A CENTRAL FACILITIES LOCATION PROBLEM INVOLVING TRAVELING SALESMAN TOURS AND EXPECTED DISTANCES

by

Robert Currie Burness

Thesis submitted to the Graduate Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Industrial Engineering and Operations Research

Approved:

John A. White, Chairman

J. William Schmidt

Wayne C. Turner

May 1974

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to all those who have contributed to the completion of this thesis.

Dr. John A. White, as major advisor, provided timely suggestions throughout the research. His helpful advice and criticism are greatly appreciated. Gratitude is also expressed to the other members of the graduate committee, Dr. J. William Schmidt and Dr. Wayne C. Turner.

The author extends his appreciation to the IEOR Department whose financial assistance made this research possible.

Thanks are also extended to Ms. Margie Strickler for her excellent typing of the manuscript under severe time limitations.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Subject of the Research

The need for cost reduction in the areas of distribution management, transportation, and urban planning has been continuously increasing over the past several decades due among other things to the growing shortages of natural resources. Facilities location techniques have been employed in these areas to help determine more efficient allocations of scare resources so as to reduce costs. While most applications have been made in industrial situations, a growing awareness in governmental areas of the advantages of location analysis is evidenced by an increased demand for personnel familiar with facilities location techniques.

The problem to be studied in this thesis involves the determination of a minimum cost location of a central facility when trips taken from the central facility to some or all of m existing facilities form traveling salesman tours. The problem is probabilistic in nature in that many possible combinations of trips involving different existing facilities may occur. While the problem formulated is original, much of the discussion draws upon similar,well studied problems in the area of facilities location.

The reader unfamiliar with the subject of facilities location and design is referred to the text of Francis and White [10]. In addition to presenting the basic different types of location problems they provide an in depth discussion of the appropriate, overall approach that one should take to both facilities location and plant layout problems. A comprehensive report on the more significant contributions to facilities location is given by Bozok [4]. Francis and Goldstein [9] and Lea [12] recently compiled extensive bibliographies of facilities location literature that should enable the researcher to keep abreast of the works of other researchers with similar interests, but who are in different fields.

Using the taxonomy of facilities location problems presented by Francis and White [10] the location problem considered in this thesis involves the location of a single new facility relative to a known number of existing facilities having known, fixed locations. A continuous solution space is assumed, rectilinear and Euclidean distances are employed, and an objective of minimizing expected total cost is specified for the problem under study. In fact, the location problem treated possesses the characteristics of the classic Steiner-Weber problem, which has been studied extensively over the past twenty years.

The problem addressed herein can be referred to as the traveling salesman location problem. It differs from the Steiner-Weber problem on the basis of the travel between the new facility and the existing facilities. Specifically, the Steiner-Weber problem assumes that trips are always made between the new facility and an existing facility each time a trip occurs. The traveling salesman location problem allows more than one existing facility to be visited during a single trip from the new facility. Additionally, the set of existing facilities to be visited during a trip is considered to be a random variable.

Importance of the Topic

Based on an extensive review of the facilities location literature it appears that the present effort is the first to formulate and study the traveling salesman location problem. Even though the problem has not been studied previously, there exist a number of instances for which the traveling salesman location problem formulation is appropriate. An obvious application is the problem of determining the location of a district sales office when the salesman services a number of customers; the location of a blood bank which supplies blood to a number of hospitals is another instance where the traveling salesman location problem occurs.

Method of Approach

The traveling salesman location problem involves the determination of both the location of the new facility and the routing to be taken when visiting the existing facilities. Given the location of the new facility and the set of existing facilities to be visited on a particular trip, the routing to be employed in minimizing the distance traveled per trip is determined by solving the associated traveling salesman problem. Since the set of existing facilities to be visited on a given trip is a random variable, the location of the new facility is to be determined such that the expected distance traveled per trip is minimized.

In considering methods for solving the problem, heuristic solution procedures will be used. Specifically, a step-wise procedure is employed in which search procedures are used to specify a location for the new facility and, given the location, a traveling salesman algorithm is employed to determine the routes which yield a minimum expected cost. Two existing search procedures, the Hyperbolic Approximation Procedure [8] and the Successive Quadratic Approximation Search Procedure [18], are employed in determining the location for the new facility. The associated traveling salesman problems are solved using the algorithm developed by Little, Murty, Sweeney, and Karel [15].

Objectives and Purpose of the Research

The objectives of the research include the presentation of an original formulation of a significant facilities location problem which has not been studied previously, the development of several procedures for determining minimum cost locations, and the comparison of the effectiveness of each solution procedure. An additional objective is to gain some insight regarding the behavior of the function under a variety of different conditions.

The purpose of this research effort is to expand the existing body of facilities location literature to include an investigation of a problem that occurs quite frequently in practice.

Order of the Discussion

To facilitate the presentation of the results of the research effort, Chapter 2 presents the problem formulation and discusses the special features of the problem, e.g., those that prohibit minimization of the function through classical optimization techniques. Several examples are given to depict the psychotic nature of the function under investigation. Chapter 3 presents a solution procedure which incorporates the Hyperbolic

Approximation Procedure. An alternative solution procedure particularly well suited for problems in which there are many existing facilities is also advanced in Chapter 3. The technique employed here is a search by regression through successive quadratic approximation. Chapter 4 presents computational results in which the merits of each procedure are compared, based on computer run times and the effectiveness of the procedures in minimizing the function. Chapter 5 summarizes the research; conclusions resulting from the study are given as well as some recommendations for further research. Attention is devoted to the most obvious extension: the multiple new facility traveling salesman location problem.

Chapter 2

PROBLEM FORMULATION

Introduction

This chapter consists of a mathematical formulation of the probabilistic location problem and a presentation of some relatively simple examples. Additionally, due to the importance of the traveling salesman problem in the research, some of the better known solution procedures to the traveling salesman problem are considered. Further insight as to the nature of the function under investigation will be gained in the study of the example problems.

Notation and Mathematical Formulation

Where convenient to do so, the notation to be used in this thesis will be consistent with the notation found in most of the recent facilities location literature. To begin, the following notation is adopted:

(x,y) = the coordinate location of the new facility.

m = the number of existing facilities.

S_h = a unique subset of facilities, including the new facility, that are to be visited on trip h, h = 1, 2, 3, ..., k. The order of visitation is not implied by the ordering of elements in S_h.

p_h = the probability of visiting the existing facilities contained in subset S_h, h = 1, 2, 3, ..., k.

$d_{ij}(x,y)$ = the distance from facility i to facility j, $1 \leq i \leq m+1$, $1 \leq j \leq m+1$

(a_i, b_i) = the coordinate location of the ith existing facility, $i = 1, 2, 3, \ldots, m$

k = $2^m - 1$, the total number of possible distinct subsets

q_h = the number of elements contained in S_h.

Note that S_h can contain at most (m+1) elements since the subset that includes all existing facilities can be written as $S_h = \{1, 2, 3, \ldots, m, (m+1)\}$, where (m+1) denotes the new facility. Also, only one subset can contain (m+1) elements since $\binom{m}{m} = 1$. Note further that facility (m+1) must be a member of all subsets since all trips begin and end with the new facility. S_h must contain at least two elements, an existing facility and the new facility. It is easy to see that there are $\binom{m}{1} = m$ unique subsets containing two elements. They are $\{1,(m+1)\}$, $\{2,(m+1)\}$, $\{3,(m+1)\}$, ..., $\{m,(m+1)\}$. In general, for subsets containing j+1 elements, $j \leq m$, there are $\binom{m}{j}$ unique subsets that can be formed. The total number of subsets that must be considered, k, equals $2^m - 1$.

The probabilistic location problem can now be expressed as

$$\text{(P0)} \quad \min_{x,y} f(x,y) = \sum_{h=1}^{k=2^m-1} p_h \Big[\min_{Z_{ij}} \sum_{i \in S_h} \sum_{j \in S_h} d_{ij}(x,y) Z_{ij}\Big] \tag{2.1}$$

s.t.

$$\sum_{i \in S_h} Z_{ij} = 1 \quad , \forall\, j \in S_h,\ h = 1, 2, \ldots, k \tag{2.1a}$$

$$\sum_{j \in S_h} Z_{ij} = 1 \quad , \forall\, i \in S_h,\ h = 1, 2, \ldots, k \tag{2.1b}$$

$$u_i - u_j + q_h Z_{ij} \leq q_h - 1 \quad , \forall\, i,j \in S_h,\ i \neq j,\ i \neq 1,\ j \neq 1,\ h = 1, 2, \ldots, k \tag{2.1c}$$

$$Z_{ij} = 0,1 \qquad , \forall\, i,j \varepsilon S_h \quad h = 1, 2, \ldots, k$$

where u_i and u_j are arbitrary real numbers that satisfy (2.1c). For each h the term in brackets along with the four constraints represents the particular traveling salesman problem corresponding to S_h. The interpretation of the first constraint, (2.1a), is that each facility contained in S_h must be entered only once. The second constraint indicates that each facility contained in S_h must be left only once. The first two constraints together impose the restriction that each facility contained in S_h can be visited once and only once. The third constraint forces the solution to be a Hamiltonian circuit, i.e., there can be no subtours in the solution. The last constraint defines the problem to be a zero-one, integer programming problem. If the order of visitation is from facility i to facility j, then $Z_{ij} = 1$. If the minimum distance solution does not dictate a routing from i to j, then $Z_{ij} = 0$.

Problem (PO) is embedded with the minimum distance solutions to k different traveling salesman problems, one for each trip, h:

$$t_h^*(x,y) = \min_{Z_{ij}} \sum_{i \varepsilon S_h} \sum_{j \varepsilon S_h} d_{ij}(x,y) Z_{ij} \tag{2.2}$$

$$\text{s.t.} \quad \sum_{i \varepsilon S_h} Z_{ij} = 1 \qquad , \forall\, j \varepsilon S_h$$

$$\sum_{j \varepsilon S_h} Z_{ij} = 1 \qquad , \forall\, i \varepsilon S_h$$

$$u_i - u_j + q_h Z_{ij} \leq q_h - 1 \quad , \forall\ i,j \varepsilon S_h, \ i \neq j,\ i \neq 1,\ j \neq 1$$

$$Z_{ij} = 0,1 \quad , \forall\ i,j \varepsilon S_h$$

where u_i and u_j are arbitrary real numbers that satisfy the third constraint. (2.2) is a slight variation of the integer programming formulation of the traveling salesman problem presented by Miller, Tucker, and Zemlin [17]. Observe that the solution (2.2) is simply the solution to the traveling salesman problem corresponding to the facilities contained in subset S_h with the new facility located at (x,y). Notice that the decision variables in (2.2) are the Z_{ij}'s rather than x and y. (P0) can now be rewritten as

$$\text{(P1)} \min_{(x,y)} f(x,y) = \sum_{h=1}^{k} p_h[t_h^*(x,y)] \qquad (2.3)$$

Before discussing further the objective function, f(x,y), attention must be given to the distance measure that is to be used. The distance measure may be Euclidean or rectilinear. If the distance measure is Euclidean, then

$$d_{ij}(x,y) = \begin{cases} [(x-a_j)^2+(y-b_j)^2]^{1/2} & , i = m+1, \ j = 1,2,3,\ldots,m \\ [(a_i-a_j)^2+(b_i-b_j)^2]^{1/2} & , i = 1,2,3,\ldots,m \ j = 1,2,3,\ldots,m \end{cases} \qquad (2.4)$$

If the distance measure is rectilinear, then

$$d_{ij}(x,y) = \begin{cases} |x-a_j|+|y-b_j| & , \; i = m+1, \\ & \;\; j = 1,2,3,\ldots,m \\ |a_i-a_j|+|b_i-b_j| & , \; i = 1,2,3,\ldots,m, \\ & \;\; j = 1,2,3,\ldots,m \end{cases} \quad (2.5)$$

In either case it is assumed that $d_{ij}(x,y) = d_{ji}(x,y)$. Notice that the distance from facility i to facility j depends on the location of the new facility, (x,y), only when either facility i or facility j is the new facility. Letting D be the (m+1)x(m+1) distance matrix, a change in the location of the new facility will only change the (m+1)th row and column of the distance matrix.

In Equation (2.3) the objective function, f(x,y), is a convex combination of solutions to $k = 2^m - 1$ traveling salesman problems. The objective is to find a value of x and y for which the convex combination is at a minimum. There is no apparent relationship between p_h, the probability of visiting the facilities contained in S_h, and the term in brackets, namely, the minimum distance solution to the traveling salesman problem for S_h. The relative frequency of occurence of subset S_h is given initially and does not change if the coordinate location of the new facility is changed. To minimize the objective function it would seem that primary attention must be focused on the solution of the $2^m - 1$ traveling salesman problems in such a manner that their weighted sum is minimized.

Before examining some of the existing solution procedures to the traveling salesman problem, several other features of (P1) must be presented. First, note that one functional evaluation of f(x,y) requires the solution of $2^m - 1$ traveling salesman problems so that as m becomes large the solution time required for even one evaluation of the function may become quite long. Hence, for problems involving many existing facilities, procedures that minimize f(x,y), but require many functional evaluations, may not be feasible due to the enormous computation time involved. For only ten existing facilities, for example, one functional evaluation requires the solution of $2^{10} - 1 = 1{,}023$ distinct traveling salesman problems. However, some of the traveling salesman problems are solved trivially. When only one or two existing facilities are to be visited, there is only one route that can be selected. Thus, $\binom{m}{1} + \binom{m}{2}$ solutions are trivial. This reduces the number of nontrivial traveling salesman problems to $2^m - 1 - \binom{m}{1} - \binom{m}{2} = 2^m - \frac{1}{2}(m^2+m+2)$. The second procedure for minimizing f(x,y) that is developed in Chapter 3 has a strong point in that it is designed to minimize the number of functional evaluations required and consequently is well suited for problems involving many existing facilities. There is a trade-off in this case, though, since the procedure will only generate near optimal solutions.

A second difficulty presented by (P1) is that there is no reason to believe apriori that f(x,y) is a convex function. The successful use of quasi-enumerative techniques is heavily dependent on the convexity of a function. As will be seen in the examples presented at the end of this chapter, f(x,y) is not always convex. The first technique developed in Chapter 3, however, takes advantage of a special structure of the problem such that it is capable of recognizing a local minimum.

The Traveling Salesman Problem

The well studied traveling salesman problem, while simply stated, has yet to have been solved by any computationally efficient solution technique. Given n cities and known distances between all cities, the minimum distance route must be determined such that the salesman starts at the first city, visits all other cities only once, and finally returns to the first city. In other words, there can be no subtours. One complete tour is known in graph theory as a Hamiltonian circuit. Starting from the first city there are (n-1)! possible routes that may be taken. Note that n = m+1 so that there are m! possible round trips for the subset in which all existing facilities are visited.

The integer programming formulation of the traveling salesman problem was used in the mathematical formulation of the traveling salesman location problem for the sole purpose of making the formulation precise. The solution procedures to be used will not use integer programming techniques. Unfortunately, the computational results reported on the use of integer programming to solve large traveling salesman problems have been unfavorable.

There have been numerous optimal seeking procedures developed for the solution of the traveling salesman problem, the most famous of which is the branch and bound algorithm developed by Little, Murty, Sweeney, and Karel [15]. The solution procedures for the probabilistic location problem will utilize this algorithm.

Another algorithm developed by Eastman [5, 6] is a branch and bound procedure using subtour eliminations. This method is not known to have been programmed, however. Bellman [2] has developed a dynamic programming formulation of the traveling salesman problem. The shortcoming of this approach is that for large problems storage requirements are excessive. A more comprehensive discussion of branch and bound techniques and strategies is given by Lawler and Wood [11]. A more in depth treatment of optimal seeking procedures can be found in Bellmore and Malone [3] and Turner, Ghare, and Foulds [19].

Though they will not be used in this thesis, some heuristic methods that generate near optimal solutions to the traveling salesman problem will be mentioned. Such methods may have to be used for large scale probabilistic location problems due to the combinatorics involved with the location problem itself. With regard to traveling salesman problems it has been found that the problem is compounded by the fact that for optimal seeking methods the computation time required for solution of traveling salesman problems increases exponentially with n. One heuristic method is that of Ashour and Parker [1] where a look-ahead procedure is combined with the procedure of selecting the nearest unvisited city. The "2-opt" procedure of Lin [13] and the "k-opt" procedure of Lin and Kernighan [14], based on their reported computational results for 200 city problems, seems to be another effective heuristic. Other heuristic procedures are discussed by Eilon, Watson-Gandy, and Christofides [7].

The solution technique to be used in this thesis, Little's method, is a branch and bound method based on penalty tour building. A complete discussion of the algorithm, including an example, is given by Wagner [20].

Little, et al., report mean computation times on an IBM 7090 of 58.5 seconds for 100 thirty-city problems and 8.37 minutes for 5 forty-city problems. Though they argue in favor of branching from the newest bounding problem, their computational results are based on the policy of branching from the node having the least lower bound. The solution procedure to be used in this thesis will adopt the recommended branching strategy. Another refinement especially effective on larger problems is suggested by Eilon, Watson-Gandy, and Christofides [7]. They found a reduction in computation times when the penalties were determined by weighting the second and third smallest row and column elements. Their suggested refinement was not used in this research.

Some Simple Examples

In order to illustrate the complexity of the problem that has been formulated, four relatively simple examples will be presented. These examples will also be used in later chapters to illustrate the various solution procedures.

Consider a situation where four existing facilities are located at $(a_1,b_1) = (0,0)$, $(a_2,b_2) = (100,0)$, $(a_3,b_3) = (100,100)$, and $(a_4,b_4) = (0,100)$. The new facility, facility number 5, must be located so as to minimize the expected distance traveled. The distance measure can be Euclidean or rectilinear. Two of the examples use Euclidean distance measures and two use rectilinear distance. The relative frequency of occurrence of subsets of facilities to be visited will be either the same over all subsets or the same among subsets involving the same number of existing facilities.

Example 2.1 Let the distance measure be Euclidean and define the subset probabilities as follows:

h	P_h	S_h
1	1/15	1,5
2	1/15	2,5
3	1/15	3,5
4	1/15	4,5
5	1/15	1,2,5
6	1/15	1,3,5
7	1/15	1,4,5
8	1/15	2,3,5
9	1/15	2,4,5
10	1/15	3,4,5
11	1/15	1,2,3,5
12	1/15	1,2,4,5
13	1/15	1,3,4,5
14	1/15	2,3,4,5
15	1/15	1,2,3,4,5

Since the existing facilities are located at the vertices of a square and since the subset probabilities give equal weight to all the existing facilities, intuition might lead one to believe that the point (50,50) is the minimum distance location for the new facility. Figure 2.1 provides a cutaway view of the three dimensional graph of the function, $z = f(x,y)$, for $0 \leq x \leq 50$, $50 \leq y \leq 100$. Close examination of the figure reveals that the function is at a minimum somewhere between the points (42,50) and (46,50), with another minimum occurring between the points (50,42) and (50,46). Actually, at the point (50,50) $f(x,y)$ is at a relative maximum. Figure 2.2 is a plot of $f(x,y)$ over the entire square. The maximum and minimum values of $f(x,y)$ are given in Table 2.1. The last four entries in Table 2.1 are not extreme points, but $f(x,y)$ appears to be strictly convex in a neighborhood of each of the four points so that it is safe to conclude that a relative minimum will occur at $x^* = 50$, and within either the interval $42 < y < 46$ or the interval $54 < y < 58$, and by symmetry at $y^* = 50$, and within either the interval

Table 2.1

Maximum and Minimum Values of Objective Function of Example 2.1

(x,y)	f(x,y)	Nature of Extreme Point
(0,0)	280.95	Global maximum
(0,100)	280.95	Global maximum
(100,100)	280.95	Global maximum
(100,0)	280.95	Global maximum
(50,50)	260.28	Local maximum
(50,44)	259.97	Minimum value obtained in grid search
(50,56)	259.97	Minimum value obtained in grid search
(44,50)	259.97	Minimum value obtained in grid search
(56,50)	259.97	Minimum value obtained in grid search

