIC VARIABLE      00014900
C       OF THE OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR                00015000
C UB    = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE     00015100
C       RANKED EXTREME POINT CURRENTLY CONSIDERED                      00015200
C UBTES = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE     00015300
C       ADJACENT EXTREME POINT CURRENTLY GENERATED                    00015400
C UBSTE = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE     00015500
C       ADJACENT EXTREME POINT STORED                                  00015600
C UBBAR = VALUE OF THE UPPER BOUND FOR THE NONBASIC VARIABLE OF THE     00015700
C       OPTIMAL/BEST BINARY SOLUTION FOUND SO FAR                      00015800
C PI    = SHADOW PRICES FOR THE RANKED EXTREME POINT CURRENTLY          00015900
C       CONSIDERED                                                      00016000
C PITES = SHADOW PRICES FOR THE ADJACENT EXTREME POINT CURRENTLY        00016100
C       GENERATED                                                         00016200
C PISTE = SHADOW PRICES FOR THE ADJACENT EXTREME POINT STORED           00016300
C PIBAR = SHADOW PRICES FOR THE OPTIMAL/BEST BINARY SOLUTION            00016400
C       FOUND SO FAR                                                      00016500
C FN    = OBJECTIVE FUNCTION VALUE FOR THE RANKED EXTREME POINT          00016600
C       CURRENTLY CONSIDERED                                              00016700
C FTES  = OBJECTIVE FUNCTION VALUE FOR THE ADJACENT EXTREME POINT        00016800
C       CURRENTLY GENERATED                                                00016900
C FSTE  = OBJECTIVE FUNCTION VALUE FOR THE ADJACENT EXTREME POINT        00017000
C       STORED                                                             00017100
C FBAR  = OBJECTIVE FUNCTION VALUE FOR THE OPTIMAL/BEST                   00017200
C       BINARY SOLUTION FOUND SO FAR                                       00017300
C       0 - ADJACENT EXTREME POINT HAS NO POTENTIAL                     00017400
C IIND  = {   FOR BEING CONSIDERED IN THE FUTURE                         00017500
C       1 - ADJACENT EXTREME POINT HAS POTENTIAL                         00017600
C       FOR BEING CONSIDERED IN THE FUTURE                               00017700
C       0 - AN IMPROVED BINARY FEASIBLE SOLUTION HAS NOT BEEN           00017800
C NOPER = {   FOUND IN THE CURRENT CYCLE                                 00017900
C       1 - AN IMPROVED BINARY FEASIBLE SOLUTION HAS BEEN               00018000
C       FOUND IN THE CURRENT CYCLE                                       00018100
C IT    = TOTAL NUMBER OF SIMPLEX ITERATIONS                             00018200
C LSTE  = TOTAL NUMBER OF DISTINCT ADJACENT EXTREME POINTS GENERATED     00018300
C                                             00018400
C *****00018500
C   INTEGER RNM1,RNM2,CLNM1,CLNM2,BLNK,POS,CDID,RO,MA,FI,EO,
C   *IBN1(20),IBN2(20),ITES1(20),ITES2(20),IBAR1(20),IBAR2(20),
C   *ISTE1(20),ISTE2(20),NBN1(110),NBN2(110),NTES1(110),
C   *NTES2(110),NBAR1(110),NBAR2(110),NSTE1(110),NSTE2(110),
C   *IFLAG(20),IFTES(20),IFBAR(20),IFSTE(20),NFLAG(110),NFTES(110),
C   *NFBAR(110),NFSTE(110),IIND(5000)
C   REAL*8 PIVOT,LST, FN,CJBAR,VALUE,BP(20),BPTES(20),BPBAR(20),
C   *BPSTE(20),RQ(20),RQTES(20),RQBAR(20),RQSTE(20),
C   *B(20,110),BTES(20,110),BBAR(20,110),BSTE(20,110),
C   *PI(110),PITES(110),PIBAR(110),PISTE(110),NBP(110),
C   *NBPTES(110),NBPBAR(110),NBPSTE(110),RB(20),RBTES(20),
C   *RBBAR(20),RBSTE(20),XPI(110),UB(110),UBTES(110),
C   *UBBAR(110),UBSTE(110),FSTE(5000),FBAR,FTES,ZBAR,THETA1,
C   *THETA2,RATIO1,RATIO2,RMIN,SAVE,XL,X
C   DATA RO,MA,POS,FI,EO,BLNK/2HRO,2HMA,1H+,2HFI,2HEO,4H----/
C *****00020100
C INITIALIZE
C *****00020200
C *****00020300
C   M=0
C   N=1
C   NROWS=0
C   NLE=0
C   NEQ=0
C   NEL=0
C   NRHS=0
C   NCOLS=0
C   NI=5
C   NO=30
C *****00021300

```

```

      READ(NI,1) M1,N1
1      FORMAT(2I5)
C*****
C CLEAR MATRIX TO ZERO
C*****
      DO 12 I=1,M1
      DO 12 J=1,N1
12     B(I,J)=0
C*****
C READ FIRST CARD-SHOULD BE ROWID
C*****
      READ(NI,2) CDID
      FORMAT(1X,A2)
      IF(CDID-RO) 3,680,3
3      WRITE(NO,3333)
3333  FORMAT(//'ROWID CARD MISSING'//)
3334  STOP
C*****
C READ AND STORE ROWID CARDS INCLUDING DUMMY READ
C FOR OBJECTIVE ROW NAME
C GENERATE POSITIVE AND NEGATIVE SLACKS AS REQUIRED
C*****
680  READ(NI,681)
681  FORMAT(80X)
101  READ(NI,102) CDID,LGE,RNM1,RNM2
102  FORMAT(1X,A2,8X,A1,1X,A4,A1)
      IF(CDID-MA) 103,104,103
103  M=M+1
      NROWS=NROWS+1
      IF(LGE-POS) 105,106,105
105  IBN1(M)=RNM1
      IBN2(M)=RNM2
      NEQ=NEQ+1
      BP(M)=-1.0
      RB(M)=1.0D20
      GO TO 101
106  IBN1(M)=RNM1
      IBN2(M)=RNM2
      NLE=NLE+1
      BP(M)=0.0
      RB(M)=1.0D20
      GO TO 101
C*****
C READ AND STORE FIRST MATRIX ELEMENT
C*****
104  READ(NI,195) CDID,CLNM1,CLNM2,RNM1,RNM2,VALUE
195  FORMAT(1X,A2,4X,A4,A1,1X,A4,A1,1X,D20.10)
      GO TO 119
109  IF(NBN1(N)-CLNM1) 111,600,111
600  IF(NBN2(N)-CLNM2) 111,112,111
112  DO 113 I=1,M
      IF(IBN1(I)-RNM1) 113,602,113
602  IF(IBN2(I)-RNM2) 113,116,113
113  CONTINUE
      WRITE(NO,8113)
8113  FORMAT(//'INCORRECT INGREDIENT CARD'//)
      STOP
116  B(I,N)=VALUE
C*****
C READ AND STORE MATRIX ELEMENTS
C*****
117  READ(NI,195) CDID,CLNM1,CLNM2,RNM1,RNM2,VALUE
      NEL=NEL+1
      GO TO 109
111  N=N+1
      NEL=NEL-1
      NCOLS=NCOLS+1
      IF(CDID-FI) 119,190,119
119  NBN1(N)=CLNM1
      NBN2(N)=CLNM2
      NBP(N)=VALUE

```

```

        UB(N)=1.0                                00028500
        GO TO 117                                00028600
C*****                                        00028700
C READ AND STORE RHS ELEMENTS                    00028800
C*****                                        00028900
190 DO 191 I=1,M                                00029000
191 RQ(I)=0.                                     00029100
120 READ(NI,121) CDID,RNM1,RNM2,VALUE           00029200
121 FORMAT(1X,A2,10X,A4,A1,1X,D20.10)         00029300
        IF(CDID-EO) 122,193,122                00029400
122 DO 124 I=1,M                                00029500
        IF(IBN1(I)-RNM1) 124,610,124           00029600
610 IF(IBN2(I)-RNM2) 124,125,124              00029700
124 CONTINUE                                    00029800
        WRITE(NO,8124) RNM1,RNM2              00029900
8124 FORMAT(///'NO ROW NAME FOR',A4,A1,///)    00030000
        STOP                                    00030100
125 RQ(I)=VALUE                                  00030200
        NRHS=NRHS+1                            00030300
        GO TO 120                               00030400
193 READ(NI,339) ZBAR                            00030500
339 FORMAT(D20.10)                              00030600
        WRITE(NO,340) ZBAR                    00030700
340 FORMAT(1X,'ZBAR= ',D20.10//)              00030800
        N=N-1                                    00030900
        WRITE(NO,551) NROWS,NCOLS,NLE,NEQ,NRHS,NEL 00031000
551 FORMAT(1X,'ROWS= ',I3//1X,'COLS= ',I4//1X,'LE ROWS= ',I3//
*1X,'EQ ROWS= ',I3//1X,'RHS = ',I3//1X,'NONZERO MATRIX ELEMENTS=
* ',I6//)                                        00031200
C*****                                        00031300
C*****                                        00031400
C BLANK OUT ARTIFICIAL NAMES                    00031500
C*****                                        00031600
        DO 10 I=1,M                             00031700
        IF(BP(I)+1.0) 10,11,10                 00031800
11 IBN1(I)=BLNK                                 00031900
        IBN2(I)=BLNK                           00032000
10 CONTINUE                                    00032100
C*****                                        00032200
C ACCUMULATE COUNT OF INFEASIBILITIES           00032300
C*****                                        00032400
        NINF=0                                    00032500
        DO 6000 I=1,M                          00032600
        IF(BP(I)) 6001,6000,6000              00032700
6001 NINF=NINF+1                              00032800
6000 CONTINUE                                  00032900
C*****                                        00033000
C GENERATE INDICATORS FOR MINIMIZATION OF INFEASIBILITY 00033100
C*****                                        00033200
        DO 6101 J=1,N                          00033300
        XPI(J)=0.                              00033400
        DO 6101 I=1,M                          00033500
        IF(BP(I)) 6102,6101,6101              00033600
6102 XPI(J)=XPI(J)-B(I,J)                    00033700
6101 CONTINUE                                  00033800
        DO 6002 I=1,M                          00033900
6002 BP(I)=0.                                  00034000
        IPHASE=1                                00034100
        DO 1203 I=1,M                          00034200
1203 IFLAG(I)=0                                00034300
        DO 1204 J=1,N                          00034400
1204 NFLAG(J)=0                                00034500
C*****                                        00034600
C MAIN ROUTINE                                  00034700
C*****                                        00034800
        IT=0                                    00034900
C*****                                        00035000
C CALCULATE SHADOW PRICES                      00035100
C*****                                        00035200
5232 DO 194 J=1,N                              00035300
        PI(J)=-NBP(J)                          00035400
        DO 194 I=1,M                          00035500

```

```

194 PI(J)=PI(J)+BP(I)*B(I,J) 00035600
C***** 00035700
C SELECT BEST NONBASIS VECTOR 00035800
C***** 00035900
9101 LST=-1.0D-7 00036000
      KCOL=0 00036100
      GO TO (751,552),IPHASE 00036200
751 IF(NINF) 5433,5433,552 00036300
552 DO 9102 J=1,N 00036400
C***** 00036500
C IGNORE ARTIFICIAL VARIABLES 00036600
C***** 00036700
      IF(NBN1(J)-BLNK+NBN2(J)-BLNK) 651,9102,651 00036800
651 GO TO (6003,6004),IPHASE 00036900
6003 IF(XPI(J)-LST) 6005,9102,9102 00037000
6005 KCOL=J 00037100
      LST=XPI(J) 00037200
      GO TO 9102 00037300
6004 IF(PI(J)-LST) 9103,9102,9102 00037400
9103 KCOL=J 00037500
      LST=PI(J) 00037600
9102 CONTINUE 00037700
      IF(KCOL) 5433,5433,9104 00037800
C***** 00037900
C DETERMINE KEY ROW 00038000
C***** 00038100
9104 KROW1=0 00038200
      KROW2=0 00038300
      CJBAR=LST 00038400
      THETA1=1.0D20 00038500
      THETA2=1.0D20 00038600
      DO 9105 I=1,M 00038700
        IF(B(I,KCOL)) 9106,9105,9107 00038800
9106 RATIO2=(RB(I)-RQ(I))/(-B(I,KCOL)) 00038900
      IF(RATIO2-THETA2) 9108,9105,9105 00039000
9108 THETA2=RATIO2 00039100
      KROW2=I 00039200
      GO TO 9105 00039300
9107 RATIO1=RQ(I)/B(I,KCOL) 00039400
      IF(RATIO1-THETA1) 9109,9105,9105 00039500
9109 THETA1=RATIO1 00039600
      KROW1=I 00039700
9105 CONTINUE 00039800
      RMIN=THETA1 00039900
      IF(RMIN-THETA2) 9110,9110,9111 00040000
9111 RMIN=THETA2 00040100
9110 IF(RMIN-UB(KCOL)) 9112,9112,9113 00040200
9113 RMIN=UB(KCOL) 00040300
9112 IF(RMIN-THETA1) 9114,9115,9114 00040400
9114 IF(RMIN-THETA2) 9116,9117,9116 00040500
C***** 00040600
C RMIN=UB(KCOL). THEREFORE, X(KCOL) REMAINS NONBASIC BUT AT ITS UPPER 00040700
C BOUND. THUS IT MUST BE SUBSTITUTED AS X(KCOL)=UB(KCOL)-X'(KCOL). 00040800
C IF NFLAG(KCOL)=0 *OR* AS X'(KCOL)=UB(KCOL)-X(KCOL) IF NFLAG(KCOL)=1 00040900
C***** 00041000
9116 ND=3 00041100
9223 IF(NFLAG(KCOL)-1) 677,678,677 00041200
677 NFLAG(KCOL)=1 00041300
      GO TO 679 00041400
678 NFLAG(KCOL)=0 00041500
679 DO 9118 I=1,M 00041600
      RQ(I)=RQ(I)-B(I,KCOL) 00041700
9118 B(I,KCOL)=-B(I,KCOL) 00041800
      IT=IT+1 00041900
      GO TO (9119,9120),IPHASE 00042000
9119 PI(KCOL)=-PI(KCOL) 00042100
      XPI(KCOL)=-XPI(KCOL) 00042200
      GO TO 9121 00042300
9120 PI(KCOL)=-PI(KCOL) 00042400
      GO TO 9121 00042500
C***** 00042600

```