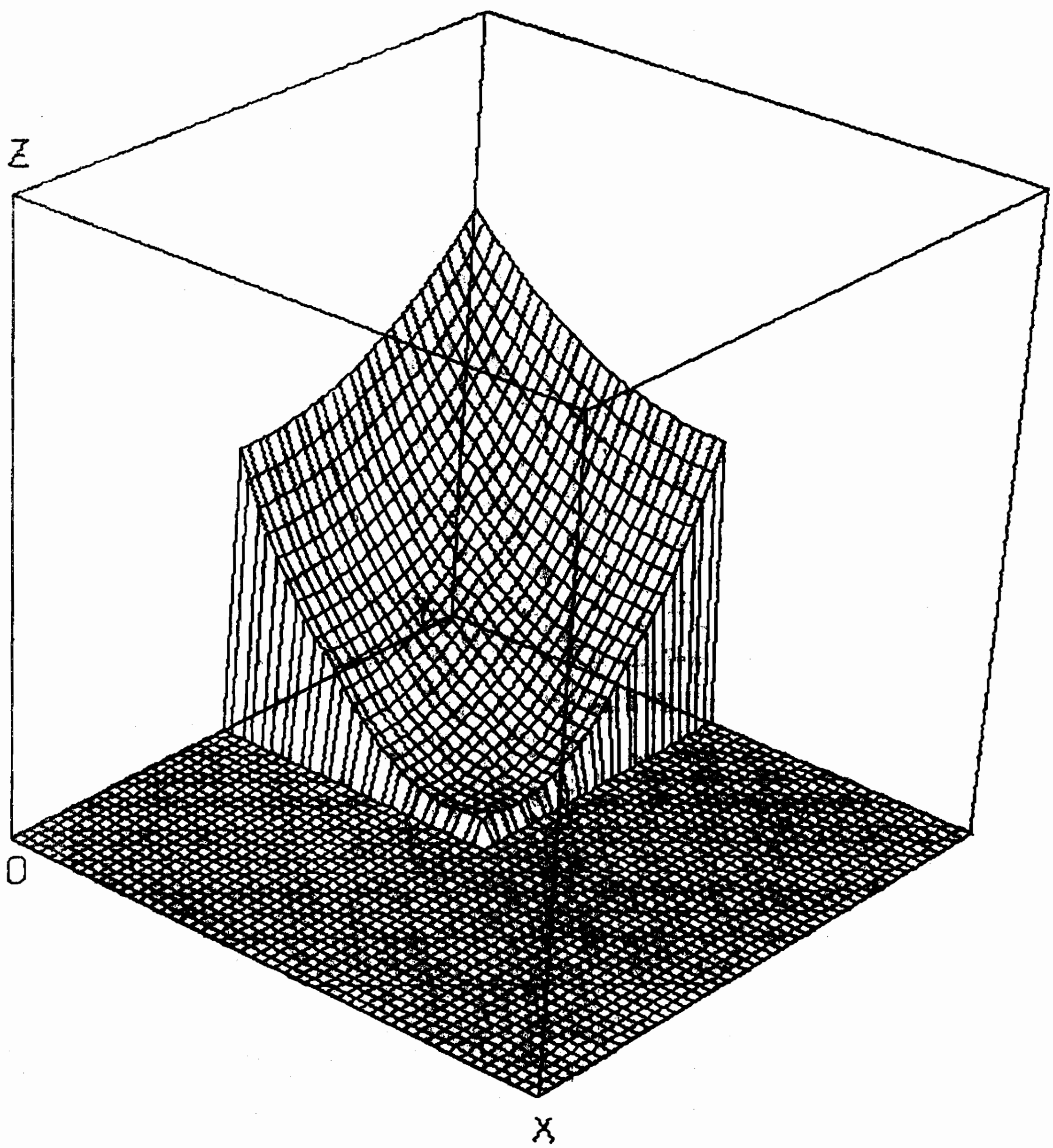

Figure 2.1

A Cutaway View of f(x,y) in Example 2.1

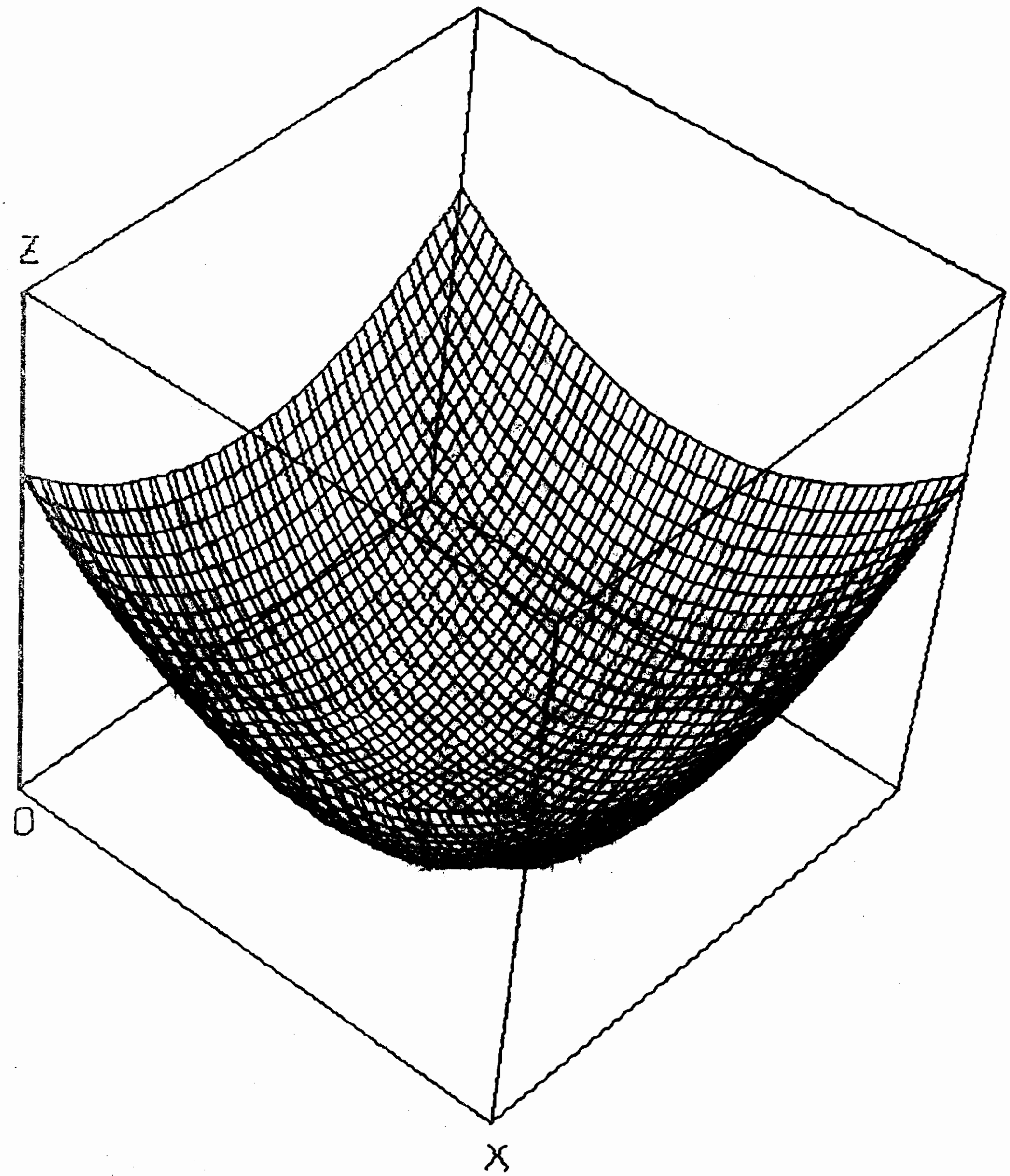

Figure 2.2

An Overall View of f(x,y) in Example 2.1

$42 < x < 46$ or the interval $54 < x < 58$. Notice in Figure 2.2 the ridges that run from the existing facilities to the point (50,50). The existence of these ridges by themselves are enough to make $f(x,y)$ non-convex over the entire region. Now consider a triangular region in the xy plane formed by any two of the four existing facilities and the point (50,50). It can be seen that $f(x,y)$ is nonconvex in this region also. Using the triangular region formed by the points (0,0), (0,100), and (50,50) and then examining $f(x,y = 50)$, $0 \leq x \leq 50$, in Figure 2.1, an inflection point can be seen at $f(18,50)$. $f(x,y)$ appears to be convex beyond the point (18,50), however.

Example 2.2 As a second example we will maintain the Euclidean distance measure and redefine the subset probabilities as follows:

h	P_h	S_h
1	.05	1,5
2	.05	2,5
3	.05	3,5
4	.05	4,5
5	.05	1,2,5
6	.05	1,3,5
7	.05	1,4,5
8	.05	2,3,5
9	.05	2,4,5
10	.05	3,4,5
11	.1	1,2,3,5
12	.1	1,2,4,5
13	.1	1,3,4,5
14	.1	2,3,4,5
15	.1	1,2,3,4,5

This example can be considered to be another special case since the subset probabilities are equal among subsets involving the same number of facilities. As a result, no particular existing facility is given an extra weight due to the assignment of subset probabilities. As in the previous example, a good intuitive guess at the minimum distance location would be the point (50,50). Figures 2.3 and 2.4 provide a cutaway view of the function and a view over the entire region, respectively, analagous to the figures used in Example 2.1. Notice that the four minimum points have shifted slightly outward in relation to the minimum points of Example 2.1, and that the relative maximum occurring at the point (50,50) is now more pronounced. The explanation for these shifts can be made by examining p_k, the probability of visiting all four existing facilities, in each example. Notice first of all, however, that if only all four existing facilities are to be visited, i.e., $p_{15} = 1$, the minimum distance solution is immediately determined by locating the new facility anywhere on the boundary of the square region formed by the existing facilities. Now p_k in Example 2.2 is slightly greater than p_k in Example 2.1 so that the distance resulting from visiting all four existing facilities is given a higher weight in the objective function. Consequently, it is expected that the minimum distance location will shift toward the minimum distance solution that results when $p_k = 1$. The shifts occur along the lines x = 50 and y = 50 because no <u>single</u> existing facility is weighted higher than another. Only the <u>subsets</u> containing a different number of existing facilities are weighted unequally. The shifting of the minimum points of f(x,y) is entirely due to the changes made in the weighting of groups of existing facilities. As an example in which unequal weighting of existing facilities occurs suppose that for S_{11}, S_{12}, S_{13}, and S_{14}, the probabilities of visiting these subsets involving three existing facilities are changed to $p_{11} = p_{12} = p_{13} = .08$ and $p_{14} = .16$. S_{14} involves all existing facilities except facility number one. Hence, facility one is now weighted lower than

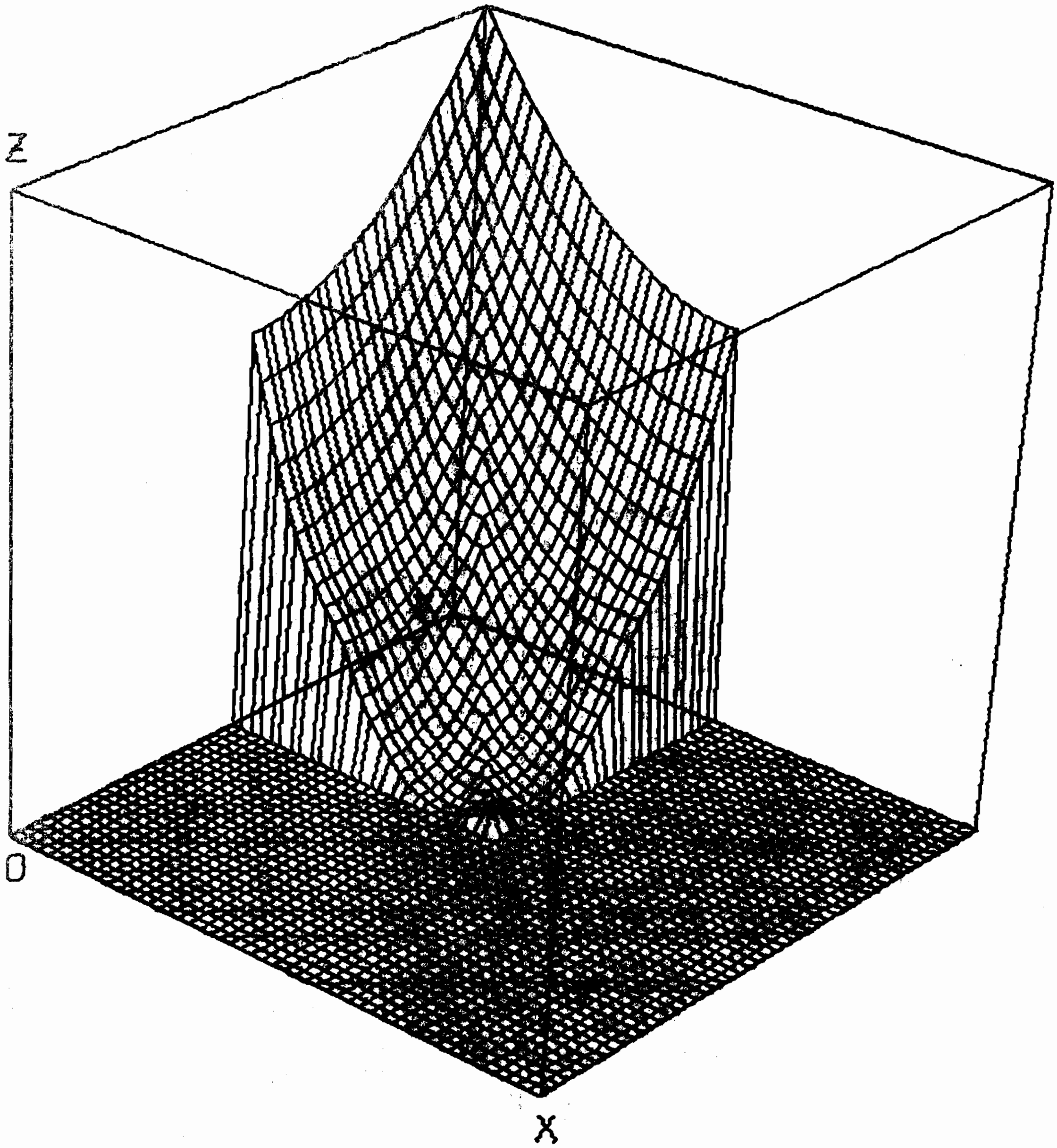

Figure 2.3

A Cutaway View of f(x,y) in Example 2.2

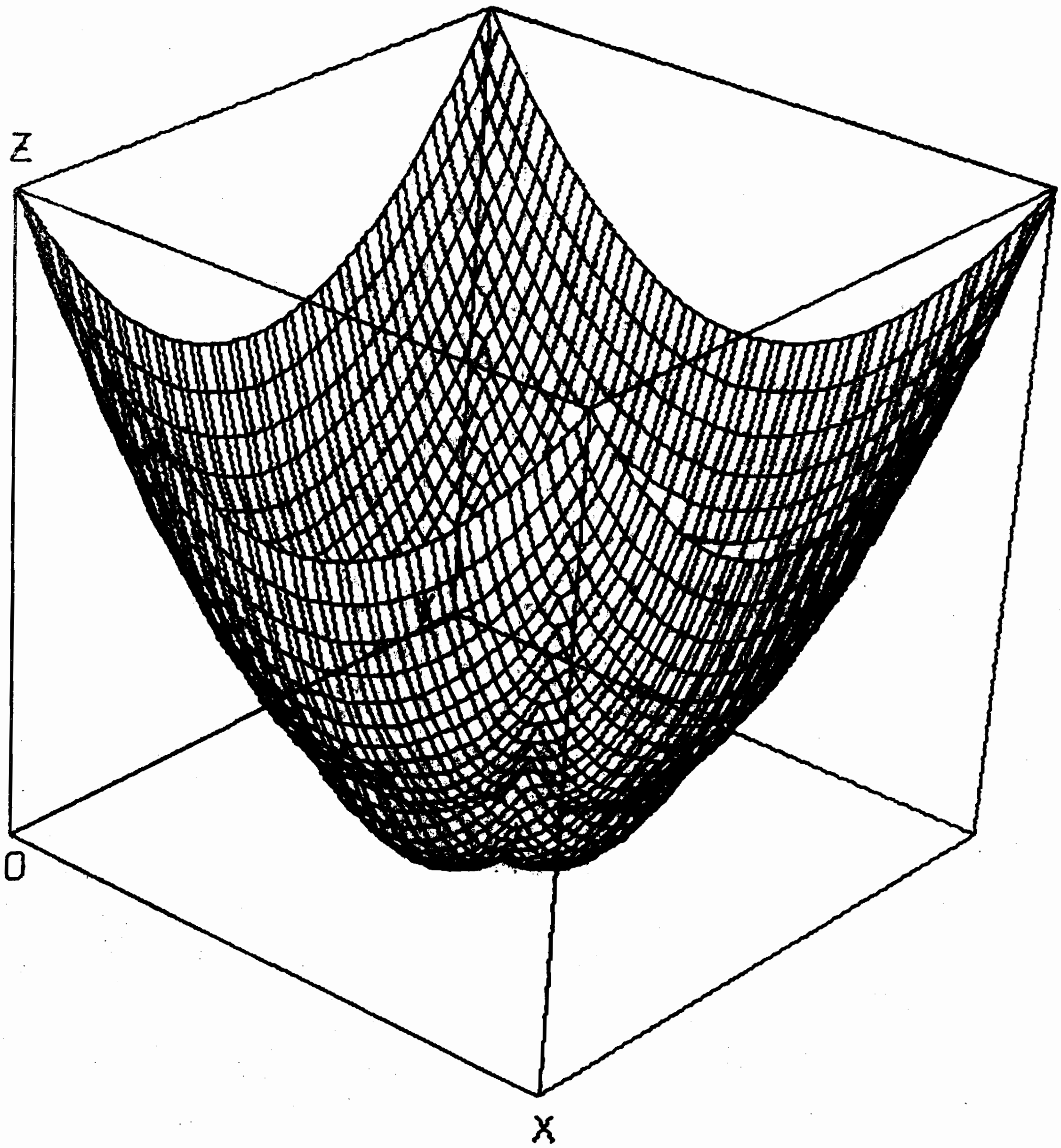

Figure 2.4

An Overall View of f(x,y) in Example 2.2

facilities two, three, and four. The minimum points can be expected to shift away from facility one somewhat, since the expected distance traveled has decreased for facility one, and has increased for the other existing facilities.

Table 2.2 lists the maximum and minimum values of $f(x,y)$ that were obtained in plotting Figures 2.3 and 2.4. The relative minima occur at $x^* = 50$, and within either the interval $38 < y < 42$ or the interval $58 < y < 62$, and by symmetry at $y^* = 50$, and within either the interval $38 < x < 42$ or the interval $58 < x < 62$.

In Example 2.2 the shifts in the minimum points and the values of $f(x,y)$ have changed due to the new probability assignments. It should be recognized, however, that the locations of the minimum points are dependent, not only on the probabilities of visiting various subsets of existing facilities, but also on the traveling salesman zones resulting from the configuration of the existing facilities. A traveling salesman zone can be defined as a region in which a new facility can be located such that when all other existing facility locations are held fixed, the set of optimal routes resulting from the solution of the traveling salesman problems remains the same. The symmetry of $f(x,y)$, as given in Examples 2.1 and 2.2, can be destroyed either by assigning subset probabilities so as to weight the existing facilities unequally or by altering the configuration of existing facilities, thus changing the traveling salesman zones.

Figure 2.5 shows, approximately, the different traveling salesman zones that result when the distance measure is Euclidean and the four existing facilities have the same locations as in the previous examples.

Table 2.2

Maximum and Minimum Values of Objective Function of Example 2.2

(x,y)	f(x,y)	Nature of Extreme Point
(0,0)	301.92	Global maximum
(0,100)	301.92	Global maximum
(100,100)	301.92	Global maximum
(100,0)	301.92	Global maximum
(50,50)	285.56	Local maximum
(50,40)	284.87	Minimum value obtained in grid search
(50,60)	284.87	Minimum value obtained in grid search
(40,50)	284.87	Minimum value obtained in grid search
(60,50)	284.87	Minimum value obtained in grid search

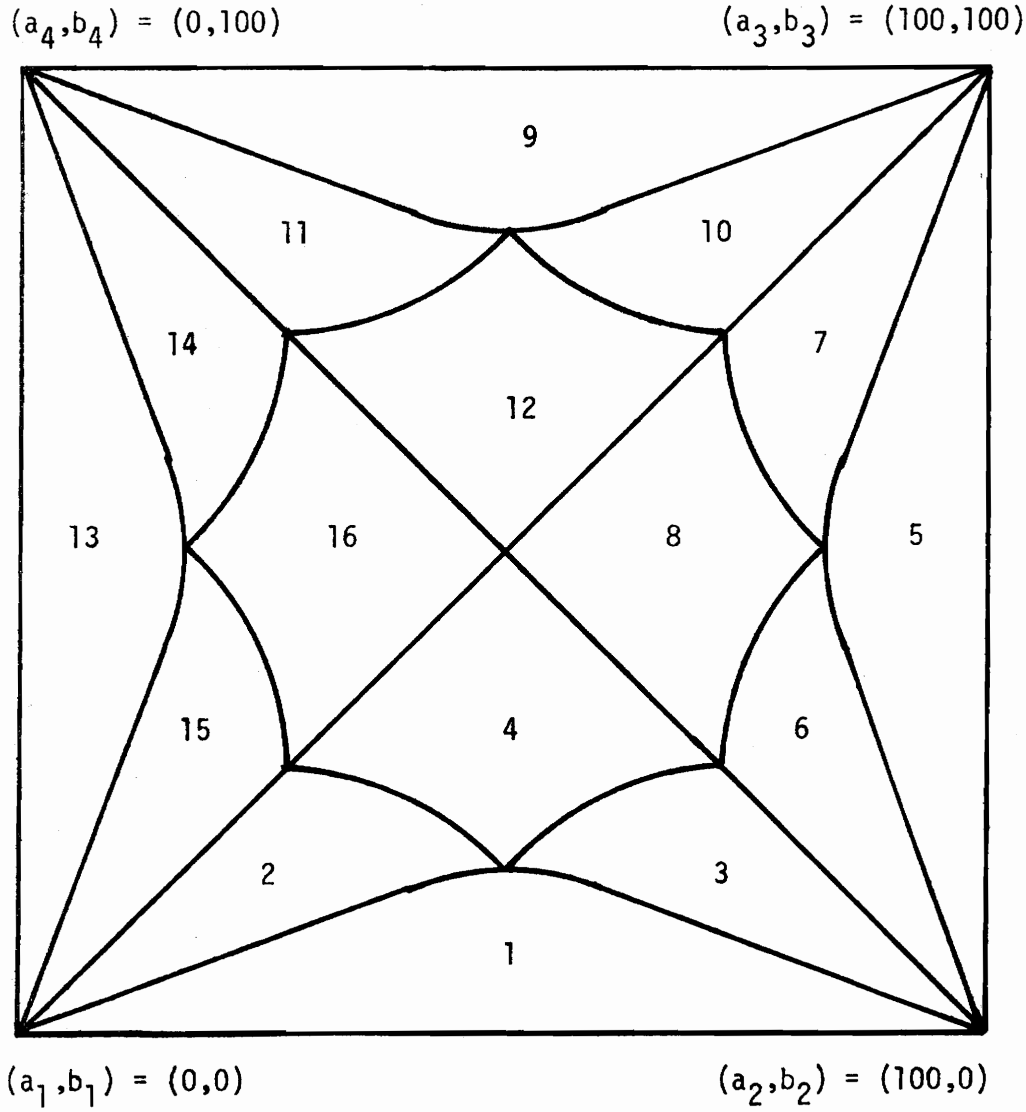

Figure 2.5

Traveling Salesman Zones

Table 2.3 lists the non-trivial itineraries for each zone. $I(S_h)$ is the optimal route or itinerary for the hth subset of facilities to be visited. Facility number 5, the new facility, is not given in the routes, $I(S_h)$, h = 11, 12, 13, 14, 15, since it is assumed that all tours begin and end with the new facility. It is interesting to note that when only all four existing facilities are visited, the traveling salesman zones turn out to be the four triangles formed by the diagonals of the square. This can be seen by comparing zone numbers in Figure 2.5 with the zones in Table 2.3 that have identical routes for $I(S_{15})$. When the four combinations of unique subsets containing three existing facilities are considered, the zones become distorted. Notice in Table 2.3 that traveling salesman zones change when at least one route, $I(S_h)$, changes. It is expected that as the total number of existing facilities is increased, resulting in many more subsets of existing facilities to be visited, the zones will become even more distorted. Furthermore, any change in the configuration of the existing facilities will tend to distort the traveling salesman zones. The importance of traveling salesman zones in relation to the traveling salesman location problem will become more apparent in Chapter 3.