```

9115   KROW=KROW1                                00042700
        ID=1                                      00042800
        ND=1                                      00042900
        GO TO 900                                00043000
9117   KROW=KROW2                                00043100
        ID=2                                      00043200
        ND=2                                      00043300
C*****                                          00043400
C TRANSFORM                                     00043500
C DIVIDE BY PIVOT                               00043600
C*****                                          00043700
900    PIVOT=B(KROW,KCOL)                        00043800
        DO 9128 J=1,N                             00043900
9128   B(KROW,J)=B(KROW,J)/PIVOT                 00044000
        RQ(KROW)=RQ(KROW)/PIVOT                  00044100
        DO 9129 I=1,M                             00044200
          IF(I-KROW) 9130,9129,9130               00044300
9130   RQ(I)=RQ(I)-RQ(KROW)*B(I,KCOL)            00044400
        DO 4444 J=1,N                             00044500
          IF(J-KCOL) 9131,4444,9131               00044600
9131   B(I,J)=B(I,J)-B(KROW,J)*B(I,KCOL)         00044700
4444   CONTINUE                                  00044800
9129   CONTINUE                                  00044900
        DO 9132 I=1,M                             00045000
9132   B(I,KCOL)=-B(I,KCOL)/PIVOT                00045100
        B(KROW,KCOL)=1.0/PIVOT                   00045200
C*****                                          00045300
C INTERCHANGE BASIS AND NONBASIS VARIABLES      00045400
C*****                                          00045500
        RNM1=NB1(KCOL)                            00045600
        RNM2=NB2(KCOL)                            00045700
        NB1(KCOL)=IB1(KROW)                       00045800
        NB2(KCOL)=IB2(KROW)                       00045900
        IB1(KROW)=RNM1                            00046000
        IB2(KROW)=RNM2                            00046100
        LST=NB(KCOL)                              00046200
        NBP(KCOL)=BP(KROW)                        00046300
        BP(KROW)=LST                              00046400
        J1=NFLAG(KCOL)                            00046500
        NFLAG(KCOL)=IFLAG(KROW)                   00046600
        IFLAG(KROW)=J1                            00046700
        XL=UB(KCOL)                                00046800
        UB(KCOL)=RB(KROW)                         00046900
        RB(KROW)=XL                               00047000
        GO TO (801,802),ID                         00047100
801    IT=IT+1                                    00047200
802    IF(NB1(KCOL)-BLNK+NB2(KCOL)-BLNK) 9138,9133,9138 00047300
9133   NINF=NINF-1                                00047400
9138   GO TO (9134,9135),IPHASE                   00047500
9134   SAVE=PI(KCOL)                              00047600
        DO 9136 J=1,N                             00047700
          PI(J)=PI(J)-SAVE*B(KROW,J)              00047800
9136   XPI(J)=XPI(J)-CJBAR*B(KROW,J)             00047900
          PI(KCOL)=-SAVE/PIVOT                    00048000
          XPI(KCOL)=-CJBAR/PIVOT                  00048100
        GO TO (9121,9116),ID                       00048200
9135   DO 9137 J=1,N                             00048300
9137   PI(J)=PI(J)-CJBAR*B(KROW,J)               00048400
          PI(KCOL)=-CJBAR/PIVOT                   00048500
        GO TO (9121,9223),ID                       00048600
C*****                                          00048700
C COMPUTE OBJECTIVE FUNCTION                    00048800
C*****                                          00048900
9121   FN=0.                                       00049000
        DO 9122 I=1,M                             00049100
          IF(IFLAG(I)-1) 9123,9124,9123           00049200
9124   FN=FN+BP(I)*(1.-RQ(I))                     00049300
        GO TO 9122                                 00049400
9123   FN=FN+BP(I)*RQ(I)                          00049500
9122   CONTINUE                                  00049600
        DO 9125 J=1,N                             00049700

```