Example 2.3 Consider as a third example using the equal subset probabilities of Example 2.1 and employing a rectilinear distance measure. The objective function now becomes pyramidal in shape. Figure 2.6 is a plot of f(x,y) over the entire square region. The point (50,50) has now become a global maximum with f(50,50) = 326.67. Any point lying on the boundary of the square region is a global minimum, i.e.,

Table 2.3

Itineraries for the Traveling Salesman Zones

Zone	$I(S_{11})$	$I(S_{12})$	$I(S_{13})$	$I(S_{14})$	$I(S_{15})$
1	1-3-2	1-4-2	1-4-3	2-3-4	1-4-3-2
2	1-2-3	1-4-2	1-4-3	2-3-4	1-4-3-2
3	1-3-2	2-1-4	1-4-3	2-3-4	1-4-3-2
4	1-2-3	2-1-4	1-4-3	2-3-4	1-4-3-2
5	2-1-3	2-1-4	1-4-3	2-4-3	2-1-4-3
6	2-1-3	2-1-4	1-4-3	2-3-4	2-1-4-3
7	1-2-3	2-1-4	1-4-3	2-4-3	2-1-4-3
8	1-2-3	2-1-4	1-4-3	2-3-4	2-1-4-3
9	1-2-3	2-1-4	3-1-4	3-2-4	3-2-1-4
10	1-2-3	2-1-4	1-4-3	3-2-4	3-2-1-4
11	1-2-3	2-1-4	3-1-4	2-3-4	3-2-1-4
12	1-2-3	2-1-4	1-4-3	2-3-4	3-2-1-4
13	1-2-3	1-2-4	1-3-4	2-3-4	1-2-3-4
14	1-2-3	2-1-4	1-3-4	2-3-4	1-2-3-4
15	1-2-3	1-2-4	1-4-3	2-3-4	1-2-3-4
16	1-2-3	2-1-4	1-4-3	2-3-4	1-2-3-4

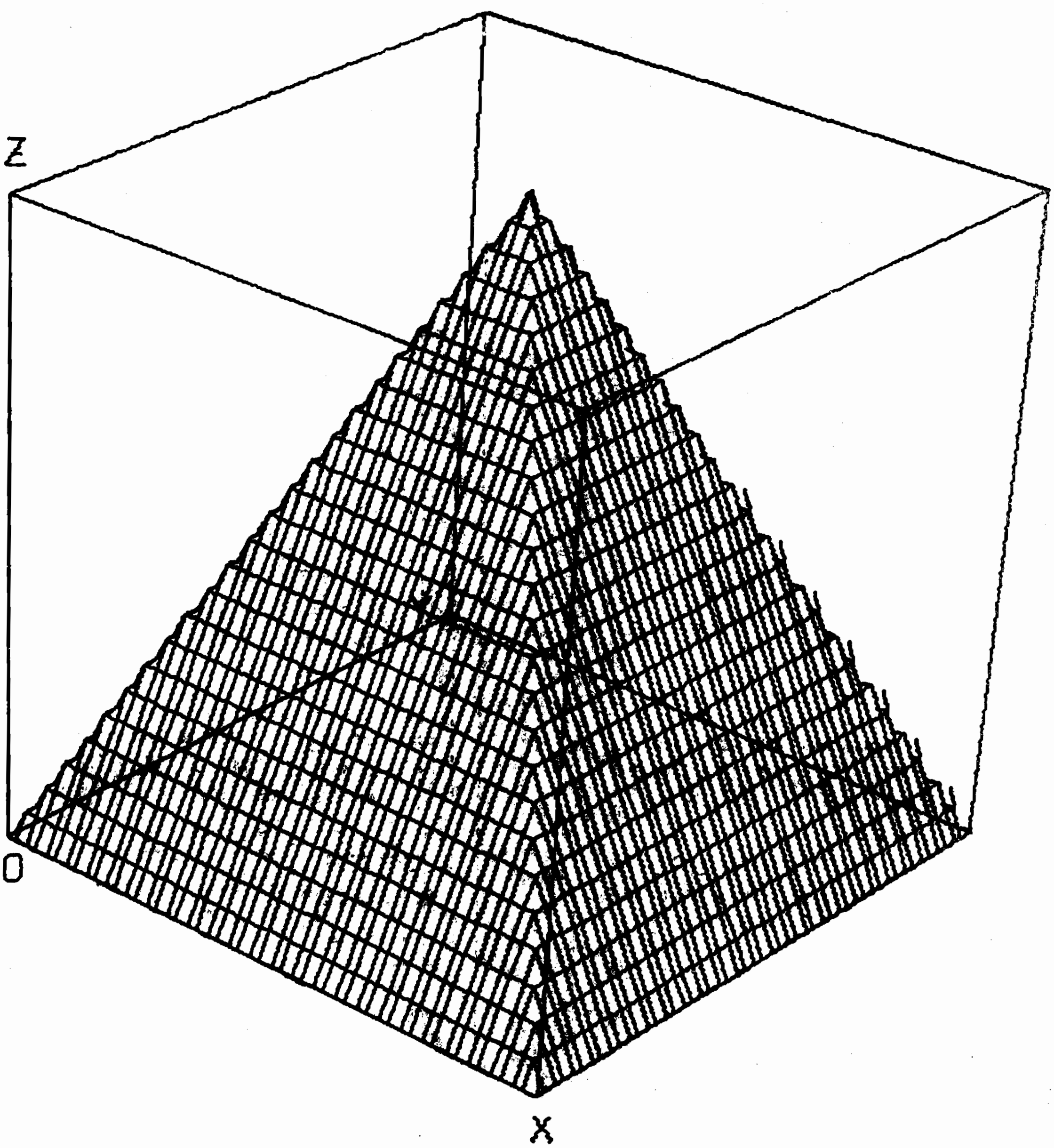

Figure 2.6

An Overall View of f(x,y) in Example 2.3

$(x^*,y^*) = \{(x,y) \mid 0 \leq x^* \leq 100,\ y^* = 0$ or $y^* = 100$, and $x^* = 0$ or $x^* = 100,\ 0 \leq y^* \leq 100\}$ with $f(x^*,y^*) = 320$.

Example 2.4 As the fourth example consider using the subset probabilities of Example 2.2 and employing rectilinear distances. The objective function forms a pyramid similar to the one in the previous example. The global maximum is still at the point (50,50) with $f(50,50) = 350$. The minimum points again lie on the boundary of the square region with $f(x^*,y^*) = 340$.

It is interesting to note that there are only four different traveling salesman zones in Examples 2.3 and 2.4 whereas 16 different zones were formed in Examples 2.1 and 2.2. When the rectilinear distance measure is used, the four traveling salesman zones are defined by the four right isosceles triangles formed by the diagonals of the square, e.g., one complete zone for the rectilinear problem is given by zones 1, 2, 3, and 4 in Figure 2.5.

Returning now to the results obtained in Examples 2.1 and 2.2, recall that as p_k increased from 0.067 to 0.1 that the four minimum points moved outward in directions perpendicular to the sides of the square region. Figures 2.7 and 2.8 provide a cutaway view and an overall view of $f(x,y)$, respectively, with $p_{15} = 0.6$, and $p_h \doteq .0286$, $h = 1, 2, \ldots, 14$. The minimum points have now moved even closer to the sides of the square. When $p_k = 1.0$, the optimal location for the new facility will lie anywhere on the boundary of the square. The reason behind this is simple. The boundary of the square represents the path of the optimal routes for travel between <u>all</u> four existing facilities given that all four existing are visited in one trip. Hence, any location of the new facility off

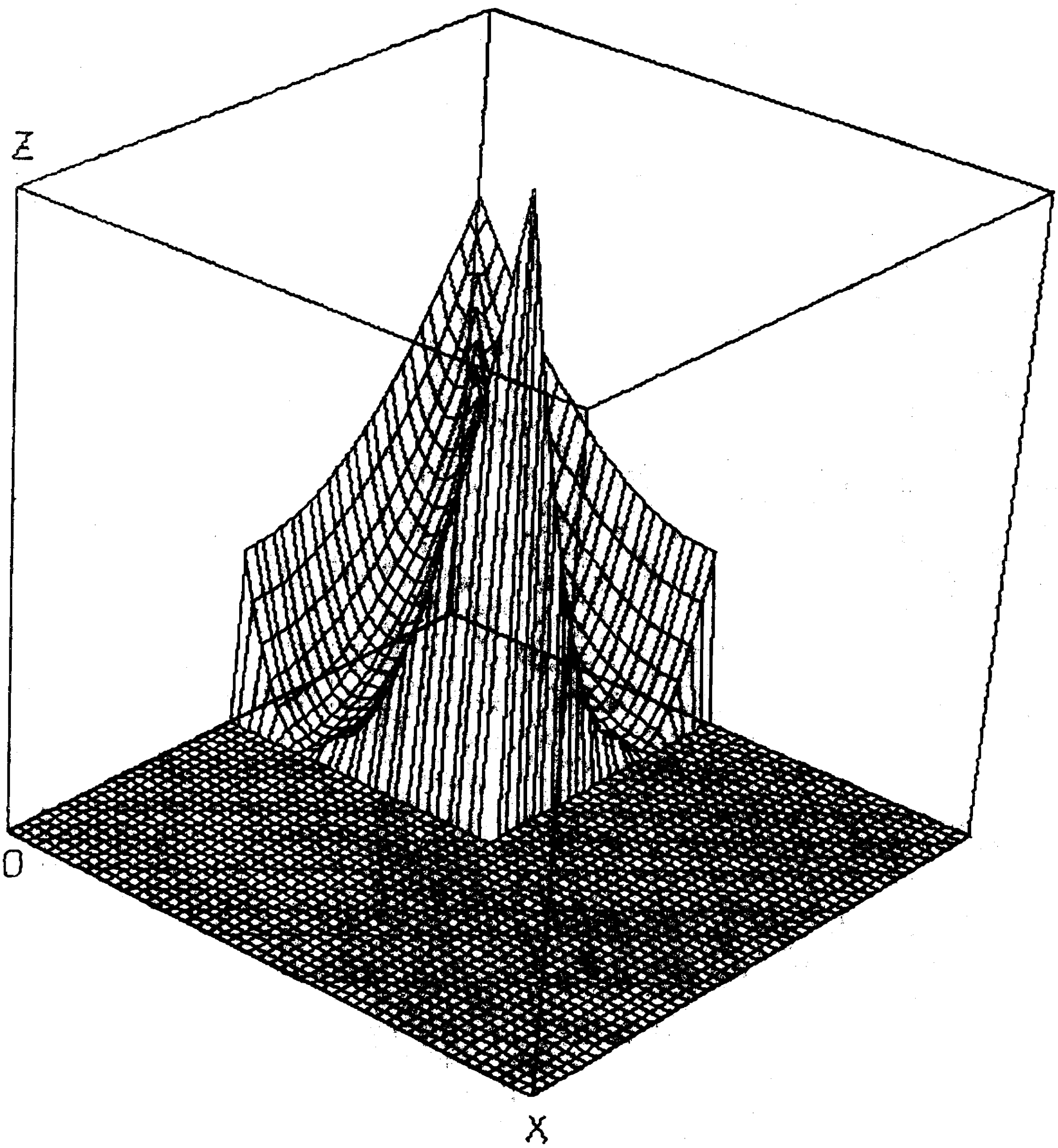

Figure 2.7

A Cutaway View of $f(x,y)$ when $p_{15} = .6$

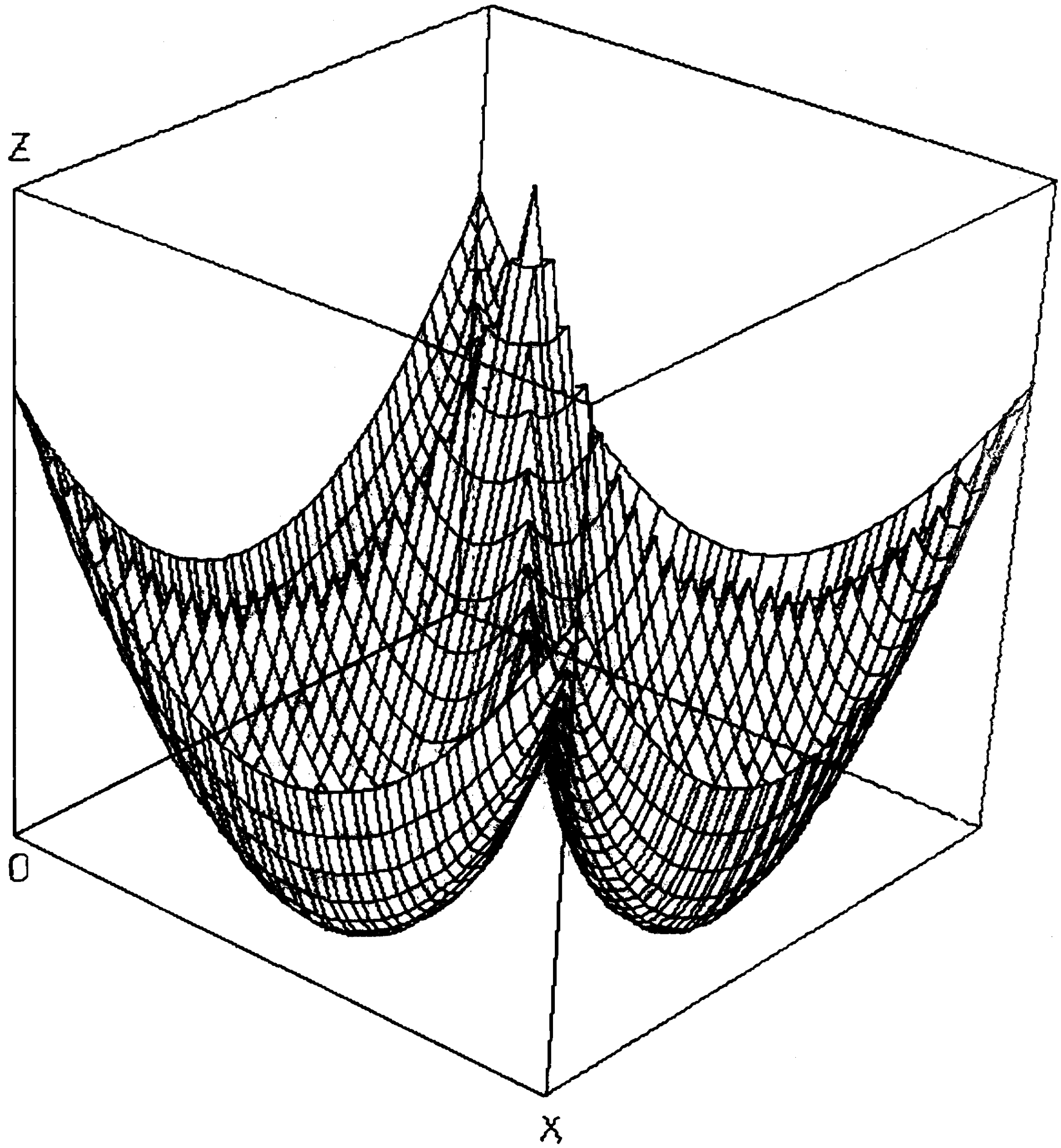

Figure 2.8

An Overall View of f(x,y) when $p_{15} = .6$

the boundary of the square can only increase the travel distance since it is known with certainty that all four existing facilities will be visited each trip. Notice in Figures 2.7 and 2.8 that the point (50,50) now represents a global maximum for f(x,y), and in addition, the ridges have become much sharper.

As was pointed out earlier, the symmetry of the objective function can be distorted either by assigning subset probabilities so as to unequally weight existing facilities or by changing the configuration of the existing facilities. It is instructive to examine the paths of the relative minima of f(x,y), for $0 \leq p_k \leq 1$, for different configurations of existing facilities.

Consider the square configuration given by the four existing facility locations in Examples 2.1 and 2.2. When all p_h, $h < k$, are held equal, and $p_k \to 0$, the resulting minimum points of f(x,y) trace out the paths given in Figure 2.9. Each point on an arrow represents a global minimum point for f(x,y) for a particular p_k. When $p_k = 0$, the single global minimum is at point (50,50). As previously mentioned, when $p_k = 1$, any point on the boundary of the square represents a global minimum. For a particular p_k, $0 < p_k < 1$, there are four global minimum points, one on each arrow, with each global minimum point located the same distance away from the point (50,50). It should be emphasized here that all four paths trace out global minimum points as $p_k \to 0$.

In general, if the configuration of m existing facilities forms a regular polygon, and all m facilities are weighted equally, then as $p_k \to 0$, there will be m paths traced out by the minimum points of f(x,y), all m of which represent global minima. A regular polygon is defined to

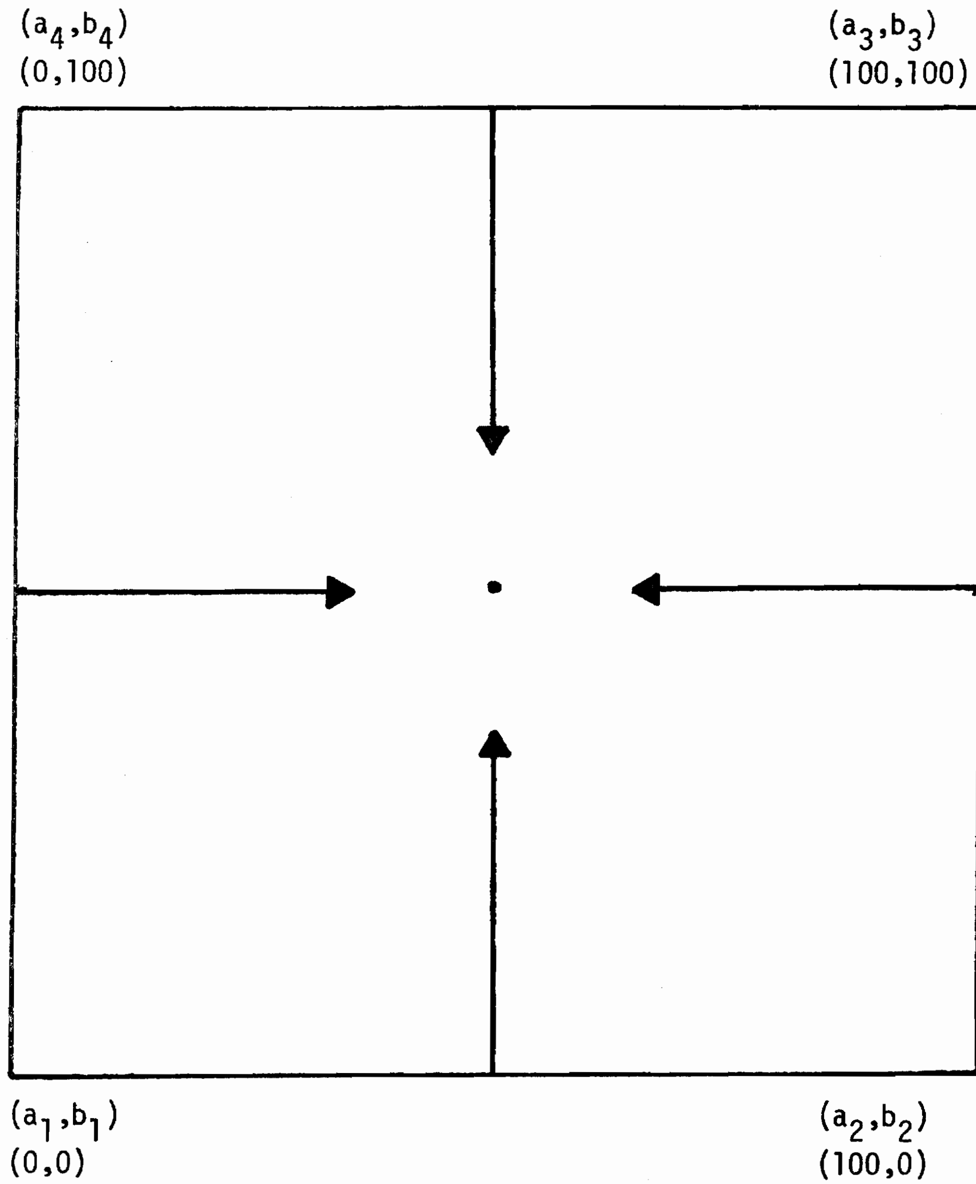

Figure 2.9

Optimal Paths as $p_k \to 0$ in Example 2.2

be a closed figure with all sides of equal length located symmetrically about the "center" of the polygon. If the polygon is not regular, then the paths may not all represent global minima. This is the case in Figure 2.10. In Figure 2.10 the configuration of the six existing facilities shown forms a hexagon. Since the two vertical sides of the hexagon are shorter than the other sides, the hexagon is not regular. The broken lines starting from the shorter sides represent local minimum paths traced out by f(x,y) as $p_k \to 0$, while the solid lines represent global minimum paths. Since the hexagon is not regular, the traveling salesman zones are not symmetric, and therefore f(x,y) is not symmetric.

The three existing facilities in Figure 2.11 are located so as to form an equilateral triangle. The traveling salesman zones are symmetric about the center of the triangle, and as a result, f(x,y) has three global minimum paths that converge on the "center" of the triangle as $p_k \to 0$. Moving any existing facility will distort the symmetry of the traveling salesman zones, and therefore, will cause f(x,y) to be asymmetric. This, in turn, will reduce the number of global minimum paths to either two or one, depending on the direction in which the existing facility is moved.

For example, if (a_3,b_3) is moved toward (a_1,b_1) and (a_2,b_2) in any fashion, only <u>one</u> global minimum path occurs, its location being near (a_3,b_3). On the other hand, if (a_3,b_3) is moved away from (a_1,b_1) and (a_2,b_2) in any fashion, <u>two</u> global minimum paths will result, one located near (a_1,b_1), and one near (a_2,b_2).

Not only can a non-regular configuration of existing facilities reduce the <u>number</u> of global minimum paths, but also it may alter the <u>direction</u> of the paths as well. The global minimum paths in Figures 2.12

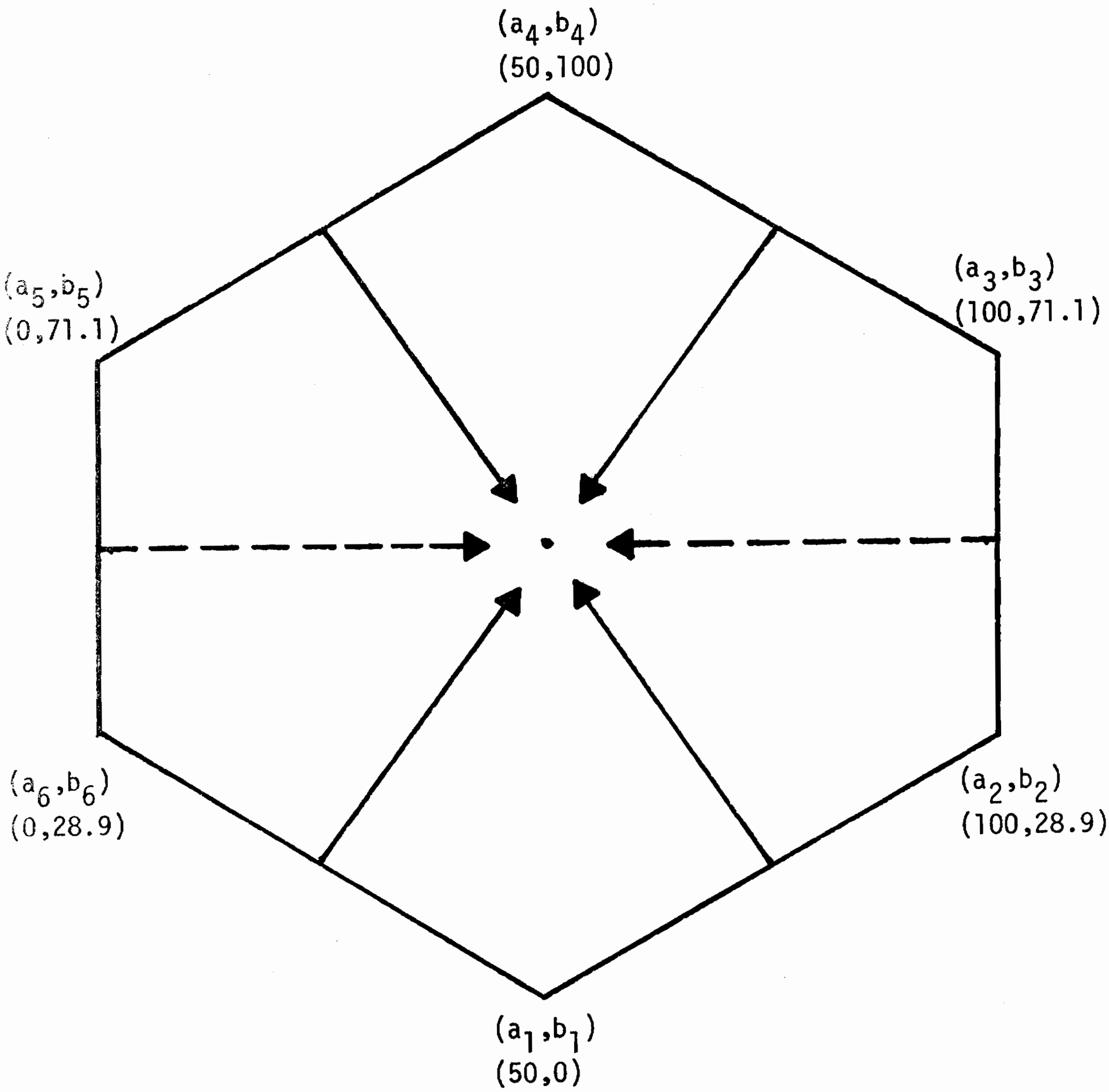

Figure 2.10

Optimal Paths as $p_k \to 0$, for $m = 6$

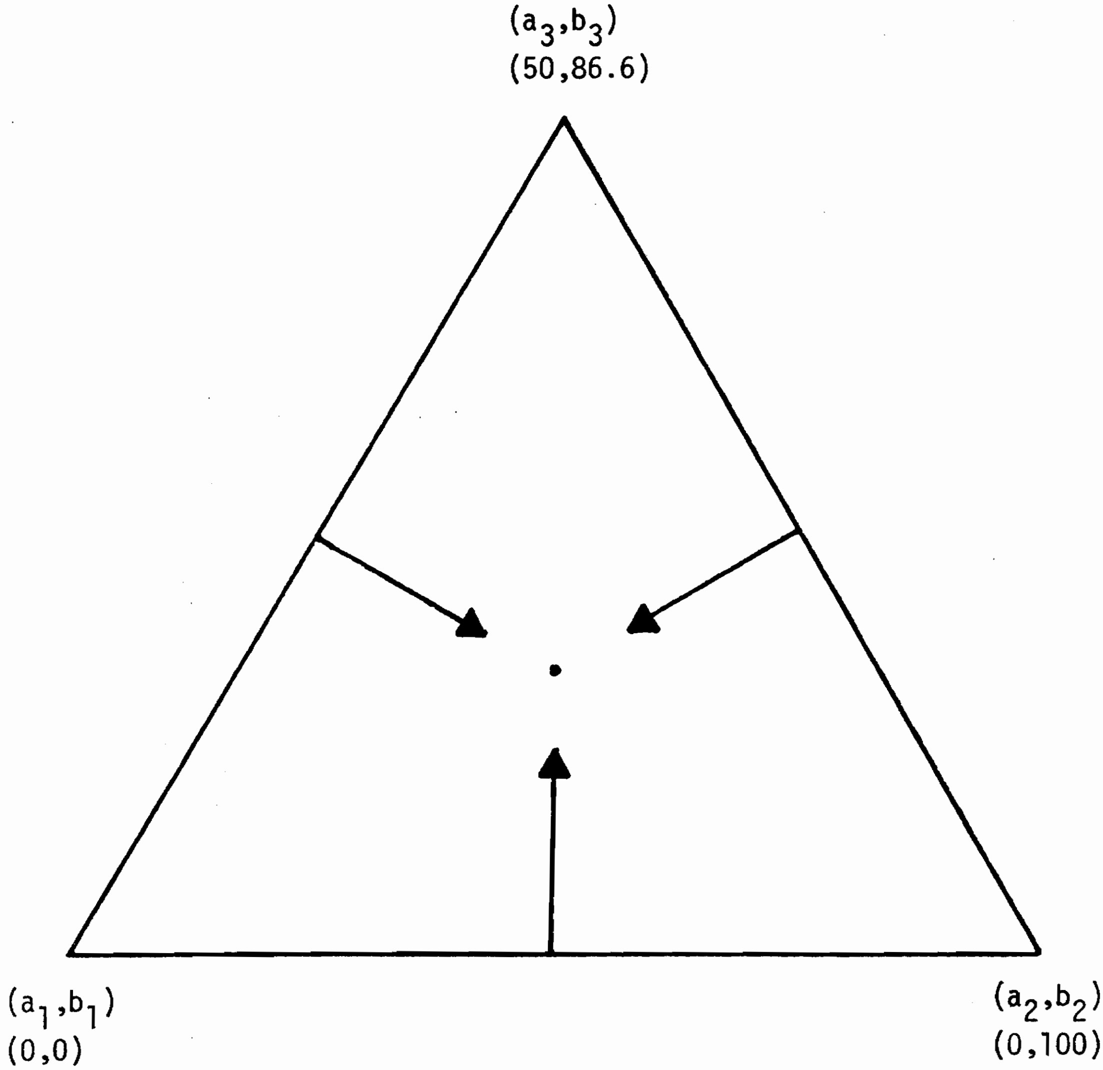

Figure 2.11

Optimal Paths as $p_k \to 0$, for m = 3

and 2.13 were approximated by letting p_k go to zero and employing the first search procedure, presented in the next chapter, to locate relative and global minimum points. Figure 2.12 shows the resulting distortion of the optimal paths when existing facility number three (Examples 2.1 and 2.2) is moved from the point (100,100) to the point (150,150). As before, the broken lines depict relative minimum paths, and the solid lines are global minimum paths. Notice that when $p_k = 0$, the single optimal location is still at the point (50,50), just as it would be if the configuration of existing facilities had formed a square. The global minimum paths tend to favor the three existing facilities that have not been moved.

When the third existing facility is moved to the point (25,25), the optimal (global) paths remain inside the closed figure. Observe in Figure 2.13 that the local minimum paths fall within the convex hull formed by the four existing facilities, however. At $p_k = 0$, the global minimum is at (25,25).

It should now be clear that the location of the minimum point, or minimum points for f(x,y) is heavily dependent not only on the subset probabilities, but also on the configuration of the existing facilities. Determining the minimum points for f(x,y), even for small problems such as those presented in this chapter, turns out to be an exceedingly difficult task. Many of the results obtained in the examples thus far have been counter-intuitive. Fortunately, however, one of the search procedures advanced in the next chapter is able to deal with the psychotic nature of f(x,y) through the use of information given by the traveling salesman zones.

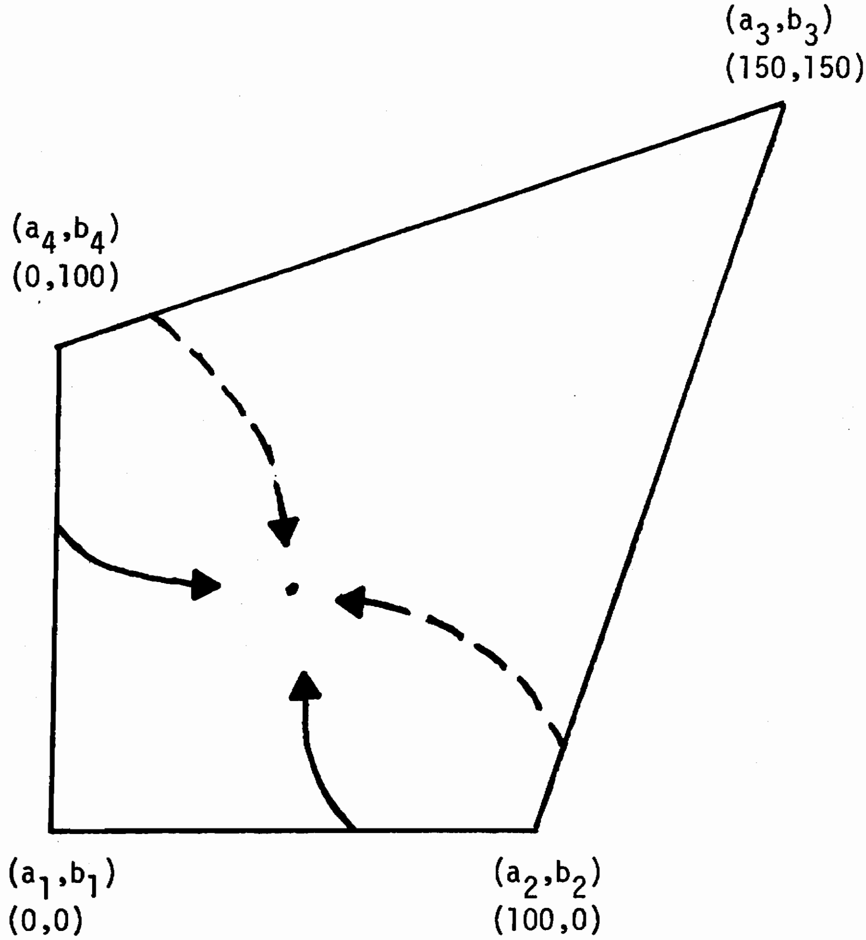

Figure 2.12

Optimal Paths as $p_k \to 0$, for an Asymmetric Configuration with $m = 4$

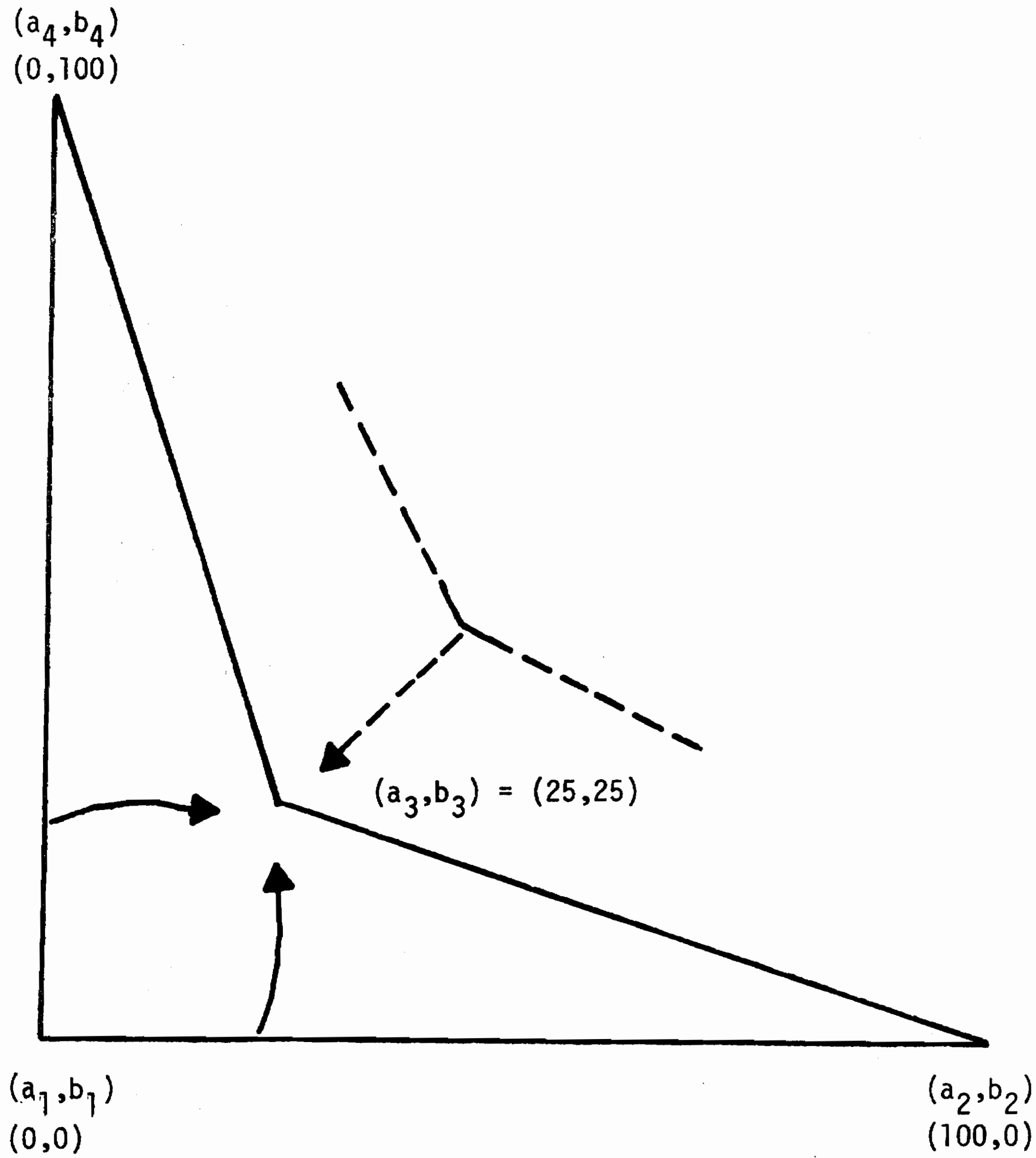

Figure 2.13

Optimal Paths as $p_k \to 0$, for an Asymmetric Configuration with $m = 4$

Summary

Chapter 2 has presented the mathematical formulation of the traveling salesman location problem. Additionally, several simple example problems were provided to better familiarize the reader with the nature of the problem under investigation. The concept of a traveling salesman zone, a term that will be used throughout this thesis, was introduced via the example problems.

The next chapter is devoted to the implementation of two existing search procedures as a means of solving the traveling salesman location problem. The example problems discussed in Chapter 2 will be used again in Chapter 3 to verify the two solution procedures.

Chapter 3

SOLUTION PROCEDURES

Introduction

In the previous chapter the traveling salesman location problem was introduced, first by way of a mathematical formulation of the problem, and then by examining several relatively simple example problems. Even for a simple problem, it was seen that the objective function, f(x,y), is non-convex. The non-convexity of f(x,y) was shown to be caused by the different traveling salesman zones formed at various coordinate locations.

Two heuristic procedures for solving the traveling salesman location problem will be given in this chapter. The first procedure is based on a structural relationship between the traveling salesman location problem and the well known Steiner-Weber problem. The second procedure proposed utilizes a search by regression. After the presentation and discussion of the algorithms, the effectiveness of each will be demonstrated by solving the problems given in Examples 2.1, 2.2, 2.3, and 2.4.

Procedure 1: Functional Minimization Using the Hyperbolic Approximation Procedure

The procedure for solving the traveling salesman location problem developed in this section is capable of producing solutions that represent local minima on f(x,y). As is usually the case when search techniques are employed to minimize non-convex functions, there can be no absolute guarantee that a global minimum will be found. It will be shown by way of the example problems that a judicious selection of several starting points for the new facility greatly increases the chances of obtaining a global minimum.

All effective heuristic procedures are capable of obtaining optimal solutions. The strength of Procedure 1 lies in its ability both to produce an optimal solution and to immediately recognize that solution as being an optimal one. Many heuristic procedures do not possess the latter trait.

Relationship Between the Steiner-Weber Problem and the Traveling Salesman Location Problem

The development of Procedure 1, based on an interesting relationship between the Steiner-Weber problem and the traveling salesman location problem, is an extension of an approach originally suggested by Lohmar [16]. This relationship is best exposed by first discussing the Steiner-Weber problem.

The single facility location problem, a special case of the Steiner-Weber problem, can be stated mathematically as

$$\text{minimize } g(x,y) = \sum_{i=1}^{m} w_i d_i(x,y) \tag{3.1}$$

where

m = the number of existing facilities

$d_i(x,y)$ = the distance from the new facility, m+1, located at coordinate position (x,y), to existing facility i, $1 \leq i \leq m$.

w_i = non-negative weight between the new facility and the ith existing facility, $1 \leq i \leq m$.

$d_i(x,y)$ is given by the first equation in (2.4) for Euclidean distances and is given by the first equation in (2.5) for rectilinear distances. The weighting factor, w_i, reflects the percentage of direct movement per unit time between the ith existing facility and the new facility. The coordinate positions of the existing facilities are (a_i,b_i), i = 1, 2, 3, ..., m, as given in the previous chapter.

A sufficient condition for g to have a minimum at a point (x^*,y^*) is that the first partial derivatives of g, with respect to x, and with respect to y, evaluated at (x^*,y^*) both be zero. When the Euclidean distance measure is used, the first partial derivatives are

$$\frac{\partial g(x,y)}{\partial x} = \sum_{i=1}^{m} w_i(x-a_i)/[(x-a_i)^2+(y-b_i)^2]^{1/2} \tag{3.2}$$

$$\frac{\partial g(x,y)}{\partial y} = \sum_{i=1}^{m} w_i(y-b_i)/[(x-a_i)^2+(y-b_i)^2]^{1/2} \tag{3.3}$$

Equations (3.2) and (3.3) could be solved by numerical techniques if the partial derivatives were defined at all points. Notice, however, that if the new facility location coincides with the location of an existing facility, the partial derivatives are undefined. Several solution techniques have been developed to handle the problem of undefined derivatives in the Steiner-Weber problem. The Hyperbolic Approximation Procedure, proposed by Eyster, White, and Wierwille [8], has proven to be an especially effective algorithm for solving the facilities location problem. Their procedure will be utilized in Procedure 1 to assist in the solution of the traveling salesman location problem. Essentially, the Hyperbolic Approximation Procedure minimizes $g(x,y)$ by roughly approximating the right circular cones in Equation (3.1) with hyperboloids, thus forcing the derivatives to be defined over the entire space, minimizing the resulting function, and then sequentially making closer approximations until an optimal solution is obtained. A complete description of the algorithm and a presentation of computational experience is given in [8].

The relationship between the single facility location problem and the traveling salesman location problem will now be given. Consider an initial location, (x^0,y^0), for the new facility in the traveling salesman location problem. In the process of determining the expected total distance traveled, $f(x^0,y^0)$, the optimal routing for each distinct subset of facilities to be visited is also determined. Recall that each of these routes, or itineraries, was previously defined in Chapter 2 by $I(S_h)$, $h = 1, 2, 3, \ldots, k$, and that while the order of visitation was given by $I(S_h)$, the new facility was excluded from $I(S_h)$. Since it is assumed that all tours begin and end with the new facility, there is no need to include it in $I(S_h)$. For a specific subset, S_h, the only existing facilities that interact directly with the new facility, m+1, are the first and last existing facilities in the optimal route. These existing facilities are given by the first and last elements of $I(S_h)$. In order to facilitate the discussion that follows, some additional notation is required. Specifically, let

$$r_{ih} = \begin{cases} 1, \text{ if the first or last element of } I(S_h) \text{ equals } i \\ 0, \text{ otherwise.} \end{cases} \qquad (3.4)$$

Note that $R = ((r_{ih}))$ is an m x k matrix with each of the first m columns having (m-1) zero entries and a single entry of one, and with each of the remaining (k-m) columns having (m-2) zero entries and two entries of one. The first m columns have only one entry of one because the first m subsets of facilities to be visited contain only one existing facility, and consequently, the first element of $I(S_h)$, h = 1, 2, 3, ..., m, is also the last element of $I(S_h)$. Therefore, for each of the first m subsets of existing facilities to be visited the new facility will interact directly with a particular existing facility twice, once on the trip to the existing facility, and once on the return trip. After the first m subsets, however, there are two or more existing facilities contained in S_h, and since traveling salesman tours are required, the new facility must interact directly with two, and only two, existing facilities. The direct interaction takes place between the first existing facility visited and the last existing facility visited. This explains why the R matrix is composed of all zeroes and ones.