```

          IF(NFLAG(J)-1) 9125,9126,9125                                00049800
9126     FN=FN+NBP(J)                                                00049900
9125     CONTINUE                                                    00050000
C*****                                                                00050100
C CHECK FOR ESSENTIAL ZERO                                          00050200
C*****                                                                00050300
          DO 6111 I=1,M                                              00050400
          DO 6111 J=1,N                                              00050500
              X=B(I,J)                                              00050600
              IF(ABS(X)-1.0D-7) 6112,6112,6111                      00050700
6112     B(I,J)=0.                                                  00050800
6111     CONTINUE                                                    00050900
C*****                                                                00051000
C LOG ITERATION                                                    00051100
C*****                                                                00051200
          GO TO (7001,7002,7003),ND                                  00051300
7001     WRITE(NO,7004)                                              00051400
7004     FORMAT(2X,'OPERATION TYPE 1'//)                             00051500
          GO TO 7010                                                  00051600
7002     WRITE(NO,7005)                                              00051700
7005     FORMAT(2X,'OPERATION TYPE 2'//)                             00051800
          GO TO 7010                                                  00051900
7003     WRITE(NO,7006)                                              00052000
7006     FORMAT(2X,'OPERATION TYPE 3 = NO BASIS CHANGE'//)         00052100
          WRITE(NO,7007)                                              00052200
7007     FORMAT(1X,'ITERATION NUMBER',7X,'KEY COLUMN VARIABLE',   00052300
              *10X,'OBJECTIVE FUNCTION'//)                          00052400
          WRITE(NO,7008) IT,NBN1(KCOL),NBN2(KCOL),FN                00052500
7008     FORMAT(I9,22X,A4,A1,17X,F14.6)                             00052600
          GO TO 7009                                                  00052700
7010     WRITE(NO,9202)                                              00052800
9202     FORMAT(1X,'ITERATION NUMBER',7X,'VAR IN',6X,'VAR OUT',11X, 00052900
              *'OBJECTIVE FUNCTION'//)                              00053000
          WRITE(NO,9127) IT,IBN1(KROW),IBN2(KROW),NBN1(KCOL),      00053100
              *NBN2(KCOL),FN                                         00053200
9127     FORMAT(I9,15X,A4,A1,8X,A4,A1,11X,F14.6)                   00053300
7009     WRITE(NO,9500)                                              00053400
          ND=0                                                         00053500
9500     FORMAT(////)                                                00053600
          GO TO 9101                                                  00053700
C*****                                                                00053800
5433     IF(IPHASE-1) 8000,8000,5434                                00053900
8000     IPHASE=2                                                    00054000
          IF(NINF) 8003,8003,8004'                                  00054100
8004     WRITE(NO,8005)                                              00054200
8005     FORMAT(1X,'SOLUTION INFEASIBLE-END OF PHASE 1'////)       00054300
          GO TO 5434                                                  00054400
8003     WRITE(NO,8002)                                              00054500
8002     FORMAT(1X,'SOLUTION FEASIBLE-END OF PHASE 1'////)        00054600
          GO TO 5232                                                  00054700
C*****                                                                00054800
C OUTPUT ROUTINE                                                  00054900
C*****                                                                00055000
5434     WRITE(NO,6701)                                              00055100
6701     FORMAT(//'***** THE CONTINUOUS SOLUTION IS *****'//)  00055200
          WRITE(NO,301) IT,FN                                         00055300
301     FORMAT(1X,'ITERATION ',I5,5X,' OBJ FN ',F14.6////)         00055400
          WRITE(NO,302)                                              00055500
302     FORMAT(3X,'BASIS VAR',5X,'UPPER BOUND? 0=NO/1=YES',15X,'AMOUNT',8X, 00055600
              *,'UNIT PROFIT',//)                                    00055700
          DO 3033 I=1,M                                              00055800
3033     WRITE(NO,304) IBN1(I),IBN2(I),IFLAG(I),RQ(I),BP(I)        00055900
304     FORMAT(7X,A4,A1,17X,I2,19X,F14.6,2X,F14.6//)               00056000
          WRITE(NO,9501)                                              00056100
9501     FORMAT(1X,////)                                             00056200
          WRITE(NO,305)                                              00056300
305     FORMAT(2X,'NB VARIABLE',12X,'REDUCED COST',9X,'NB VARIABLE AT UPPE 00056400
              *R BOUND',9X,'UNIT PROFIT'//)                        00056500
          DO 309 J=1,N                                                00056600
          IF(NBN1(J)-BLNK+NBN2(J)-BLNK) 311,309,311                00056700
311     IF(NFLAG(J)-1) 312,313,312                                  00056800

```

```

313 WRITE(NO,314) NBN1(J),NBN2(J),PI(J),NBP(J) 00056900
GO TO 309 00057000
314 FORMAT(5X,A4,A1,8X,F14.6,25X,'YES',17X,F14.6//) 00057100
312 WRITE(NO,310) NBN1(J),NBN2(J),PI(J) 00057200
310 FORMAT(5X,A4,A1,8X,F14.6,25X,'NO',27X,'O.O'//) 00057300
309 CONTINUE 00057400
C 00057500
C TEST WHETHER THE CONTINUOUS SOLUTION FOUND IS OPTIMAL 00057600
C 00057700
DO 1004 I=1,M 00057800
IF(RB(I)-1.0) 1004,1005,1004 00057900
1005 X=RQ(I) 00058000
IF(ABS(X)-1.0D-7) 1004,1004,1006 00058100
1006 X=RQ(I)-1.0 00058200
IF(ABS(X)-1.0D-7) 1004,1004,1007 00058300
1004 CONTINUE 00058400
WRITE(NO,1008) 00058500
1008 FORMAT(1X,'THE CONTINUOUS SOLUTION FOUND IS OPTIMAL'//) 00058600
STOP 00058700
1007 WRITE(NO,1009) 00058800
1009 FORMAT(1X,'THE CONTINUOUS SOLUTION FOUND IS NOT OPTIMAL-CONTINUE WITH THE EXTREME POINT RANKING METHODOLOGY'//) 00058900
C 00059000
C SET NOPER=0, LSTE=1, KEXT=0, NSTOP=50000, NHALT= 4000, NFOUND=0, 00059100
C NINT=0, NTEST=0 00059200
C 00059300
C 00059400
NOPER=0 00059500
LSTE=1 00059600
KEXT=0 00059700
NSTOP=50000 00059800
NHALT=4000 00059900
NFOUND=0 00060000
NINT=0 00060100
NTEST=0 00060200
C 00060300
C SET THE FIRST RANKED EXTREME POINT SOLUTION EQUAL TO THE CONTINUOUS 00060400
C SOLUTION FOUND 00060500
C 00060600
KEXT=KEXT+1 00060700
C 00060800
C DEFINE OPEN STATEMENTS TO CREATE A FILE AND CONNECT IT TO A UNIT 00060900
C 00061000
OPEN(UNIT=8,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00061100
* RECL=80) 00061200
OPEN(UNIT=9,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00061300
* RECL=80) 00061400
OPEN(UNIT=10,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00061500
* RECL=80) 00061600
OPEN(UNIT=11,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00061700
* RECL=160) 00061800
OPEN(UNIT=12,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00061900
* RECL=160) 00062000
OPEN(UNIT=13,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00062100
* RECL=160) 00062200
OPEN(UNIT=14,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00062300
* RECL=17600) 00062400
OPEN(UNIT=15,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00062500
* RECL=440) 00062600
OPEN(UNIT=16,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00062700
* RECL=440) 00062800
OPEN(UNIT=17,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00062900
* RECL=440) 00063000
OPEN(UNIT=18,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00063100
* RECL=880) 00063200
OPEN(UNIT=19,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00063300
* RECL=880) 00063400
OPEN(UNIT=20,ERR=9999,STATUS='SCRATCH',ACCESS='DIRECT', 00063500
* RECL=880) 00063600
C 00063700
C SET THE TEST SOLUTION EQUAL TO THE FIRST RANKED EXTREME POINT SOLUTION 00063800
C 00063900

```