The R matrix is determined by solving the "string" of traveling salesman problems when the new facility is located at the coordinate position (x,y). For example, consider the case where m = 4, and k = 2^m-1 or 15. The R matrix may appear as follows:

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

The first 10 columns of R are strictly determined since the solutions to the traveling salesman problems involving one or two existing facilities are trivial. The entries in the last five columns of R are not strictly determined, but rather are dependent on the optimal routes given by the solutions to the traveling salesman problems for S_{11}, S_{12}, S_{13}, S_{14}, and S_{15}. It was demonstrated in Chapter 2 that the determination of the non-trivial routes is dependent on the configuration of existing facilities and the particular coordinate location of the new facility. Recall that in this case, the non-trivial routes are given by one of 16 possible traveling salesman zones, depending on the new facility location.

Given the initial location of the new facility, (x^0, y^0), and the resulting itineraries, $I^0(S_h)$, h = 1, 2, 3, ..., k, the new facility can be optimally located, inasmuch as the direct interaction between new and existing facilities is concerned. This is accomplished by minimizing g(x,y) in Equation (3.1), the single facility location problem. Each of the appropriate weights to be used is given by

$$w_i = 2p_i + \sum_{h=m+1}^{k} p_h r_{ih} \qquad , i = 1, 2, 3, \ldots, m \tag{3.5}$$

where p_h is the probability of visiting subset S_h, and r_{ih}, Equation (3.4), is the indicator variable for direct interaction between the new facility and existing facility i of S_h.

To illustrate the method of forming the weights consider existing facility number 1 in the case where m = 4. $S_1 = \{1,5\}$, and S_1 is visited with probability p_1. The resulting itinerary is 5-1-5, i.e., $I(S_1) = \{1\}$, so that for this subset, existing facility 1 interacts directly with the new facility twice. Hence, w_1 is at least as large as $2p_1$. Table 3.1 lists the subsets of facilities to be visited and the itineraries that are strictly determined. Recall, an itinerary is strictly determined if there is only one possible Hamiltonian circuit for that itinerary. Observe in Table 3.1 that facility 1 is not involved in another itinerary until the fifth subset, and then it is involved in the fifth, sixth, and seventh subsets, whose itineraries are strictly determined. As a result, w_1 is now at least as large as $2p_1 + p_5 + p_6 + p_7$. Beyond this point, however, it cannot be determined whether or not facility 1 will directly interact again with the new facility without knowing the first and last elements of the non-trivial itineraries, $I(S_{11})$, $I(S_{12})$, $I(S_{13})$, and $I(S_{15})$. The non-trivial itineraries, and, therefore, the r_{jh}'s cannot be determined for any new facility location without solving the associated set of traveling salesman problems. It is in this sense that the weights, w_j, $j = 1, 2, 3, \ldots, m$, can be viewed as random variables.

Suppose now that the new facility is optimally located with respect to the direct interaction between new and existing facilities in accordance with Equation (3.1), using the weights given by Equation (3.5). As a result of solving the Steiner-Weber problem, assume that the new facility has been moved from (x^0,y^0) to (x^1,y^1). If the routes obtained in solving $f(x^1,y^1)$ are identical to the routes that occurred at (x^0,y^0), then the new facility has also been optimally located with respect to the indirect interaction between the new facility and the existing facilities. As a result, $f(x,y)$ will have a local minimum at (x^1,y^1). To see this, consider the internal distances given by each non-trivial route, $I(S_h)$, i.e., the distance from the first element of $I(S_h)$ to the

Table 3.1

Subsets of Facilities to be Visited and Trivial Itineraries for the case where m = 4

h	S_h	$I(S_h)$ (if strictly determined)
1	{1,5}	{1}
2	{2,5}	{2}
3	{3,5}	{3}
4	{4,5}	{4}
5	{1,2,5}	{1,2}
6	{1,3,5}	{1,3}
7	{1,4,5}	{1,4}
8	{2,3,5}	{2,3}
9	{2,4,5}	{2,4}
10	{3,4,5}	{3,4}
11	{1,2,3,5}	-
12	{1,2,4,5}	-
13	{1,3,4,5}	-
14	{2,3,4,5}	-
15	{1,2,3,4,5}	-

second, plus the distance from the second to the third, plus the distance from the third to the fourth, and so on, plus the distance from the next to the last element to the last element. If the itineraries associated with $f(x^0,y^0)$ are identical with those associated with $f(x^1,y^1)$, then the internal distances must be the same, and since (x^1,y^1) is the minimum distance location for the direct interaction between the new and existing facilities, $f(x^1,y^1)$ must be the minimum value of $f(x,y)$ relative to all (x,y) in the neighborhood of (x^1,y^1).

This relationship between the Steiner-Weber problem and the traveling salesman location problem suggests an algorithm for minimizing $f(x,y)$. A general description of Procedure 1 follows: Select a starting position, (x^0,y^0), for the new facility, evaluate $f(x^0,y^0)$ by solving the associated traveling salesman problems, relocate the new facility to (x^1,y^1) using the Hyperbolic Approximation Procedure (HAP), evaluate $f(x^1,y^1)$, and compare the sets of itineraries for (x^0,y^0) and (x^1,y^1). If the sets of itineraries are the same, then a local minimum has been found at (x^1,y^1). If the sets of itineraries are not the same, compute the new weights, relocate the new facility to (x^2,y^2) using HAP, evaluate $f(x^2,y^2)$, and compare the itineraries corresponding to (x^1,y^1) and (x^2,y^2). Continue in this manner until a relative minimum has been found or until a predetermined time limit is exceeded.

Before formally presenting the algorithm an important point needs to be emphasized. First, suffice it to say that if successive sets of itineraries are the same, i.e., if $I^{\ell}(S_h) = I^{\ell+1}(S_h)$, $h = 1, 2, 3, \ldots, k$, after moving the new facility from (x^{ℓ}, y^{ℓ}) to $(x^{\ell+1},y^{\ell+1})$, then a local minimum point for $f(x,y)$ has been obtained. Furthermore, if $I^{\ell}(S_h)$

$= I^{\ell+1}(S_h)$, then the weights, w_i, i = 1, 2, 3, ..., m, formed at (x^ℓ,y^ℓ) must be identical to the weights formed at the point $(x^{\ell+1},y^{\ell+1})$. The only way that the weights can be the same for new facility locations is if the two locations are in the same traveling salesman zone. Hence, if the weights corresponding to the new facility location (x^ℓ,y^ℓ) are equal to the weights at new facility location $(x^{\ell+1},y^{\ell+1})$, then $(x^{\ell+1},y^{\ell+1})$ represents a local minimum point for f(x,y). In other words, a check for optimality at each iteration can be performed either by comparing the routes or by comparing the weights.

Figure 3.1 gives a macro-flow chart of the check for optimality. To see that the check for a local minimum point can be made by the examination of either the route vectors or the weight vectors suppose the new facility is located at (x^ℓ,y^ℓ), where ℓ denotes the iteration number. In the process of evaluating the objective function, TS^ℓ, the optimal routes are also obtained. The new facility is moved to $(x^{\ell+1},y^{\ell+1})$ by using the weights, W^ℓ, and the Hyperbolic Approximation Procedure. Suppose that the routes obtained at $TS^{\ell+1}$ are the same as those obtained at TS^ℓ. If this is the case, then $(x^{\ell+1},y^{\ell+1})$ must be a local minimum point. Furthermore, the weights at $W^{\ell+1}$ will be the same as those that were formed at W^ℓ. Consequently, if the new facility is moved to $(x^{\ell+2},y^{\ell+2})$ it will be discovered that $(x^{\ell+2},y^{\ell+2})$ is the same coordinate position as $(x^{\ell+1},y^{\ell+1})$. This must be the case since the weights at $W^{\ell+1}$ are equal to the weights at W^ℓ, and g(x,y) has already been minimized at iteration ℓ. Hence, it makes no difference whether the optimality test is made by comparing the routes determined at TS^ℓ and $TS^{\ell+1}$ or by comparing the weights formed at W^ℓ and $W^{\ell+1}$.

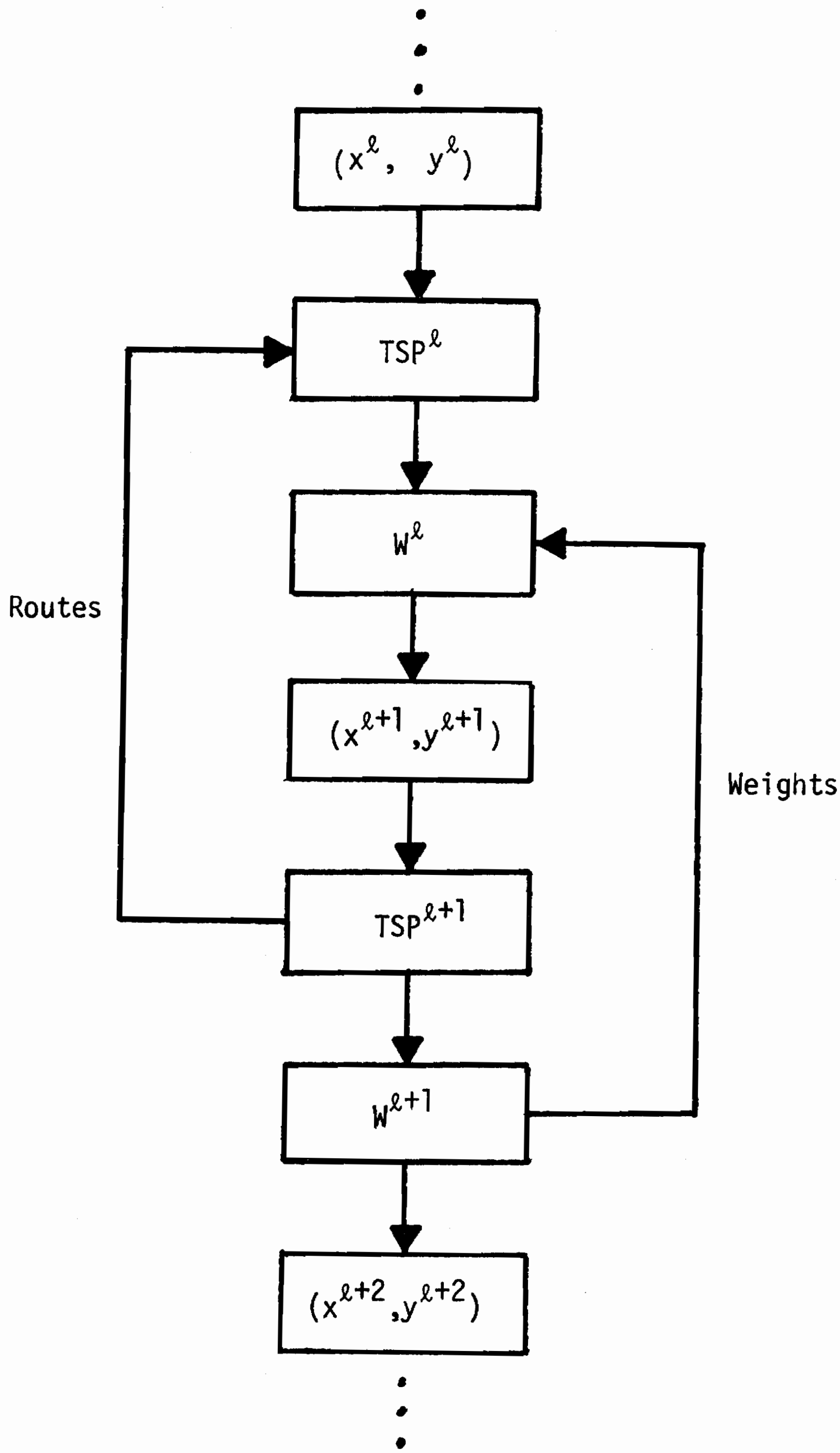

Figure 3.1

Equivalent Checks for Optimality in Procedure 1

Presentation of the Algorithm

Procedure 1 will now be presented in a formal, step-wise fashion. A macro-flow chart is given in Figure 3.2.

Procedure 1:

1. Set $\ell = 0$. Select an initial starting location for the new facility, $(x^\ell, y^\ell) = (x^0, y^0)$, and evaluate $f(x^0, y^0)$.
2. Compute W^0, the weight factor reflecting the level of direct activity between the new and existing facilities when the new facility is located at (x^0, y^0).
3. Set $\ell = \ell+1$. Relocate the new facility to (x^ℓ, y^ℓ) using $W^{\ell-1}$ and the Hyperbolic Approximation Procedure.
4. Evaluate $f(x^\ell, y^\ell)$, and compute W^ℓ, based on the itineraries, $I^\ell(S_h)$, $h = 1, 2, 3, \ldots, k$, and the subset probabilities.
5. Compare the weight vectors between successive iterations: if $W^\ell = W^{\ell-1}$, go to 7. Otherwise, go to 6.
6. If ℓ = the maximum number of iterations allowed, go to 8. Otherwise, go to 3.
7. $(x,y)^\ell$ represents a local minimum point for $f(x,y)$. Write out the minimum point and $f(x^\ell, y^\ell)$. Go to 9.
8. Terminate search: write out the lowest $f(x,y)$ value found. Go to 9.
9. Stop.

Example 3.1 To emphasize the importance of selecting several different initial locations for the new facility when employing Procedure 1, several starting points will be used on Example 2.1. Recall that in

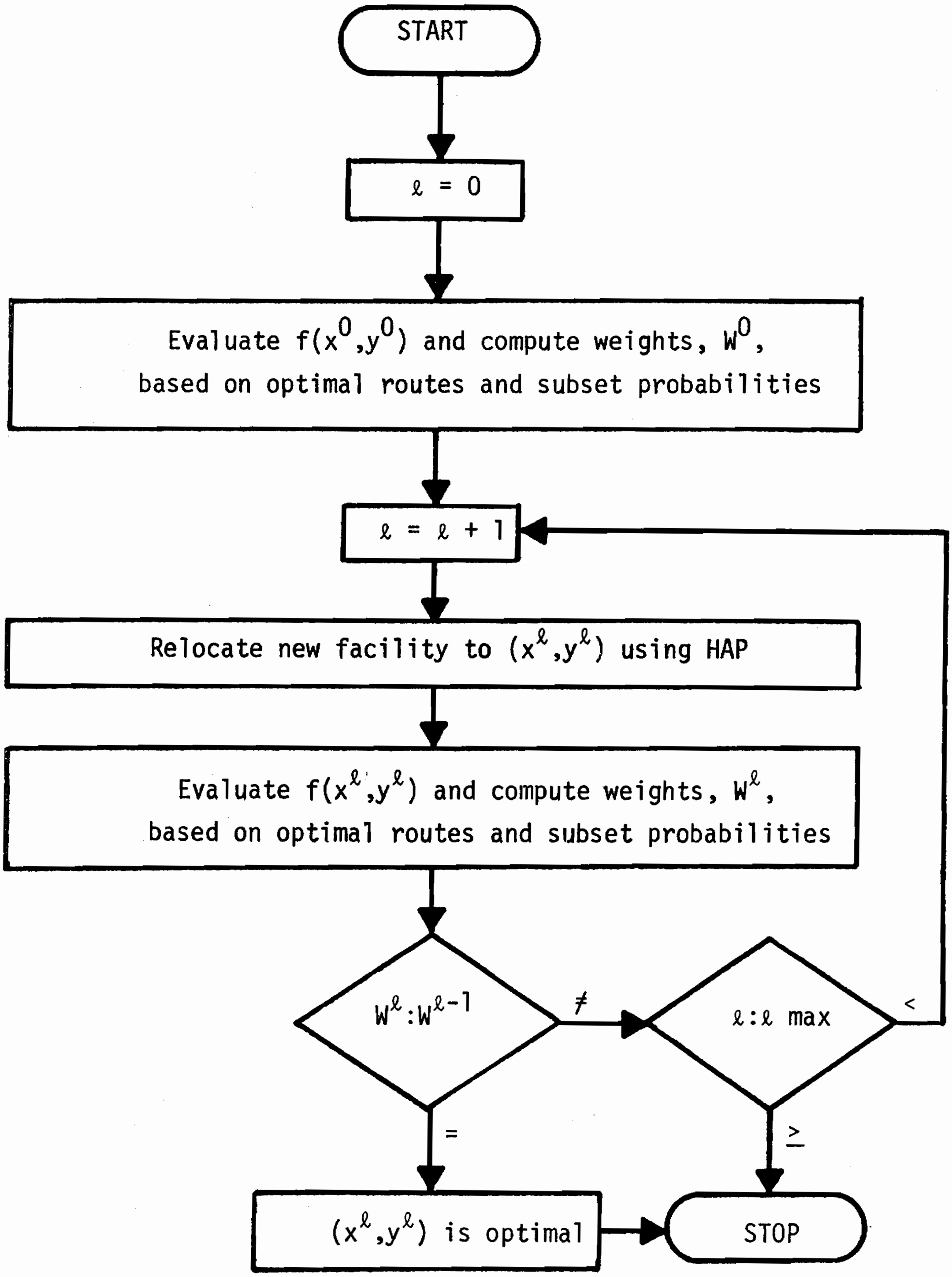

Figure 3.2

Macro-Flowchart of Procedure 1

Example 2.1 the distance measure was Euclidean, all subset probabilities were equal, and that the four existing facilities were located at coordinate positions (0,0), (100,0), (100,100), and (0,100), respectively. Table 3.2 lists 10 different initial locations for the new facility and the resulting paths to the optimum. Figure 3.3 and Figure 3.4 provide a graphical view of the movement of the new facility at each iteration, for each of the 10 initial locations. Observe that all of the final minimum distance locations fall within one of the intervals of uncertainty for the optimal locations given in Example 2.1. In all cases the objective function value, $z^* = 259.965$, was .01 less than any value obtained in plotting the function in Example 2.1. As a result, it is reasonably safe to conclude that the four different, final locations for the new facility, given in Table 3.2, all are global minimum points.

Notice that Procedure 1, in this example, achieves the optimum in at most two iterations, and when the new facility is located initially at (50,50), only one iteration is required. From this example there appears to be no advantage in selecting the existing facilities as starting points since the optimum is achieved in no fewer iterations.

When Procedure 1 is applied to the problem given in Example 2.2, using the same starting locations for the new facility, similar paths to the local minimum points are obtained. Table 3.3 lists for each starting point the movement of the new facility location at each iteration of Procedure 1. The four global minimum points obtained are the points (50,40.3), (40.3,50), (59.7,50), and (59.7,50). The expected total distance traveled at all of these points is $z^* = 284.868$, and is 0.002

Table 3.2

Effect of Using Different Starting Points for Example 2.1

Starting Point No.	Path to the Optimum		
1	(0,0)	to (31.9,50)	to (43.4,50)
2	(100,0)	to (62.8,49.1)	to (56.6,50)
3	(100,100)	to (50,68.1)	to (50,56.6)
4	(0,100)	to (31.9,50)	to (43.4,50)
5	(30,20)	to (49.1,37.2)	to (50,43.4)
6	(50,0)	to (50,31.9)	to (50,43.4)
7	(100,50)	to (68.1,50)	to (56.6,50)
8	(50,100)	to (50,68.1)	to (50,56.6)
9	(0,50)	to (31.9,50)	to (43.4,50.0)
10	(50,50)	to (50,56.6)	