```

      FTES=FN                                00064000
      L=LSTE                                  00064100
      FSTE(L)=FN                              00064200
      IIND(L)=O                               00064300
DO 1001 I=1,M                               00064400
      ITES1(I)=IBN1(I)                       00064500
      ITES2(I)=IBN2(I)                       00064600
      IFTES(I)=IFLAG(I)                     00064700
      BPTES(I)=BP(I)                        00064800
      RQTES(I)=RQ(I)                        00064900
      RBTES(I)=RB(I)                        00065000
DO 1002 JR=1,N                               00065100
1002 BTES(I,JR)=B(I,JR)                    00065200
1001 CONTINUE                               00065300
DO 1003 JR=1,N                               00065400
      NTES1(JR)=NBN1(JR)                   00065500
      NTES2(JR)=NBN2(JR)                   00065600
      NFTES(JR)=NFLAG(JR)                 00065700
      NBPTES(JR)=NBP(JR)                  00065800
      UBTES(JR)=UB(JR)                    00065900
1003 PITES(JR)=PI(JR)                      00066000
C                                             00066100
C STORE THE FIRST RANKED EXTREME POINT SOLUTION IN THE 00066200
C DIRECT ACCESS FILES                       00066300
C                                             00066400
      WRITE(UNIT=8,REC=L) (IBN1(I),I=1,M)  00066500
      WRITE(UNIT=9,REC=L) (IBN2(I),I=1,M)  00066600
      WRITE(UNIT=10,REC=L) (IFLAG(I),I=1,M) 00066700
      WRITE(UNIT=11,REC=L) (BP(I),I=1,M)    00066800
      WRITE(UNIT=12,REC=L) (RQ(I),I=1,M)    00066900
      WRITE(UNIT=13,REC=L) (RB(I),I=1,M)    00067000
      WRITE(UNIT=14,REC=L) ((B(I,JR),JR=1,N),I=1,M) 00067100
      WRITE(UNIT=15,REC=L) (NBN1(JR),JR=1,N) 00067200
      WRITE(UNIT=16,REC=L) (NBN2(JR),JR=1,N) 00067300
      WRITE(UNIT=17,REC=L) (NFLAG(JR),JR=1,N) 00067400
      WRITE(UNIT=18,REC=L) (NBP(JR),JR=1,N)  00067500
      WRITE(UNIT=19,REC=L) (UB(JR),JR=1,N)  00067600
      WRITE(UNIT=20,REC=L) (PI(JR),JR=1,N)  00067700
C                                             00067800
      LSTE=LSTE+1                            00067900
C                                             00068000
C GENERATE THE EXTREME POINTS ADJACENT TO THE RANKED EXTREME POINT 00068100
C SOLUTION DETERMINED BY INTRODUCING NON BASIC VARIABLES ONE AT A TIME 00068200
C                                             00068300
889 DO 1010 J=1,N                            00068400
C                                             00068500
C DISREGARD ARTIFICIAL VARIABLES           00068600
C                                             00068700
      IF(NTES1(J)-BLNK+NTES2(J)-BLNK) 771,2979,771 00068800
771 KCOL=J                                    00068900
C*****00069000
C DETERMINE KEY ROW                        00069100
C*****00069200
      KROW1=O                                00069300
      KROW2=O                                00069400
      CUBAR=PITES(J)                        00069500
      THETA1=1.OE20                          00069600
      THETA2=1.OE20                          00069700
DO 1011 I=1,M                               00069800
      IF(BTES(I,KCOL)) 1012,1011,1014       00069900
1012 RATIO2=(RBTES(I)-RQTES(I))/(-BTES(I,KCOL)) 00070000
      IF(RATIO2-THETA2) 1015,1011,1011     00070100
1015 THETA2=RATIO2                          00070200
      KROW2=I                               00070300
      GO TO 1011                            00070400
1014 RATIO1=RQTES(I)/BTES(I,KCOL)          00070500
      IF(RATIO1-THETA1) 1016,1011,1011     00070600
1016 THETA1=RATIO1                          00070700
      KROW1=I                               00070800
1011 CONTINUE                               00070900
      RMIN=THETA1                           00071000

```



```

      IF(RMIN-THETA2) .1017,1017,1018      00071100
1018  RMIN=THETA2                          00071200
1017  IF(RMIN-UBTES(KCOL)) 1019,1019,1020  00071300
1020  RMIN=UBTES(KCOL)                     00071400
1019  IF(RMIN-THETA1) 1021,1022,1021      00071500
1021  IF(RMIN-THETA2) 1023,1024,1023      00071600
C*****
C RMIN=UB(KCOL). THEREFORE, X(KCOL) REMAINS NONBASIC BUT AT ITS UPPER  00071800
C BOUND. THUS IT MUST BE SUBSTITUTED AS X(KCOL)=UB(KCOL)-X'(KCOL).    00071900
C IF NFLAG(KCOL)=0 *OR* AS X'(KCOL)=UB(KCOL)-X(KCOL) IF NFLAG(KCOL)=1  00072000
C*****
1023  ND=3                                  00072200
      FTES=FTES-RMIN*PITES(KCOL)           00072300
9742  IF(FTES-FN) 9743,9743,2979           00072400
9743  IF(NINT-O) 4013,4012,4013            00072500
4012  X=FTES-ZBAR                          00072600
      IF(ABS(X)-1.0D-6) 4014,4014,4013     00072700
4014  WRITE(ND,4079)                       00072800
4079  FORMAT(1X,'AN ADJACENT EXTREME POINT HAVING THE SAME FUNCTIONAL VA 00072900
      *LUE AS THE INITIAL L.B. SOLUTION HAS BEEN FOUND'///)           00073000
      NTEST=1                              00073100
      GO TO(1031,1031,1025),ND              00073200
4013  GO TO(4016,4016,4017),ND             00073300
4016  IF(FTES-ZBAR) 4010,4010,1031         00073400
4017  IF(FTES-ZBAR) 4010,4010,1025        00073500
4010  WRITE(ND,4011) FTES                  00073600
4011  FORMAT(1X,'FTES (LESS THAN ZBAR) = ',F14.6//)                    00073700
      GO TO 2979                            00073800
1025  IF(NFTES(KCOL)-1) 1026,1027,1026    00073900
1026  NFTES(KCOL)=1                        00074000
      GO TO 1028                            00074100
1027  NFTES(KCOL)=0                        00074200
1028  DO 1029 I=1,M                        00074300
      RQTES(I)=RQTES(I)-BTES(I,KCOL)       00074400
1029  BTES(I,KCOL)=-BTES(I,KCOL)          00074500
      IT=IT+1                               00074600
      IF(IT-NSTOP) 477,1092,1092           00074700
477  PITES(KCOL)=-PITES(KCOL)             00074800
      GO TO 1041                            00074900
C*****
1022  KROW=KROW1                            00075100
      ID=1                                  00075200
      ND=1                                  00075300
      FTES=FTES-RMIN*PITES(KCOL)           00075400
      GO TO 9742                            00075500
1024  KROW=KROW2                            00075600
      ID=2                                  00075700
      ND=2                                  00075800
      FTES=FTES-RMIN*PITES(KCOL)           00075900
      GO TO 9742                            00076000
C*****
C TRANSFORM                               00076100
C DEVIDE BY PIVOT                          00076200
C DEVIDE BY PIVOT                          00076300
C*****
1031  PIVOT=BTES(KROW,KCOL)                00076400
      DO 1032 JR=1,N                        00076500
1032  BTES(KROW,JR)=BTES(KROW,JR)/PIVOT    00076600
      RQTES(KROW)=RQTES(KROW)/PIVOT        00076700
      DO 1033 I=1,M                         00076800
      IF(I-KROW) 1034,1033,1034            00076900
1034  RQTES(I)=RQTES(I)-RQTES(KROW)*BTES(I,KCOL) 00077000
      DO 1035 JR=1,N                        00077100
      IF(JR-KCOL) 1036,1035,1036          00077200
1036  BTES(I,JR)=BTES(I,JR)-BTES(KROW,JR)*BTES(I,KCOL) 00077300
1035  CONTINUE                             00077400
1033  CONTINUE                             00077500
      DO 1037 I=1,M                         00077600
1037  BTES(I,KCOL)=-BTES(I,KCOL)/PIVOT    00077700
      BTES(KROW,KCOL)=1.0/PIVOT            00077800
C*****
C INTERCHANGE BASIS AND NONBASIS VARIABLES 00077900
      00078000
      00078100

```

```

C*****00078200
      RNM1=NTES1(KCOL) 00078300
      RNM2=NTES2(KCOL) 00078400
      NTES1(KCOL)=ITES1(KROW) 00078500
      NTES2(KCOL)=ITES2(KROW) 00078600
      ITES1(KROW)=RNM1 00078700
      ITES2(KROW)=RNM2 00078800
      LST=NBPTES(KCOL) 00078900
      NBPTES(KCOL)=BPTES(KROW) 00079000
      BPTES(KROW)=LST 00079100
      J1=NFTES(KCOL) 00079200
      NFTES(KCOL)=IFTES(KROW) 00079300
      IFTES(KROW)=J1 00079400
      XL=UBTES(KCOL) 00079500
      UBTES(KCOL)=RBTES(KROW) 00079600
      RBTES(KROW)=XL 00079700
      GO TO(1038,1039),ID 00079800
1038 IT=IT+1 00079900
      IF(IT-NSTOP) 1039,1092,1092 00080000
1039 DO 1040 JR=1,N 00080100
1040 PITES(JR)=PITES(JR)-CJBAR*BTES(KROW,JR) 00080200
      PITES(KCOL)=-CJBAR/PIVOT 00080300
      GO TO(1041,1025),ID 00080400
C*****00080500
C CHECK FOR ESSENTIAL ZERO 00080600
C*****00080700
1041 DO 1042 I=1,M 00080800
      DO 1042 JR=1,N 00080900
          X=BTES(I,JR) 00081000
          IF(ABS(X)-1.0D-7) 1043,1043,1042 00081100
1043 BTES(I,JR)=0. 00081200
1042 CONTINUE 00081300
C 00081400
C COMPARE THE BASES WITH ALREADY STORED ADJACENT EXTREME POINTS 00081500
C 00081600
      IF(LSTE-2) 1054,1055,1055 00081700
1055 L1=LSTE-1 00081800
      DO 1056 L=1,L1 00081900
          X=FSTE(L)-FTES 00082000
          IF(ABS(X)-1.0D-7) 984,984,1056 00082100
984 READ(UNIT=8,REC=L) (ISTE1(I),I=1,M) 00082200
      READ(UNIT=9,REC=L) (ISTE2(I),I=1,M) 00082300
      READ(UNIT=10,REC=L) (IFSTE(I),I=1,M) 00082400
      READ(UNIT=12,REC=L) (RQSTE(I),I=1,M) 00082500
      DO 1057 I=1,M 00082600
          DO 1058 IR=1,M 00082700
              IF(ISTE1(I)-ITES1(IR)) 1058,1059,1058 00082800
              IF(ISTE2(I)-ITES2(IR)) 1058,1060,1058 00082900
              IF(IFSTE(I)-IFTES(IR)) 1061,1062,1063 00083000
              X=RQSTE(I)-1.0+RQTES(IR) 00083100
              IF(ABS(X)-1.0D-7) 1057,1057,1056 00083200
              X=RQSTE(I)-RQTES(IR) 00083300
              GO TO 8008 00083400
              X=RQTES(IR)-1.0+RQSTE(I) 00083500
              GO TO 8008 00083600
1058 CONTINUE 00083700
      DO 7411 JR=1,N 00083800
          IF(NTES1(JR)-BLNK+NTES2(JR)-BLNK) 7412,7411,7412 00083900
7412 IF(ISTE1(I)-NTES1(JR)) 7411,7413,7411 00084000
7413 IF(ISTE2(I)-NTES2(JR)) 7411,7414,7411 00084100
7414 IF(IFSTE(I)-NFTES(JR)) 7415,7416,7415 00084200
7415 X=RQSTE(I)-1.0 00084300
8007 IF(ABS(X)-1.0D-7) 1057,1057,1056 00084400
7416 X=RQSTE(I)-0.0 00084500
      GO TO 8007 00084600
7411 CONTINUE 00084700
1057 CONTINUE 00084800
      WRITE(NO,1065) IT,FTES 00084900
1065 FORMAT(1X,'IT= ',I5,5X,'FTES= ',F14.6,5X,'ADJACENT EXTREME POINT I 00085000
      *S THE SAME AS ONE OF THE STORED POINTS'///) 00085100
      GO TO 2979 00085200

```