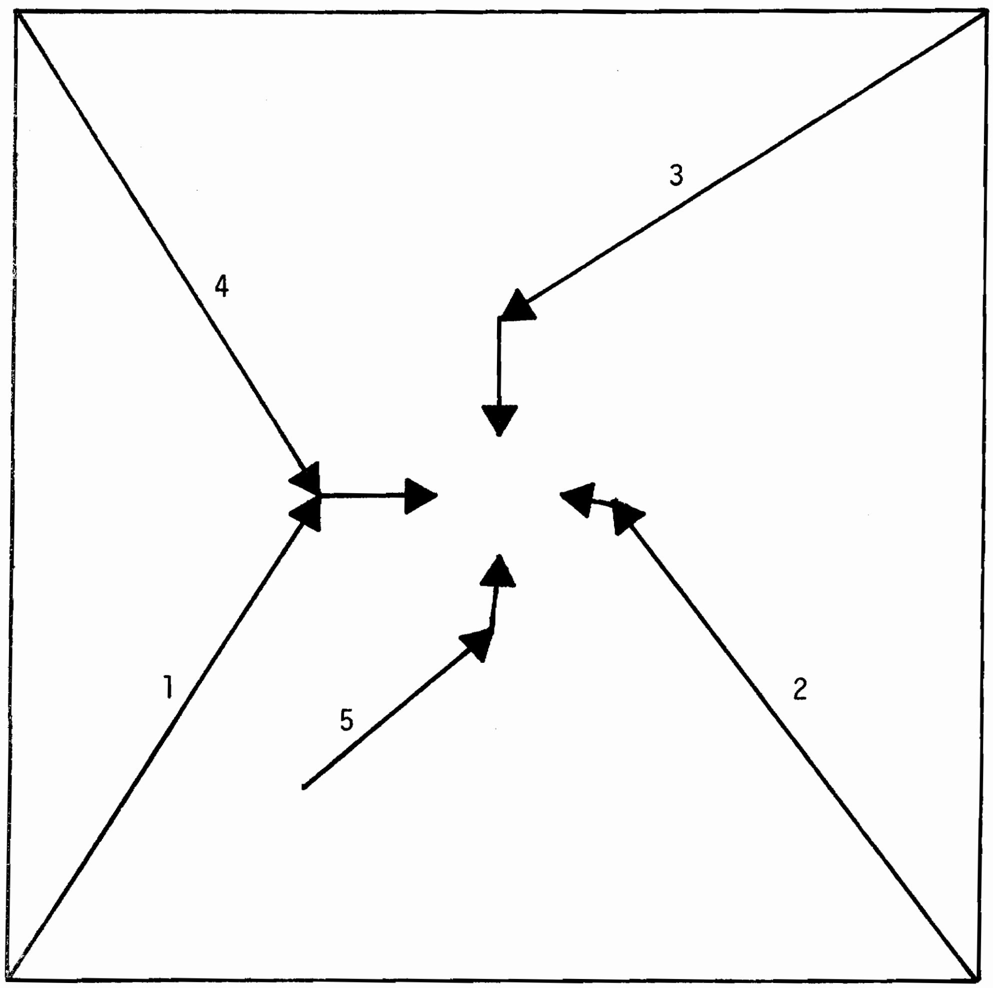

Figure 3.3

Paths to the Optimum Using Different
Starting Points in Procedure 1

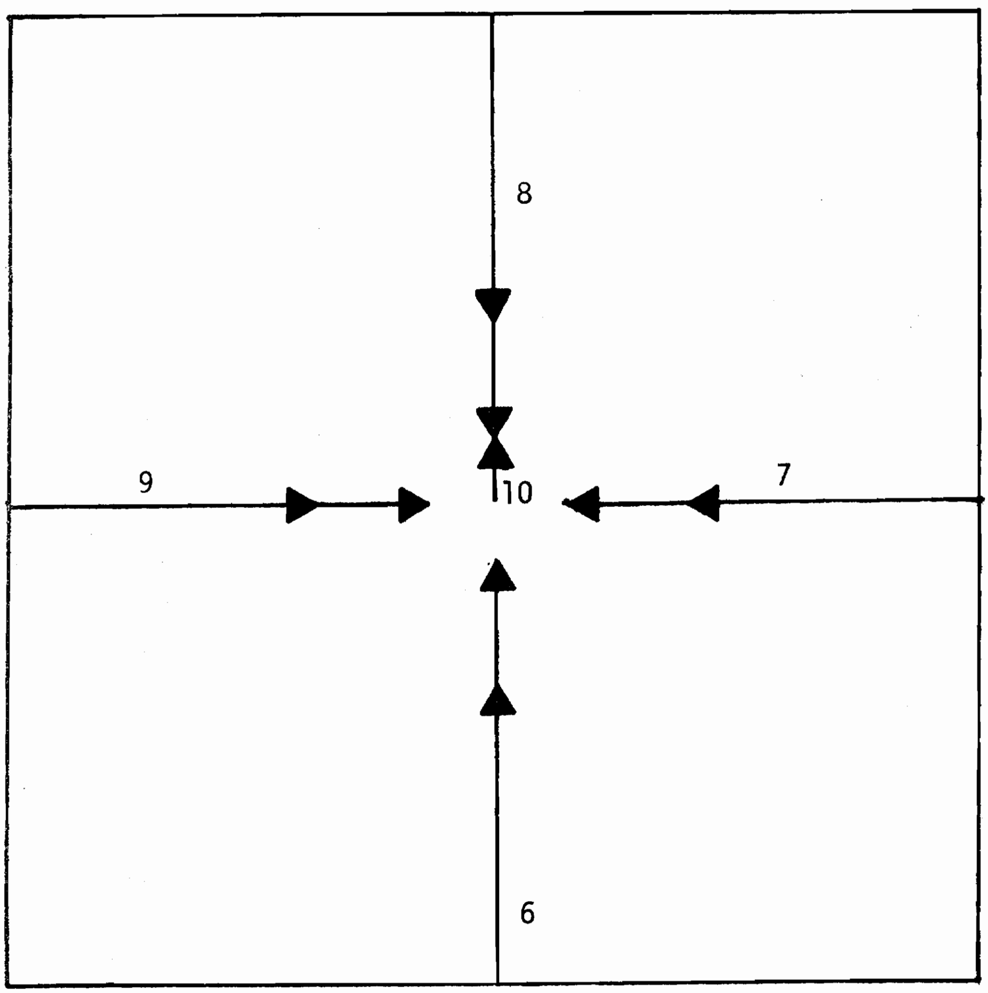

Figure 3.4

Paths to the Optimum Using Different Starting Points in Procedure 1

Table 3.3

Effect of Using Different Starting Points for Example 2.2

Starting Point No.	Path to the Optimum		
1	(0,0)	to (25,50)	to (40.3,50)
2	(100,0)	to (68.3,47.9)	to (59.7,50)
3	(100,100)	to (50,74.9)	to (50,59.7)
4	(0,100)	to (25,50)	to (40.3,50)
5	(30,20)	to (49,36.2)	to (50,40.3)
6	(50,0)	to (50,25)	to (50,40.3)
7	(100,50)	to (60.2,50)	to (59.7,50)
8	(50,100)	to (50,60.2)	to (50,59.7)
9	(0,50)	to (25,50)	to (40.3,50)
10	(50,50)	to (50,59.7)	

less than any objective function value obtained in plotting f(x,y) in Example 2.2. Notice in Table 3.3 that at most two iterations are required to obtain an optimal solution.

Recall, in Examples 2.3 and 2.4, that f(x,y) formed a pyramid such that the global maximum occurred at the point (50,50). For any starting point selected in the square region formed by the existing facilities, Procedure 1 locates the optimum in one iteration. This is to be expected since it was mentioned in Chapter 2 that when the rectilinear distance measure is used, only four traveling salesman zones result. Hence, any starting point selected must be in one of four traveling salesman zones, and when the new facility is relocated so as to minimize the distance traveled according to Equation (3.1), the new facility is "pulled" down the side of the pyramid to a point on the boundary of the square region.

The ability of Procedure 1 to obtain optimal solutions has been verified by applying the procedure to the four example problems in Chapter 2. Procedure 1 was also used to determine the minimum paths f(x,y) takes as $p_k \to 0$ for the different configurations given in Figure 2.9 through Figure 2.13. Procedure 1 is quite effective in solving traveling salesman location problems involving only a few existing facilities. However, as m becomes large, more traveling salesman zones are formed, and consequently, it is anticipated that the number of iterations required to find a local minimum will increase. Furthermore, for asymmetric configurations of existing facilities and unequal weighting of facilities due to the subset probability assignments, several different starting points should be used so that all local minima are explored. The procedure proposed in the next section may have an advantage for larger problems since it may require fewer functional evaluations in finding "improved" locations for the new facility.

Procedure 2: Functional Minimization using a Successive Quadratic Approximation Procedure

The search technique employed in Procedure 2 assumes that $f(x,y)$ is a convex function which can be closely approximated by a quadratic function, $\hat{f}(x,y)$. The approximation of $f(x,y)$ is accomplished through the use of regression analysis. The search technique, which will be called a search by regression, was developed by Schmidt and Taylor [18]. A complete description of the algorithm can be found in [18]. Essentially, the procedure is carried out by

1) Initially evaluating $f(x,y)$ at a number of points,
2) Fitting $f(x,y)$ with $\hat{f}(x,y)$ by the method of least squares,
3) Minimizing $\hat{f}(x,y)$ using classical optimization techniques,
4) Replacing the worst observation with the new minimum point,
5) Evaluating $f(x,y)$ at the new minimum point, and then
6) Successively fitting $f(x,y)$ with $\hat{f}(x,y)$, minimizing $\hat{f}(x,y)$, and replacing the (x,y) coordinate corresponding to the highest value contained in the observation vector with the new minimum point, continuing in this manner until some set of termination criteria are satisfied.

There are three factors of the search by regression that make it an appealing solution procedure for the traveling salesman location problem.

1) Only one functional evaluation is required per iteration. As m increases the computation time required for solving 2^m-1 traveling salesman problems becomes excessive. Therefore, for large problems it is desirable that the amount of time used per iteration in evaluating the objective function be minimized. In contrast, pattern search may involve as many as four perturbations about a point at each iteration.
2) The step size at each iteration is unlimited in magnitude and direction. This desirable feature is possessed by both of the solution procedures considered.
3) The iterative scheme need be carried out only once. Thus, after generating enough points to approximate f(x,y), and f(x,y) is sequentially minimized, it is not necessary to return to other initial starting points and repeat the entire procedure. Ideally, the search by regression will ignore relative minima and locate the new facility near the global minimum. It is not expected that the procedure will locate the new facility at the exact minimum point, however, since f(x,y) is non-convex.

The search by regression has several drawbacks in its application to the traveling salesman location problem.

1) The procedure is relatively ineffective in finding improved locations for the new facility when the rectilinear distance measure is used. Recall in Example 2.3, that f(x,y) turned out to be a concave function. When the search by regression is applied to Example 2.3, the new facility is continually

located at the maximum point (50,50). The search procedure is unable to detect that the function is concave instead of convex, and as a result, the stationary point selected is a relative maximum point. Procedure 2 can be modified slightly to handle the situation where f(x,y) is concave over a small part of the solution space, but even with the modification the search procedure cannot find a minimum point when f(x,y) is concave over the entire solution space.

2) In contrast to Procedure 1, an optimal location, if obtained, cannot be recognized since the search by regression does not utilize any of the information given by the traveling salesman zones. Only the objective function values obtained in previous iterations are retained for purposes of comparison.

The point should be stressed that Procedure 2 is most applicable as a means of solving the traveling salesman location problem when the number of existing facilities is so large that many functional evaluations cannot be performed because of the enormous computation time involved. In this case, the advantage of gaining more information regarding f(x,y) through many evaluations is outweighed by the high cost involved.

Presentation of the Algorithm

Let the approximating function, $\hat{f}(x,y)$, be the quadratic function given by

$$\hat{f}(x,y) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 y + \hat{\beta}_3 x^2 + \hat{\beta}_4 y^2 \ , \qquad (3.6)$$

where the $\hat{\beta}_j$'s, j = 1, 2, 3, 4, are coefficients determined by ordinary least squares. Cross product terms of the form xy, x/y, and y/x may be included in $\hat{f}(x,y)$, if necessary. A derivation of the least squares estimates for a second order model containing two independent variables is given in Appendix A.

Procedure 2:

1) Determine the minimum and maximum x and y coordinate values of the m existing facilities, (a_j,b_j), j = 1, 2, 3, ..., m, and denote these values as L_a, L_b, U_a, U_b.
2) Randomly generate n, n $\geq$ 5, (x,y) coordinate values:

   $x_i = (U_a - L_a)r + L_a$, i = 1, 2, 3, ..., n

   $y_i = (U_b - L_b)r + L_b$, i = 1, 2, 3, ..., n,

   where r is a uniformly distributed random number.
3) Evaluate $z_i = f(x_i,y_i)$, i = 1, 2, 3, ..., n. Set ℓ, the iteration number, equal to one.
4) Denote the (x_i,y_i) coordinate with the greatest z_i value by (x_1^{ℓ},y_1^{ℓ}). This point is to be replaced with a new point in step 12.
5) Compute the values of the A matrix (see Appendix A) and the $\underset{\sim}{c}$ vector, and then determine $\underset{\sim}{\hat{\beta}} = A^{-1}\underset{\sim}{c}$.
6) Set the first partial derivatives of $\hat{f}(x,y)$ with respect to x and y equal to zero, and solve for the minimum point for $\hat{f}(x,y)$: $(x_0^{\ell},y_0^{\ell}) = (-\hat{\beta}_1/2\hat{\beta}_3, -\hat{\beta}_2/2\hat{\beta}_4)$.

7) Constrain (x_0^{ℓ}, y_0^{ℓ}) to be in the solution space, i.e.,

if $x_0^{\ell} < L_a$, set $x_0^{\ell} = L_a$,

if $x_0^{\ell} > U_a$, set $x_0^{\ell} = U_a$,

if $y_0^{\ell} < L_b$, set $y_0^{\ell} = L_b$,

and if $y_0^{\ell} > U_b$, set $y_0^{\ell} = U_b$.

8) Evaluate $z_0^{\ell} = f(x_0^{\ell}, y_0^{\ell})$, and store z_0^{ℓ} and (x_0^{ℓ}, y_0^{ℓ}).

9) If ℓ is equal to the maximum number of iterations allowed, go to 13. Otherwise, go to 10.

10) If $\ell > 1$, check the termination criteria:
If successive objective function values differ by less than a predetermined value, δ, i.e., if $|z_0^{\ell} - z_0^{\ell-1}| < \delta$, go to 11. Otherwise, go to 12.

11) Before terminating make certain that a relative maximum point has not been obtained by performing the following test:
If z_0^{ℓ}, the stationary point for the current regression surface, $f(x_0^{\ell}, y_0^{\ell})$, differs from $z_0^{\ell-1}$ by the specified δ, and $z_0^{\ell-1}$ is the worst observation determined in step 12, then $f(x,y)$ must be concave in the region given by the vector of observations. Rather than terminate at a relative maximum point, move away from the point (x_0^{ℓ}, y_0^{ℓ}), and evaluate the objective function at the new point. Hopefully, the random move away from a maximum point will place the new current observation, $f(x_0^{\ell+1}, y_0^{\ell+1})$, in a region where f is convex. Such random steps should be made only several times during the entire

search, however. If the random move has been made, and the function has been evaluated at the new point, $(x_0^{\ell+1}, y_0^{\ell+1})$, replace (x_0^{ℓ}, y_0^{ℓ}) with $(x_0^{\ell+1}, y_0^{\ell+1})$, and go to 12. Otherwise, the termination criteria has been met. Therefore, go to 13.

12) Replace (x_1^{ℓ}, y_1^{ℓ}), the worst observation, with (x_0^{ℓ}, y_0^{ℓ}), the minimum point on the approximating surface. Set $\ell = \ell + 1$, and go to 4.

13) Stop. Write out the lowest z_0 obtained and its coordinate position, (x_0, y_0).

The check made for a relative maximum given by step 11 allows the procedure to work effectively on functions that are only approximately convex. This is best seen by applying Procedure 2 to Examples 2.1 and 2.2.

Example 3.2 Consider again the traveling salesman location problem of Example 2.1. Applying Procedure 2 to this problem, using as the stopping criterion a δ of 10^{-5}, the minimum point is obtained at the eleventh iteration. The total number of iterations is 11 also. It should be recognized that the minimum point is not necessarily always found on the last iteration. The minimum value obtained in the search is $f(x,y) = 259.967$ at $(x,y) = (43.0, 50.2)$. The minimum value of $f(x,y)$ here is only .002 greater than the minimum value obtained using Procedure 1. During the first few iterations of the search the minimum points of the regression surfaces occur near the known relative maximum point, (50,50), but the test for a relative maximum moves the predicted minimum point away from the point (50,50), thus allowing the procedure to find a point very close to the true minimum distance location.

When Procedure 2 is applied to Example 2.2 using $\delta = 10^{-5}$, the minimum value obtained is f(x,y) = 284.872 with (x,y) = (40.9,49.5). The minimum value is only .004 greater than the best value obtained using Procedure 1. The minimum value is obtained on the twelfth iteration, but because the termination criteria, δ, is so small, the search continues for a total of 30 iterations. By increasing δ from 10^{-5} to 5×10^{-2}, the search terminates at iteration number 5, with f(x,y) = 285.043 at (x,y) = (43.4,46.2). Clearly, there is a trade off involved between the cost savings resulting from finding the optimal location for the new facility and the cost of computer time incurred in finding the optimal location. This trade off should be kept in mind whenever the total, overall objective is profit maximization or cost minimization.

Summary

Chapter 3 has proposed two heuristic solution procedures for the traveling salesman location problem. By applying each procedure to several of the example problems contained in Chapter 2 it was verified that both procedures are capable of obtaining optimal solutions. Procedure 1 has two advantages over Procedure 2. First, Procedure 1 is capable of recognizing an optimal solution as soon as it is obtained, whereas Procedure 2 cannot recognize a solution as being optimal. Second, Procedure 1 works well for either Euclidean or rectilinear distance measures. Procedure 2 appears to be ineffective when applied to rectilinear distance problems. An appealing feature of Procedure 2 is that it minimizes the number of functional evaluations required. As a result, Procedure 2 may be preferred to Procedure 1 for large problems since as the number of existing facilities increases, the time required for one functional evaluation becomes quite lengthy.

Chapter 4

COMPUTATIONAL EXPERIENCE

Introduction

In the preceding chapter of this thesis two heuristic procedures for solving the traveling salesman location problem were proposed. The purpose of this chapter is to compare the two procedures by applying them to a number of problems of varying size. The bases for comparison of the procedures will be:

1) The effectiveness of each procedure in minimizing the objective function, and
2) The amount of time required for each procedure to obtain an "optimum" solution to the traveling salesman location problem.

After a discussion of problem generation and the presentation of computational results, inferences will be drawn regarding the performance of the two procedures.

Problem Generation

In order to examine the effectiveness of each procedure under a variety of conditions a large number of problems were randomly generated. The solution space was constrained to the square region of Examples 2.1, 2.2, 2.3, and 2.4, i.e., the existing facility locations were randomly generated according to the following probability density functions:

$$f(a_i) = .01 \quad , 0 < a_i < 100, \ i = 1, 2, 3, \ldots, m,$$

$$f(b_i) = .01 \quad , 0 < b_i < 100, \ i = 1, 2, 3, \ldots, m.$$

The existing facilities were constrained to be within the square region in order to facilitate the comparison of the procedures. The numerical values obtained for f(x,y) would be in the same range as the objective function values obtained in the example problems. The ratios of the maximum values to the minimum values in Examples 2.1 - 2.4 were 1.08, 1.06, 1.02, and 1.03, respectively. These ratios were low due to the manner in which the subset probabilities were formed. In generating the subset probabilities for some of the randomly generated problems a number of the subset probabilities were forced to be relatively large with respect to all other subset probabilities. This was done to prevent the loss of generality that would result by examining only problems whose objective functions were "smoothed" out due to equal subset probabilities. Hence, it was intended that all types of general problems be evaluated. All problems generated had asymmetric configurations of existing facilities; some had relatively identical subset probabilities, and some had subset probability assignments favoring particular existing facilities.

Each procedure was applied to the same randomly generated problems. The size of the problems examined was limited to $m \leq 11$ due to excessive storage requirements and lengthy execution times for problems involving more than 11 existing facilities. For the same reasons, the number of problems generated and solved decreased as m was increased. Specifically, 50 problems were generated and solved for m = 4, 40 problems for m = 6, 30 problems for m = 8, 15 problems for m = 10, and 1 problem was solved for m = 11.

Computational Results

The effectiveness of each procedure was determined by comparing the minimum values obtained for f(x,y), and by comparing the execution times required to achieve the minimum values. The computational results dealing with the relative accuracies of the procedures will first be presented, after which the results relating to the execution times for the procedures will be given, followed by a discussion of the merits of each basis for comparison.

Since the global minimum points for the randomly generated problems were not known, the two procedures were compared by examining the ratios of the minimum values obtained for each problem. For any given problem the ratio was set up so that the ratio was greater than one. If Procedure 1 yielded a solution with objective function value less than the objective function value obtained using Procedure 2, then the ratio used was $R_1 = f_2/f_1$. If Procedure 2 yielded the lower objective function value, then the ratio formed was $R_2 = f_1/f_2$. Hence, the subscript of R denotes which of the two procedures located the new facility closer to the global minimum of f(x,y), and the value of (1-R) indicates the percent improvement in the minimum value obtained resulting from using the preferred procedure.

Table 4.1 lists the overall performance of each procedure based on the number of times Procedure 1 yielded better solutions than Procedure 2. Considering only the first method for comparing the procedures and using the percentages given in Table 4.1, it would appear that Procedure 1 is overwhelmingly more effective than Procedure 2. However, it must be determined how much more effective Procedure 1 is. To this end, cumulative frequency distributions are given in Tables 4.2 - 4.5 for each set of problems generated. The distance measure used for all problems was Euclidean. Tables 4.1 and 4.2, together, indicate that even though Procedure 1 yielded functional values lower than Procedure 2 98 percent of the time, 77.5 percent of those functional values differed from the values obtained using Procedure 2 by a factor of no more than .0076. This factor is hardly significant even when considering the low maximum/minimum ratios of Examples 2.1 - 2.4. In the one instance when Procedure 2 achieved a lower value, the percentage difference, .01 percent, was even less significant. Less than 3 percent, (1-.979), of the problems solved using Procedure 1 differed from Procedure 2 by more than 2.28 percent. The greatest difference in minimum values obtained was 3.8 percent. Thus, it must be concluded that for m = 4, there is no significant difference between Procedures 1 and 2.

In Example 2.1 the percentage difference between the global maximum and the global minimum values of the objective function was only 8 percent. In Example 2.2 the difference was only 6 percent. In view of these small differences it seems reasonable to assume that a difference of 2 to 3 percent between solution values obtained using Procedures 1 and 2 can be considered to be significant. Hence, if a large enough percentage of the problems solved show one procedure to be significantly better than the other, then a decision can be made as to which procedure is better.

Table 4.3 gives the cumulative frequency distributions of R_1 and R_2 for the 40 problems solved involving six existing facilities. Procedure 1 yielded a lower cost solution for 87.5 percent of the problems solved.

Table 4.1

Overall Performance - Procedure 1 vs. Procedure 2

m	Number of Problems	Percentage of R_1's	Percentage of R_2's
4	50	98	2
6	40	87.5	12.5
8	30	90	10
10	15	86.67	13.33
11	1	100	0

Table 4.2

Cumulative Frequency Distributions
for Problems Involving Four Existing Facilities

R_1	$F(R_1)$	R_2	$F(R_2)$
1.0076	.775	1.0001	1.0000
1.0152	.878		
1.0228	.979		
1.0304	.979		
1.0380	1.000		

Table 4.3

Cumulative Frequency Distributions
for Problems Involving Six Existing Facilities

R_1	$F(R_1)$	R_2	$F(R_2)$
1.0104	.571	1.0005	.200
1.0208	.771	1.0006	.600
1.0313	.857	1.0009	.800
1.0417	.971	1.0148	1.000
1.0521	1.000		

Table 4.4

Cumulative Frequency Distributions
for Problems Involving Eight Existing Facilities

R_1	$F(R_1)$	R_2	$F(R_2)$
1.0123	.444	1.0024	.667
1.0246	.741	1.0101	1.000
1.0369	.852		
1.0492	.926		
1.0614	1.000		

Table 4.5

Cumulative Frequency Distributions
for Problems Involving Ten Existing Facilities

R_1	$F(R_1)$	R_2	$F(R_2)$
1.0088	.615	1.0074	.500
1.0176	.769	1.0144	1.000
1.0262	.846		
1.0350	.923		
1.0429	1.000		

About 23 percent, i.e., (1-.771), of the problems having ratios R_1 differed from Procedure 2 solutions by a factor of .0208. In other words, about 20 percent, or (87.5%)(23%), of all problems solved had ratios favoring Procedure 1 by a factor of more than 2 percent, (.0208). Notice in Table 4.3 that of the five problems solved favoring Procedure 2, all of the ratios were less than 1.02.