```

1056 CONTINUE                                00085300
      WRITE(ND,1066) IT,FTES                 00085400
1066 FORMAT(1X,'IT= ',I5,5X,'FTES= ',F14.6,5X,'ADJACENT EXTREME POINT I 00085500
      *S NOT THE SAME AS ANY OF THE STORED POINTS'//) 00085600
C                                             00085700
C TEST FOR INTEGER FEASIBILITY              00085800
C                                             00085900
1054 DO 1067 I=1,M                          00086000
      IF(RBTES(I)-1.0) 1067,1068,1067      00086100
1068 X=RQTES(I)                             00086200
      IF(ABS(X)-1.0D-7) 1067,1067,1069    00086300
1069 X=RQTES(I)-1.0                         00086400
      IF(ABS(X)-1.0D-7) 1067,1067,1070    00086500
1067 CONTINUE                              00086600
      WRITE(ND,1071)                       00086700
1071 FORMAT(1X,'THE ADJACENT EXTREME POINT GENERATED IS INTEGER FEASIBL 00086800
      *E'//)                                00086900
      IF(NINT-O) 6169,6170,6169           00087000
6170 IF(NTEST-1) 6169,6171,6169           00087100
6171 NINT=1                                 00087200
      WRITE(ND,6172)                       00087300
6172 FORMAT(1X,'THE ADJACENT EXTREME POINT SOLUTION FOUND EQUAL TO THE 00087400
      *INITIAL L.B. SOLUTION IS INTEGER FEASIBLE'//) 00087500
      ZBAR=FTES                            00087600
      GO TO 2979                            00087700
C                                             00087800
C TEST WHETHER AN IMPROVEMENT OVER ZBAR HAS BEEN FOUND 00087900
C                                             00088000
6169 IF(ZBAR-FTES) 6173,1070,1070        00088100
6173 IF(NINT-O) 1072,6174,1072           00088200
6174 NINT=1                                00088300
C                                             00088400
C UPDATE ZBAR                              00088500
C                                             00088600
1072 IF(NFOUND-O) 4966,4473,4966         00088700
4473 NFOUND=1                              00088800
4966 ZBAR=FTES                             00088900
      WRITE(ND,6337) ZBAR                  00089000
6337 FORMAT(1X,'AN IMPROVED INTEGER FEASIBLE SOLUTION IS FOUND WITH A V 00089100
      *ALUE = ',F14.6//)                  00089200
C                                             00089300
C STORE THE IMPROVED INTEGER FEASIBLE SOLUTION. MAKE USE OF DIFFERENT 00089400
C ARRAYS FOR THIS PURPOSE                  00089500
C                                             00089600
      FBAR=ZBAR                            00089700
      DO 1073 I=1,M                        00089800
        IBAR1(I)=ITES1(I)                  00089900
        IBAR2(I)=ITES2(I)                  00090000
        IFBAR(I)=IFTES(I)                  00090100
        BPBAR(I)=BPTES(I)                  00090200
        RQBAR(I)=RQTES(I)                  00090300
        RBBAR(I)=RBTES(I)                  00090400
      DO 1074 JR=1,N                        00090500
1074 BBAR(I,JR)=BTES(I,JR)                 00090600
1073 CONTINUE                              00090700
      DO 1075 JR=1,N                        00090800
        NBAR1(JR)=NTES1(JR)                00090900
        NBAR2(JR)=NTES2(JR)                00091000
        NFBAR(JR)=NFTES(JR)                00091100
        NBPBAR(JR)=NBPTES(JR)              00091200
        UBBAR(JR)=UBTES(JR)                00091300
1075 PIBAR(JR)=PITES(JR)                   00091400
C                                             00091500
C SET NOPER=1                              00091600
C                                             00091700
      NOPER=1                              00091800
C                                             00091900
C STORE THE ADJACENT EXTREME POINT GENERATED IN THE 00092000
C DIRECT ACCESS FILES                      00092100
C                                             00092200
1070 L=LSTE                                00092300

```

```

FSTE(L)=FTES                                00092400
IIND(L)=1                                    00092500
WRITE(UNIT=8,REC=L) (ITES1(I),I=1,M)        00092600
WRITE(UNIT=9,REC=L) (ITES2(I),I=1,M)        00092700
WRITE(UNIT=10,REC=L) (IFTES(I),I=1,M)       00092800
WRITE(UNIT=11,REC=L) (BPTES(I),I=1,M)       00092900
WRITE(UNIT=12,REC=L) (RQTES(I),I=1,M)       00093000
WRITE(UNIT=13,REC=L) (RBTES(I),I=1,M)       00093100
WRITE(UNIT=14,REC=L) ((BTES(I,JR),JR=1,N),I=1,M) 00093200
WRITE(UNIT=15,REC=L) (NTES1(JR),JR=1,N)     00093300
WRITE(UNIT=16,REC=L) (NTES2(JR),JR=1,N)     00093400
WRITE(UNIT=17,REC=L) (NFTES(JR),JR=1,N)     00093500
WRITE(UNIT=18,REC=L) (NBPTES(JR),JR=1,N)    00093600
WRITE(UNIT=19,REC=L) (UBTES(JR),JR=1,N)     00093700
WRITE(UNIT=20,REC=L) (PITES(JR),JR=1,N)     00093800
LSTE=LSTE+1                                  00093900
WRITE(NO,6480) L                             00094000
6480 FORMAT(1X,'THE ADJACENT EXTREME POINT STORED IS = ',I5//) 00094100
      IF(LSTE-5000) 2979,2979,6476          00094200
6476 IF(NFOUND-O) 6477,6478,6477           00094300
6478 GO TO 3438                             00094400
6477 WRITE(NO,6479)                          00094500
6479 FORMAT(1X,'BEST INTEGER SOLUTION FOUND WITHIN THE FIRST FIVE THOUSAND 00094600
      *AND STORED ADJACENT EXTREME POINTS IS'//) 00094700
      GO TO 1085                             00094800
C                                             00094900
C SET TEST SOLUTION EQUAL TO THE RANKED EXTREME POINT 00095000
C SOLUTION UNDER CONSIDERATION              00095100
C                                             00095200
2979 FTES=FN                                  00095300
      DO 2074 I=1,M                          00095400
        ITES1(I)=IBN1(I)                    00095500
        ITES2(I)=IBN2(I)                    00095600
        IFTES(I)=IFLAG(I)                  00095700
        BPTES(I)=BP(I)                     00095800
        RQTES(I)=RQ(I)                     00095900
        RBTES(I)=RB(I)                     00096000
      DO 2075 JR=1,N                          00096100
        BTES(I,JR)=B(I,JR)                 00096200
2075 CONTINUE                                00096300
      DO 2076 JR=1,N                          00096400
        NTES1(JR)=NBN1(JR)                 00096500
        NTES2(JR)=NBN2(JR)                 00096600
        NFTES(JR)=NFLAG(JR)                00096700
        NBPTES(JR)=NBP(JR)                 00096800
        UBTES(JR)=UB(JR)                   00096900
2076 PITES(JR)=PI(JR)                       00097000
1010 CONTINUE                                00097100
C                                             00097200
C IF AN IMPROVED INTEGER FEASIBLE SOLUTION HAS BEEN FOUND I.E. 00097300
C NOPER=1, TEST WHETHER ANY OF THE ADJACENT EXTREME POINTS FOUND 00097400
C CAN BE DISREGARDED                        00097500
C                                             00097600
      IF(NOPER-1) 1080,1081,1080           00097700
1081 L=LSTE-1                                00097800
      DO 1077 I=1,L                          00097900
        IF(IIND(I)-O) 1078,1077,1078       00098000
1078 IF(FSTE(I)-ZBAR) 1079,1079,1077       00098100
1079 IIND(I)=O                               00098200
1077 CONTINUE                                00098300
      NOPER=O                                00098400
C                                             00098500
C TEST WHETHER THERE ARE ANY POTENTIAL ADJACENT EXTREME POINTS 00098600
C FOR CONSIDERATION IN THE LIST; IF NONE STOP! 00098700
C                                             00098800
1080 L=LSTE-1                                00098900
      DO 1082 I=1,L                          00099000
        IF(IIND(I)-O) 1083,1082,1083       00099100
1082 CONTINUE                                00099200
C                                             00099300
C NONE OF THE ADJACENT EXTREME POINTS IN THE STORED LIST 00099400

```