Tables 4.4 and 4.5 present the cumulative frequency distributions of R_1 and R_2 for the problems involving eight existing facilities, and 10 existing facilities, respectively. Only one problem involving 11 existing facilities was solved.

Based on the assumption that a ratio starting somewhere between 1.02 and 1.03 indicates significant improvement of one procedure over another, the results obtained thus far can be summarized as follows:

1) There is no significant difference between Procedure 1 and Procedure 2 for $m = 4$.
2) For $m = 6$, 20 percent of all problems solved had ratios favoring Procedure 1 at a significant level of $R_1 \geq 1.0228$.
3) For $m = 8$, 23 percent of all problems solved had ratios favoring Procedure 1 at a significant level of $R_1 \geq 1.0246$.
4) For $m = 10$, 13 percent of all problems solved had ratios favoring Procedure 1 at a significant level of $R_1 \geq 1.0262$.
5) For $m = 11$, the ratio obtained was $R_1 = 1.0211$.

It was suggested in Chapter 2 that Procedure 2 would probably not be very effective if rectilinear distances were used. Procedure 2 was tested against Procedure 1 using rectilinear distances for $m = 4$

and m = 6. More than 80 percent of the problems solved in both cases had ratios of R_1, $1.08 < R_1 < 1.12$. Furthermore, there were no R_2 ratios formed. At this point it was concluded that Procedure 2 is totally ineffective when rectilinear distances are used. Apparently, f(x,y) is more concave than it is convex even when configurations of existing facilities and subset probabilities are randomized.

It is interesting to note in passing that the resulting ratios of minimum rectilinear distance solutions to minimum Euclidean distance solutions obtained by Procedure 1 was always greater than one. This result should be intuitive. The ratios ranged from a minimum of 1.0925 to a maximum of 1.4692, with the overwhelming majority of the ratios being about 1.3. This result is useful for real world applications of the traveling salesman location problem, since it indicates that selection of the appropriate distance measure is rather important.

A discussion of the shortcomings of the first measure of effectiveness will be postponed until the results regarding the second measure of effectiveness are presented. Recall that the second measure of effectiveness was based on execution times required to achieve a minimum solution for each procedure. The average execution times, $\overline{T}$, and the standard deviations of execution times, σ_T, for Procedures 1 and 2 are given in Table 4.6. The computer programs for Procedures 1 and 2 were written in Fortran IV and were run on an IBM 370/Model 158 computer and are presented in Appendix B. There are no claims made regarding the efficiency of the programs.

Comparisons of the effectiveness of Procedures 1 and 2 on the basis of mean execution times must be carried out with great caution. The

Table 4.6

Execution Time Data for Euclidean Distance Measure - Procedure 1 and Procedure 2

m	No. of Problems	Procedure 1		Procedure 2	
		Mean Execution Time $\bar{T}$ (sec.)	Std. Deviation of T	Mean Execution Time $\bar{T}$ (sec.)	Std. Deviation of T
4	50	2.9	.9	3.5	1.5
6	40	15.5	6.5	39.3	15.0
8	30	90.3	32.8	102.4	40.8
10	15	376.5	90.3	304.1	82.1
11	1	792.4	-	604.3	-

execution time data for the two procedures is highly unreliable for two reasons:

1) Due to the time-sharing nature of the operation of the computer facility at Virginia Polytechnic Institute and State University, execution times cannot be accurately measured. If one program solving the same problem is submitted twice, the resulting execution times may be quite different.
2) There is certain to be a difference in the degree of efficiency between the computer program written for Procedure 1 and the computer program written for Procedure 2.

Judgements regarding the effectiveness of Procedures 1 and 2 based on relative differences in execution times is therefore reserved only for large relative differences.

Returning now to the mean execution time data given in Table 4.6, it is seen that for m = 4, the mean execution times required for each procedure were approximately the same. Each functional evaluation requires the solution of only four non-trivial traveling salesman problems. Consequently, the execution times are quite low. Procedure 2 probably required a longer execution time due to the fact that eight functional evaluations were required before beginning the regression procedure.

The difference between the mean execution times for m = 6 is significant. A possible explanation for the difference is that Procedure 1 required fewer evaluations of f(x,y) than did Procedure 2. Procedure 1 usually required about six iterations, i.e., seven evaluations of f(x,y), whereas Procedure 2 required a minimum of 10 evaluations, an initial eight plus two iterations, and usually required about 15 evaluations.

Notice that as m is increased the mean execution time for Procedure 1 eventually becomes larger than the mean execution times required for Procedure 2. The most probable cause for the differences is the greater number of functional evaluations required for Procedure 1 to find a local minimum. For m = 10, the differences in mean execution times may or may not be significant. It is felt that the differences are significant due to the different upward trends in execution times. For m = 11, the differences are certainly significant. Unfortunately, however, the great reduction in execution time resulting from the use of Procedure 2 occurs at a point where the solution of larger problems becomes infeasible. The solution of larger problems becomes infeasible due to excessive storage requirements and lengthy execution times. The time required to solve a problem can probably be reduced significantly by employing one of the heuristic solution procedures for solving the traveling salesman problem, however.

Table 4.7 lists the mean execution times required for solving each set of problems for each distance measure. The execution times required for solving the traveling salesman location problem using rectilinear distances and using Euclidean distances were about the same. This result was anticipated since the majority of execution time is used in evaluating f(x,y), and there is no reason to believe that the traveling salesman problems involving rectilinear distances are solved faster than traveling salesman problems involving Euclidean distances.

Table 4.7

Execution Time Data for Euclidean and Rectilinear Distance Measures - Procedure 1

Distance Measure	m	No. of Problems	Mean Execution Time $\bar{T}$ (sec.)	Std. Deviation of T
Euclidean	4	50	2.9	.9
Rectilinear			4.6	4.4
Euclidean	6	40	15.5	6.5
Rectilinear			11.9	2.5
Euclidean	8	30	90.3	32.8
Rectilinear			86.5	27.7
Euclidean	10	15	376.5	90.3
Rectilinear			352.7	95.7

Discussion

A few qualifications must be added before any final conclusions can be made regarding the effectiveness of Procedures 1 and 2. Each procedure might be improved in several ways. For example, experimentation with different values of δ, the stopping criteria for Procedure 2, might lead to reduced execution times for Procedure 2. Increasing the value of δ tends to decrease execution time, but increase the value of the minimum solution. Small values of δ tend to make the procedure terminate on the maximum number of allowable iterations. The values of δ that were used in obtaining the computational results were selected not to bias the results, but rather to prevent execution times from being excessive. For $m = 4$ and $m = 6$, a value of $\delta = 5 \times 10^{-2}$ was used. For $m = 8$, $m = 10$, and $m = 11$, a value of $\delta = 10^{-1}$ was used.

A similar refinement can be made with Procedure 1. The Hyperbolic Approximation Procedure begins by setting the hyperboloid constant equal to 10^{-2}, and the value is then sequentially reduced at each approximation until the optimal solution to the Steiner-Weber problem is obtained. Rather than go through many approximations to get the optimal solution to the Steiner-Weber problem, an improved strategy would be to go through one or two approximations, and then check to see if the new facility has been moved to a new traveling salesman zone. Most of the movement of the new facility takes place during the first few approximations. If the new facility has moved to a new traveling salesman zone in the first few approximations, further approximations are just wasted execution time.

Another manner in which the execution time and the minimum value obtained can be affected is by altering the number of starting points to be used in Procedures 1 and 2. The minimum number of starting points required by Procedure 2 is five. When only five starting points were used for Procedure 2, it was found that the matrix A, used in the regression procedure, often became close enough to being singular that the search procedure was abnormally terminated. It was found by experimentation that the best balance between execution time and near optimal solutions is achieved by using eight starting points.

For m = 4, five initial starting points were randomly generated for each of the 50 problems solved using Procedure 1. The resulting execution time data were $\overline{T} = 4.6$ seconds and $\sigma_T = 1.0$. The same sets of problems were then solved using three randomly generated starting positions. The resulting execution time data were as given in Table 4.6, $\overline{T} = 2.9$ seconds and $\sigma_T = .9$. Using three random starting positions, the same minimum values for f(x,y) were obtained for all 50 problems. Hence, it was observed that increasing the number of starting points adversely affects execution times while doing little to improve the minimum values obtained. Additional starting points should not be used unless they can be placed strategically. Since little was known regarding the randomly generated problems, only one starting point was used for the problems solved for m = 6, m = 8, m = 10, and m = 11. The starting point generated for each problem was constrained to be within the smallest square containing all the existing facilities.

It is interesting to note that in using Procedure 1 local minimum points were found in all problems generated. However, there appears to exist a mathematical possibility that Procedure 1 may lead to cycling difficulties. Figure 4.1 depicts the conditions under which cycling would occur. Suppose the new facility is located in Zone 1 at the point (x_1, c), and the weights formed at this point place the new facility on g_1, the function that is to be minimized using HAP. Suppose that in minimizing g_1, the new facility is moved into Zone 2 to the point (x_2, c). If the weights formed at (x_2, c) place the new facility on g_2, then Procedure 1 will cycle indefinitely between Zones 1 and 2. Notice that the minimum distance solution in this case is at the point of intersection of g_2 and the boundary line for Zones 1 and 2. In all the problems solved to date, no cycling difficulty has been encountered.

Conclusions

No absolute judgement can be made regarding the effectiveness of the two solution procedures. On the basis of the computational results obtained, three rather general statements can be made:

1) Procedure 1 obtains better solutions than Procedure 2 about 90 percent of the time.
2) About 20 percent of the time Procedure 1 locates the new facility such that the resulting expected distance traveled is at least two percent less than the solution given by Procedure 2.
3) As m becomes large the computation time becomes excessive for both procedures, especially Procedure 1.

Chapter 5 summarizes the research effort contained in this thesis and suggests some areas for further research.

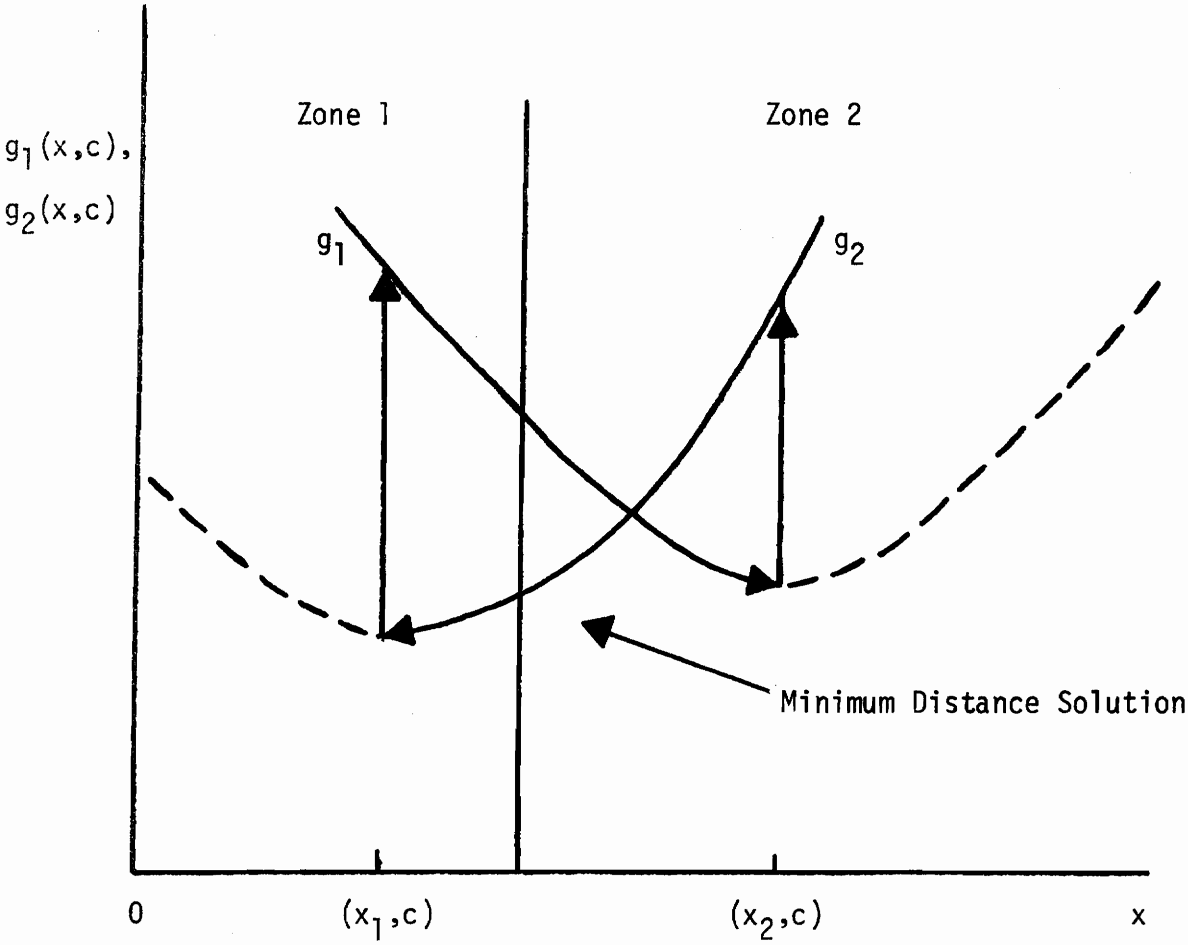

Figure 4.1

An Illustration of Cycling Between Traveling Salesman Zones

Chapter 5

SUMMARY AND RECOMMENDATIONS FOR FURTHER RESEARCH

Summary

The objectives of this research were to present an original formulation of a significant facilities location problem, the traveling salesman location problem, and to develop several heuristic solution procedures for determining minimum distance locations. Despite the wide applicability of the traveling salesman location problem, a survey of the facilities location literature revealed that this research effort apparently was the first to address the problem.

After mathematically formulating the problem, several rather simple example problems were investigated in order to gain some insight regarding the behavior of the function under a variety of different conditions. Many of the results stemming from the study of the simple examples were counter-intuitive. Additionally, it was demonstrated that even for problems involving only a few existing facilities the resulting objective function is non-convex.

Due to the non-convexity of the objective function and the overwhelming combinatorics involved with just one functional evaluation, it was desired that the solution procedures developed be capable of obtaining near optimal solutions in the shortest time possible. One of the solution techniques proposed, Procedure 2, was based on the Successive Quadratic Approximation Procedure. This procedure was selected for two reasons:

1) It was expected that the procedure would yield minimum solutions to large problems rather quickly, and

2) It was hoped that by approximating the function over the entire solution space, the procedure would tend to overlook local minimum points, and instead, find a global minimum point.

It was demonstrated that while Procedure 2 is capable of obtaining optimal solutions, it does not immediately recognize a particular solution as being optimal. The other procedure proposed, Procedure 1, based on a relationship between the Steiner-Weber problem and the traveling salesman location problem, was selected because of its ability to immediately recognize a particular solution as being a local minimum point. At each iteration Procedure 1 required the solution of a Steiner-Weber problem as well as solutions to the "string" of traveling salesman problems. The Steiner-Weber problems were solved through the use of the Hyperbolic Approximation Procedure. It was verified that both procedures are capable of obtaining optimal solutions by applying each procedure to several of the example problems.

The effectiveness of each procedure in finding minimum distance solutions was determined by applying each procedure to a number of randomly generated problems, and then comparing the resulting execution times and minimum distance solutions. A difference of two percent or more between the minimum distance solutions obtained for a given problem was considered to be significant. Problems involving 4, 6, 8, 10, and 11 existing facilities were solved. No attempts were made to solve larger problems due to the excessively long execution times required.

On the basis of the computational results obtained, it was concluded that

1) There is no significant difference between Procedures 1 and 2 for problems involving four existing facilities.
2) For about 15 to 20 percent of the problems involving 6, 8, 10, or 11 existing facilities, Procedure 1 performs better than Procedure 2.

By examining the mean execution times for each procedure, it was found that there was little significant difference between the two procedures until rather large problems were solved. Procedure 2 required relatively shorter execution times than Procedure 1 for problems involving 10 or 11 existing facilities. However, the reduction in execution time for Procedure 2 occurred at a point where it was considered economically infeasible to continue to examine larger problems. The length of execution times for larger problems can probably be reduced by:

1) Eliminating the need to evaluate all traveling salesman problems by setting many of the p_h's, the subset probabilities, equal to zero, and
2) Replacing the branch and bound algorithm for solving traveling salesman problems with one of the more effective heuristic procedures that have been developed.

Recommendations for Further Research

The results of this research suggest two major areas for extended research:

1) Concentration on the development of more efficient techniques for solving the traveling salesman location problem, and
2) Extending the problem formulation to include multiple new facilities.

One method of reducing the time required to evaluate the objective function has already been mentioned: apply some of the existing heuristic methods for solving the traveling salesman problem. Ideally, one would like to be able to evaluate all 2^m-1 traveling salesman problems at once. However, a great deal more must be learned about the traveling salesman problem itself before such an approach can be taken.

Another possible approach would be to attempt to reduce the problem size by representing a number of existing facilities located in the same area by a "pseudo" existing facility. Figure 5.1 shows graphically how "pseudo" existing facilities might be located. The X's represent the existing facilities, and the circled X's denote the "pseudo" existing facilities. If the X's can be separated according to major traveling salesman zones, and a weighting factor can be given to the "pseudo" existing facilities, then the new problem to be solved is a much smaller one. The difficulty presented here is determining the manner in which

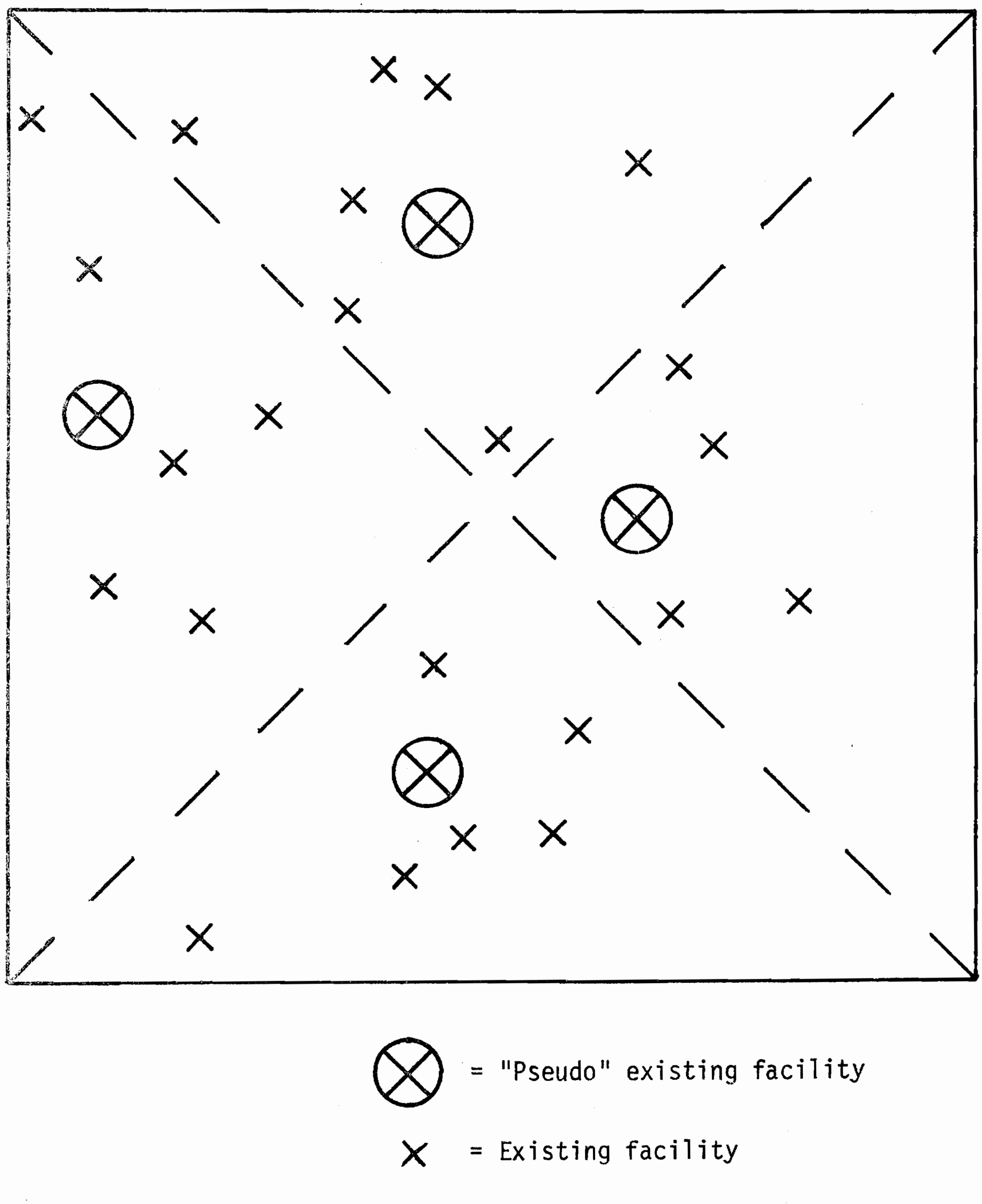

Figure 5.1

The Formation of "Pseudo" Existing Facilities
from Existing Facilities

the "pseudo" existing facilities are to be weighted. Clearly, these weights will be heavily dependent upon the subset probabilities and the distances between all the existing facilities in a zone.

Much of the success in the second major area for further research will be dependent upon the advances made in the first major area. However, there is one avenue of approach that may be taken using the procedures proposed in this thesis. Suppose that in the multiple traveling salesman location problem there are m existing facilities and n new facilities. The proposed method of solution would be to define (n-1) of the new facilities to be "pseudo" existing facilities. Holding the locations of the "pseudo" existing facilities fixed, the single facility traveling salesman location problem involving (m+n-1) existing facilities could be solved using either Procedure 1 or Procedure 2. After relocating the new facility, its location would be fixed, and one of the other "pseudo" existing facilities would be treated as a new facility. After solving n of these single new facility traveling salesman location problems, a second pass would be made to see if any of the n new facilities are moved. If none of the new facilities are moved on the second set of problems solved, then the solution might be optimal. If the cost of maintaining the operating new facilities is included in the analysis, the optimal number of new facilities needed to minimize total cost can also be determined.