```

3439 FORMAT(1X,'NO BETTER INTEGER SOLUTION HAS BEEN FOUND SO FAR'//) 00106600
      GO TO 197 00106700
3437 WRITE(NO,3024) 00106800
3024 FORMAT(1X,'BEST INTEGER SOLUTION FOUND WITHIN THE FIRST FOUR THOUS00106900
      *AND RANKED EXTREME POINTS IS'//) 00107000
      GO TO 1085 00107100
1092 IF(NFOUND-O) 3440,3438,3440 00107200
3440 WRITE(NO,1093) 00107300
1093 FORMAT(1X,'BEST INTEGER SOLUTION FOUND WITHIN THE FIRST FIFTY THOU00107400
      *SAND ITERATIONS IS'//) 00107500
      GO TO 1085 00107600
C*****00107700
C FINAL OUTPUT ROUTINE 00107800
C*****00107900
3441 IF(NFOUND-O) 1085,3438,1085 00108000
1085 WRITE(NO,1094) 00108100
1094 FORMAT(1X,'***** THE FINAL RESULTS ARE *****'//) 00108200
      WRITE(NO,1095) IT,FBAR 00108300
1095 FORMAT(1X,' ITERATION ',I5.5X,' OBJ FN ',F14.6'//) 00108400
      WRITE(NO,1096) 00108500
1096 FORMAT(3X,'BASIS VAR',5X,'UPPER BOUND? O=NO/1=YES',15X,'AMOUNT',8X00108600
      *, 'UNIT PROFIT',//) 00108700
      DO 1097 IK=1,M 00108800
1097 WRITE(NO,1098) IBAR1(IK),IBAR2(IK),IFBAR(IK),RQBAR(IK),BPBAR(IK) 00108900
1098 FORMAT(7X,A4,A1,17X,I2,19X,F14.6,2X,F14.6'//) 00109000
      WRITE(NO,1099) 00109100
1099 FORMAT(1X,'//') 00109200
      WRITE(NO,1100) 00109300
1100 FORMAT(2X,'NB VARIABLE',12X,'REDUCED COST',9X,'NB VARIABLE AT UPPE00109400
      *R BOUND',9X,'UNIT PROFIT'//) 00109500
      DO 1101 JF=1,N 00109600
          IF(NBAR1(JF)-BLNK+NBAR2(JF)-BLNK) 1102,1101,1102 00109700
1102 IF(NFBAR(JF)-1) 1103,1104,1103 00109800
1104 WRITE(NO,1105) NBAR1(JF),NBAR2(JF),PIBAR(JF),NBPBAR(JF) 00109900
1105 FORMAT(5X,A4,A1,8X,F14.6,25X,'YES',17X,F14.6'//) 00110000
      GO TO 1101 00110100
1103 WRITE(NO,1106) NBAR1(JF),NBAR2(JF),PIBAR(JF) 00110200
1106 FORMAT(5X,A4,A1,8X,F14.6,25X,'NO',27X,'O.O'//) 00110300
1101 CONTINUE 00110400
      GO TO 197 00110500
9999 WRITE(NO,7053) 00110600
7053 FORMAT(1X,'AN ERROR IN OPEN STATEMENT OCCURED') 00110700
197 STOP 00110800
      END 00110900

```

VITA

Rasaratnam Logendran

Candidate for the Degree of

Doctor of Philosophy

Thesis: PLANT LOCATION-ALLOCATION PROBLEMS WITH PRICE SENSITIVE
STOCHASTIC DEMANDS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Kopay, Sri Lanka, August 2, 1953, the son
of Vythilingam and Nageswari Rasaratnam.

Education: Graduated from St. John's College, Jaffna, Sri Lanka in
October, 1970; received the Bachelor of Science (Hons.) degree
in Mechanical Engineering from the University of Sri Lanka,
Katubedda, Sri Lanka, in July, 1975; received the Master of
Engineering degree in Industrial Engineering and Management
from the Asian Institute of Technology, Bangkok, Thailand, in
April, 1980; completed requirements for the Doctor of Philosophy
degree at Oklahoma State University in December, 1984.

Professional Experience: Mechanical Engineer, Equipment Branch, Sri
Lanka Sugar Corporation, Hingurana, Sri Lanka, February, 1976
to December, 1976; Research Officer, Ceylon Institute of
Scientific and Industrial Research, Colombo, Sri Lanka,
January, 1977 to August, 1978; Research Associate, Division of
Industrial Engineering and Management, Asian Institute of
Technology, Bangkok, Thailand, August, 1980 to August, 1981;
Teaching Associate, Department of Management, College of
Business, Oklahoma State University, September, 1982 to
December, 1982; Teaching and Research Assistant, School of
Industrial Engineering and Management, Oklahoma State
University, September, 1981 to June, 1984.

Professional Organizations: Institute of Industrial Engineers,
Alpha Pi Mu.