One of the first efforts directed at studying further the traveling salesman location problem should be to establish more exact relationships between particular configurations of existing facilities, the number of existing facilities, and particular probability distributions of the subsets to be visited. After these relationships are better understood, the minimum distance locations for new facilities will be more easily obtained. A more thorough understanding of the behavior of the objective function under various conditions may lead to the development of better and faster solution methods for the traveling salesman location problem. Additionally, it may be possible to develop a procedure that will solve the traveling salesman location problem in a non-sequential fashion. Such a development would quite likely result in a great reduction in solution times. Consequently, much larger problems could be solved.

The traveling salesman location problem as formulated in this thesis assumes a continuous solution space. An examination of the problem using a discrete solution space might lead to new and more efficient solution procedures for the continuous case. The multiple new facility traveling salesman location problem might be solved using the Quadratic Assignment problem, just as the Hyperbolic Approximation Procedure can be used for a continuous solution space. Unfortunately, however, the existing techniques for solving the Quadratic Assignment problem are not very efficient.

The traveling salesman location problem as formulated assumes that there is only one salesman. An interesting extension would be to study the problem as an n traveling salesman location problem. In this case the objective would be to locate the new facility so as to minimize the expected distance traveled given n salesmen and 2^m-1 different subsets of facilities to be visited. The problem can be solved using the existing techniques with but one modification. The distance matrix must be augmented to solve the n traveling salesman problem. There is no reason to believe that for a given problem the minimum distance location for the new facility in the n salesman problem will coincide with the minimum distance location for the one salesman problem. Many of the larger itineraries may be broken up so that more than one salesman is involved in the optimal route. The addition of time and/or capacity constraints can be added to the n traveling salesman location problem as well as the single salesman problem. While these additions would make the problem more interesting, the procedures developed to solve such a problem would most certainly have to be heuristic procedures.

BIBLIOGRAPHY

1. Ashour, S., and Parker, R. G., "A Schedule Algebra Approach to the Traveling Salesman Problem," Unpublished working paper, Department of Industrial Engineering, Kansas State University, Kansas.

2. Bellman, R. E., "Dynamic Programming Treatment of the Traveling Salesman Problem," Journal of the Association for Computing Machinery, Vol. 9, No. 1, 1962.

3. Bellmore, M., and Malone, J. C., "Pathology of Traveling-Salesman Subtour-Elimination Algorithms," Operations Research, Vol. 19, No. 2, 1971.

4. Bozok, A. S., "A Review of Facility Location and Design Research," M.E. Report, Virginia Polytechnic Institute and State University, 1972.

5. Eastman, W. L., "Linear Programming with Pattern Constraints," Ph.D. Thesis, Report No. BL. 20, The Computation Laboratory, Harvard University, 1958.

6. Eastman, W. L., "A Solution to the Traveling Salesman Problem," Presented at the American Summer Meeting of the American Econometric Society, Cambridge, Massachusetts, August 1958.

7. Eilon, S., Watson-Gandy, C. D. T., and Christofides, N., Distribution Management: Mathematical Modelling and Practical Analysis, Griffin, London, 1971.

8. Eyster, J. W., White, J. A., and Wierwille, W. W., "On Solving Multifacility Location Problems Using a Hyperboloid Approximation Procedure," AIIE Transactions, Vol. 5, No. , 1973.

9. Francis, R. L., and Goldstein, J. M., "Location Theory: A Selective Bibliography," Operations Research, Vol. 22, No. 22, 1974.

10. Francis, R. L., and White, J. A., Facility Layout and Location: An Analytical Approach, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

11. Lawler, E. L., and Wood, D. E., "Branch and Bound Methods: A Survey," Operations Research, Vol. 14, No. 4, 1966.

12. Lea, A. C., "Location-Allocation Systems - An Annotated Bibliography," Discussion Paper No. 13, Department of Geography, University of Toronto, May 1973.

13. Lin, S., "Computer Solutions of the Traveling-Salesman Problem," The Bell System Technical Journal, Vol. 44, No. 10, 1965.

14. Lin, S., and Kirnighan, B. W., "An Effective Algorithm for the Traveling-Salesman Problem," Operations Research, Vol. 21, No. 2, 1973.

15. Little, J. D., Murty, K. G., Sweeney, D. W., and Karel, C., "An Algorithm for the Traveling-Salesman Problem," Operations Research, Vol. 11, No. 6, 1963.

16. Lohmar, W., "Comprehensive Examination for the Ph.D. Degree in Industrial Engineering and Operations Research," Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1972.

17. Miller, C., Tucker, A., and Zemlin, R., "Integer Programming Formulation of Traveling Salesman Problems," Journal of the Association for Computing Machinery, Vol. 7, No. 4, 1960.

18. Schmidt, J. W., and Taylor, R. E., Simulation and Analysis of Industrial Systems, Richard D. Irwin, Inc., Homewood, Illinois, 1970.

19. Turner, W. C., Ghare, P. M., and Foulds, L. R., "Transportation Routing Problem - A Survey," Unpublished paper, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1973.

20. Wagner, H. M., Principles of Operations Research, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.

21. Wells, M. B., Elements of Combinatorial Computing, Pergamon Press, New York, 1971.

APPENDIX A

Derivation of the Least Squares Estimates for a Second Order Model Involving Two Independent Variables

The method by which the least squares estimates are obtained for use in Procedure 2 will now be given.

Assume that $z = f(x,y)$ is a non-convex function that is well approximated by a quadratic function:

$$\hat{f}(x,y) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 \ . \qquad \text{(A.1)}$$

By obtaining five or more paired observations, (x_i, y_i), z_i, $i = 1, 2, 3, 4, 5, \ldots$, the objective function, $f(x,y)$, can be fitted by the method of least squares. The least squares estimates are found by minimizing

$$\sum_{i=1}^{k} [z_i - \hat{f}(x_i, y_i)]^2 = \sum_{i=1}^{k} [z_i - \beta_0 - \beta_1 x_i - \beta_2 y_i - \beta_3 x_i^2 - \beta_4 y_i^2]^2 , \qquad \text{(A.2)}$$

where k, $k \geq 5$, is the total number of observations. Setting the first partial derivatives with respect to β_0, β_1, β_2, β_3, and β_4 equal to zero, the following system of equations is obtained:

$$\sum_i z_i = k\hat{\beta}_0 + \hat{\beta}_1 \sum_i x_i + \hat{\beta}_2 \sum_i y_i + \hat{\beta}_3 \sum_i x_i^2 + \hat{\beta}_4 \sum_i y_i^2$$

$$\sum_i x_i z_i = \hat{\beta}_0 \sum_i x_i + \hat{\beta}_1 \sum_i x_i^2 + \hat{\beta}_2 \sum_i x_i y_i + \hat{\beta}_3 \sum_i x_i^3 + \hat{\beta}_4 \sum_i x_i y_i^2$$

$$\sum_i y_i z_i = \hat{\beta}_0 \sum_i y_i + \hat{\beta}_1 \sum_i x_i y_i + \hat{\beta}_2 \sum_i y_i^2 + \hat{\beta}_3 \sum_i x_i^2 y_i + \hat{\beta}_4 \sum_i y_i^3$$

$$\sum_i x_i^2 z_i = \hat{\beta}_0 \sum_i x_i^2 + \hat{\beta}_1 \sum_i x_i^3 + \hat{\beta}_2 \sum_i x_i^2 y_i + \hat{\beta}_3 \sum_i x_i^4 + \hat{\beta}_4 \sum_i x_i^2 y_i^2$$

$$\sum_i y_i^2 z_i = \hat{\beta}_0 \sum_i y_i^2 + \hat{\beta}_1 \sum_i x_i y_i^2 + \hat{\beta}_2 \sum_i y_i^3 + \hat{\beta}_3 \sum_i x_i^2 y_i^2 + \hat{\beta}_4 \sum_i y_i^4 \ .$$

The above system of equations can be expressed in matrix notation as

$$A \underset{\sim}{\hat{\beta}} = \underset{\sim}{c} \ , \qquad (A.3)$$

where

$$A = \begin{bmatrix} k & A_1 & A_2 & A_3 & A_4 \\ A_1 & A_3 & A_5 & A_6 & A_7 \\ A_2 & A_5 & A_4 & A_8 & A_9 \\ A_3 & A_6 & A_8 & A_{10} & A_{11} \\ A_4 & A_7 & A_9 & A_{11} & A_{12} \end{bmatrix}$$

$A_1 = \sum_i x_i$ $\qquad A_5 = \sum_i x_i y_i$ $\qquad A_9 = \sum_i y_i^3$

$A_2 = \sum_i y_i$ $\qquad A_6 = \sum_i x_i^3$ $\qquad A_{10} = \sum_i x_i^4$

$A_3 = \sum_i x_i^2$ $\qquad A_7 = \sum_i x_i y_i^2$ $\qquad A_{11} = \sum_i x_i^2 y_i^2$

$A_4 = \sum_i y_i^2$ $\qquad A_8 = \sum_i x_i^2 y_i$ $\qquad A_{12} = \sum_i y_i^4$,

and

$$\underset{\sim}{\hat{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4)^T ,$$

and

$$\underset{\sim}{c} = (c_1, c_2, c_3, c_4, c_5)^T ,$$

$c_1 = \sum_i z_i$ $\qquad c_3 = \sum_i y_i z_i$ $\qquad c_5 = \sum_i y_i^2 z_i$.

$c_2 = \sum_i x_i z_i$ $\qquad c_4 = \sum_i x_i^2 z_i$

The vector of least squares estimates is given by

$$\hat{\underset{\sim}{\beta}} = A^{-1} \underset{\sim}{c} \; . \tag{A.4}$$

The next step in Procedure 2 is to find the minimum point on the regression surface:

$$(x^*,y^*) = (-\hat{\beta}_1/2\hat{\beta}_3, \; -\hat{\beta}_2/2\beta_4) \; . \tag{A.5}$$

Recall that it is assumed in Procedure 2 that the sufficient conditions for a minimum at (x^*,y^*) are always satisfied.

APPENDIX B

Documentation of Computer Programs

A. TITLE

Two separate programs, each designed to minimize the expected total distance traveled.

Program 1 - Procedure 1

Program 2 - Procedure 2

1. Programmer - Robert C. Burness
2. Machine - IBM 370/158
3. Language - FORTRAN IV
4. Date completed - April 1, 1974
5. Approximate compile time - Program 1 - 17 seconds

   Program 2 - 18 seconds
6. Execution time - Variable, depending on the number of existing facilities and the number of iterations required by each search technique.
7. Lines of output - Program 1 - Approximately 2,000 lines

   Program 2 - Approximately 2,000 lines
8. Storage required - Program 1 - 140,000 characters for 6 existing facilities

   Program 2 - 140,000 characters for 6 existing facilities

B. PURPOSE

Program 1 - This program finds the minimum distance location for the new facility by utilizing the Hyperbolic Approximation Procedure (Procedure 1, Chapter 3).

Program 2 - This program finds the minimum distance location for the new facility by utilizing the Successive Quadratic Approximation Procedure (Procedure 2, Chapter 3).

C. RESTRICTIONS

Programs 1 and 2 - All subscripted variables must be dimensioned exactly as indicated in the comments sections since variable dimensions are used in subroutines.

D. DEFINITIONS

The subscripted and unsubscripted variables are described in the listing.

E. COMPUTER SYSTEM LANGUAGE LIMITATIONS

The input and output statements are written with I/O numbers 5 and 6 respectively. Other systems may require different unit specifications.

Subroutines TIMEON and TIMECK, used for determining execution times, are supplied by the computer facility at Virginia Polytechnic Institute and State University.

F. LISTING OF PROGRAMS

Name of Subroutine	First Sequence No.
MAIN PROGRAM - PROCEDURE 1	PR1- 1
RANDU	PR1-332
HAP	PR1-344
OBJ	PR1-426
UPDATR	PR1-458
UPDATE	PR1-495
MAIN PROGRAM - PROCEDURE 2	PR2- 1
RANDU	PR2-471
GELS	PR2-481

The following subroutines are used in both programs:

TSP	TSP- 1
REDUCE	TSP-233
PENALT	TSP-298
NULINK	TSP-380

```
C***********************************************************************
C   MAIN PROGRAM--PROCEDURE 1
C***********************************************************************
C
C   EXPLANATION OF VARIABLES-SUBSCRIPTED VARIABLES ARE TO BE DIMENSIONED
C   EXACTLY AS GIVEN.
C
C   N-THE NUMBER OF EXISTING FACILITIES
C
C   M-N+1-THE TOTAL NUMBER OF FACILITIES
C
C   NPROB-THE NUMBER OF SUBSET PROBABILITIES
C
C   XN1(M),XN2(M)-COORDINATE POSITION OF FACILITY M
C
C   PROB(NPROB)-VECTOR OF SUBSET PROBABILITIES
C
C   D(M,M,N),E(M,M)-DISTANCE ARRAYS,THE M-TH ROW AND COLUMN OF WHICH
C   CONTAIN DISTANCES FROM NEW TO EXISTING FACILITIES
C
C   RPEN(M),CPEN(M)-ARRAYS CONTAINING ROW AND COLUMN PENALTIES-USED IN
C   TSP
C
C   W(N),WN(N)-ARRAYS CONTAINING LOWER BOUNDS-USED IN TSP
C
C   IP(M),JP(M)-ROUTE VECTORS-USED IN TSP
C
C   ITINR(N,10)-ARRAY CONTAINING OPTIMAL ROUTES-USED IN TSP
C
C   NPOINT(M)-ARRAY OF INDICATOR VARIABLES-USED TO GENERATE VARIOUS
C   SUBSETS OF FACILITIES TO BE VISITED
C
```

```
C   LINK(2,N)-ARRAY USED IN TSP TO PREVENT FORMATION OF SUBTOURS
C
C   WGT(1,N)-WEIGHTS USED IN HAP
C
C   WLAST(1,N)-WEIGHTS USED IN HAP THE PREVIOUS ITERATION-USED WITH THE
C   WGT ARRAY FOR THE OPTIMALITY CHECK
C
C   V(1,1)-ARRAY USED IN HAP
C
C   X(1),Y(1)-NEW FACILITY LOCATION-RETURNED FROM HAP
C
C   A(N),B(N)-EXISTING FACILITY LOCATIONS-USED IN HAP
C
      COMMON/HYPER/X,Y,EPS,V,MLM,IPRINT
      COMMON/TSP NL/NTSPS,NRTE
      COMMON/ALL/M,N,K,ITER,NOUT
      DOUBLE PRECISION X(1),Y(1),EPS,A(8),B(8)
      DIMENSION XN1(9),XN2(9),P(255),D(9,9,8),E(9,9),WGT(1,8),V(1,1),
     1CPEN(9),RPEN(9),W(8),WN(8),WLAST(1,8)
      INTEGER*2 LINK(2,8),IP(9),JP(9),NPOINT(9),ITINR(8,10)
C
C   IF IPRINT=1,PERTINENT INFORMATION BETWEEN ITERATIONS IS PRINTED OUT.
C   IF IPRINT=0,ONLY FINAL SOLUTION VALUES ARE PRINTED OUT.
C
      READ (5,460) IPRINT
      WRITE (6,460) IPRINT
C
C   NSTART IS THE TOTAL NUMBER OF DIFFERENT STARTING POSITIONS.
C
      READ (5,460) NSTART
      WRITE (6,460) NSTART
C
```

```
C   NITER IS THE MAXIMUM NUMBER OF ITERATIONS PER STARTING POSITION.
C
      READ (5,460) NITER
      WRITE (6,460) NITER
C
C   IF ISTART=-1,STARTING POSITIONS ARE GENERATED RANDOMLY.
C   IF ISTART= 0,STARTING POSITIONS ARE TO BE READ IN.
C   IF ISTART= 1,STARTING POSITIONS ARE THE EXISTING FACILTITY LOCATIONS.
C
      READ (5,460) ISTART
      WRITE (6,460) ISTART
C
C   M IS THE TOTAL NUMBER OF FACILITIES.(INCLUDES THE NEW FACILITY)
C
      READ (5,460) M
      WRITE (6,460) M
C
C   IF KP=1,RECTILINEAR DISTANCE MEASURE IS USED.
C   IF KP=2, EUCLIDEAN DISTANCE MEASURE IS USED.
C
      READ (5,460) KP
      WRITE (6,470) KP
      N=M-1
      MLM=N
      NN=N-1
      NPROB=2**N-1
      IX=32155
      BIG=10.**20
      ZBEST=BIG
      JSTART=1
      ITER=0
      NPOINT(M)=0
```

```
C
C   INPUT EXISTING FACILITY LOCATIONS.
C
       IF (IPRINT.EQ.1) WRITE (6,480)
       DO 20 I=1,N
       READ (5,490) XN1(I),XN2(I)
       IF (IPRINT.EQ.1) WRITE (6,490) XN1(I),XN2(I)
    20 CONTINUE
       DO 30 I=1,NPROB
       READ (5,500) P(I)
    30 WRITE (6,610) I,P(I)
C
C   COMPUTE THE CONSTANT ELEMENTS OF THE DISTANCE MATRIX.
C
       DO 50 I=1,N
       A(I)=XN1(I)
       B(I)=XN2(I)
       D(I,I,1)=BIG
       IF (I.EQ.N) GO TO 50
       II=I+1
       DO 40 J=II,N
       IF (KP.EQ.1) D(I,J,1)=ABS(XN1(I)-XN1(J))+ABS(XN2(I)-XN2(J))
       IF (KP.EQ.2) D(I,J,1)=SQRT((XN1(I)-XN1(J))**2+(XN2(I)-XN2(J))**2)
    40 D(J,I,1)=D(I,J,1)
    50 CONTINUE
       D(M,M,1)=BIG
C
C   FIND UPPER AND LOWER LIMITS FOR X1 AND X2.
C
       X1U=XN1(1)
       X1L=XN1(1)
       X2U=XN2(1)
```

```
      X2L=XN2(1)
      DO 60 I=2,N
      IF (XN1(I).LT.X1L) X1L=XN1(I)
      IF (XN1(I).GT.X1U) X1U=XN1(I)
      IF (XN2(I).LT.X2L) X2L=XN2(I)
      IF (XN2(I).GT.X2U) X2U=XN2(I)
   60 CONTINUE
C
C  BEGIN SEARCH PROCEDURE 1.
C  SET CLOCK TO ZERO.
C
      CALL TIMEON
   70 IF (ISTART) 80,90,100
C
C  RANDOMLY GENERATE THE INITIAL STARTING POSITION OF THE NEW FACILITY
C
   80 XN1(M)=(X1U-X1L)*RANDU(IX)+X1L
      XN2(M)=(X2U-X2L)*RANDU(IX)+X2L
      GO TO 110
   90 READ (5,490) XN1(M),XN2(M)
      GO TO 110
  100 XN1(M)=XN1(JSTART)
      XN2(M)=XN2(JSTART)
  110 X(1)=XN1(M)
      Y(1)=XN2(M)
C
C  COMPUTE THE NEW DISTANCES.
C
  120 DO 130 J=1,N
      IF (KP.EQ.1) D(M,J,1)=ABS(XN1(M)-XN1(J))+ABS(XN2(M)-XN2(J))
      IF (KP.EQ.2) D(M,J,1)=SQRT((XN1(M)-XN1(J))**2+(XN2(M)-XN2(J))**2)
  130 D(J,M,1)=D(M,J,1)
```

```
      IF (IPRINT) 160,160,140
  140 WRITE (6,580)
      WRITE (6,590) ITER
      WRITE (6,600) JSTART
      WRITE (6,510)
      WRITE (6,550) XN1(M),XN2(M)
      DO 150 I=1,M
  150 WRITE (6,620) (D(I,J,1),J=1,M)
  160 NTRIV=0
      FNO=0
      ITIN=1
      DO 170 I=1,N
  170 WGT(1,I)=0.
C
C  THIS SECTION GENERATES ALL (N,R) COMBINATIONS, R=1,2,....,N,USING A
C  GENERALIZATION OF AN ALGORITHM PRESENTED BY WELLS (21).
C
      DO 340 JK=1,N
      NOUT=N-JK
      RA=N+1
      RB=N-JK+1
      RC=JK+1
      NC=GAMMA(RA)/(GAMMA(RB)*GAMMA(RC))
      IF (JK.LT.3) GO TO 180
      NTRIV=NTRIV+JK*NC
C
C  GENERATE ALL COMBINATIONS OF R OBJECTS.
C
  180 DO 330 I=1,NC
      IF (JK.EQ.1) ICIT=I
      IF (I.GT.1) GO TO 200
      DO 190 J=1,N
```

```
      NPOINT(J)=0
  190 IF (J.GT.JK) NPOINT(J)=1
C
C  IF NPOINT(J) EQUALS 1, THEN FACILITY J IS NOT VISITED.
C
      GO TO 250
C
C  INTERCHANGE LEFTMOST (0,1) PAIR.
C
  200 DO 210 J=1,NN
      IF (NPOINT(J).NE.0.OR.NPOINT(J+1).NE.1) GO TO 210
      JG=J-1
      NPOINT(J)=1
      NPOINT(J+1)=0
      GO TO 220
  210 CONTINUE
C
C  MOVE ALL ZEROES TO THE LEFT OF THE INTERCHANGE POSITION AS FAR
C  TO THE LEFT AS POSSIBLE.
C
  220 IF (JG.LT.1) GO TO 250
      DO 240 J=1,JG
      IF (NPOINT(J).EQ.0) GO TO 240
      IF (J.EQ.JG) GO TO 240
      JH=J+1
      DO 230 JJ=JH,JG
      IF (NPOINT(JJ).NE.0) GO TO 230
      NPOINT(JJ)=1
      NPOINT(J)=0
      GO TO 240
  230 CONTINUE
  240 CONTINUE
```

```
C
C  IF NPOINT(J) EQUALS 1, THEN FACILITY J IS NOT VISITED.
C
  250 IF (JK-2) 260,270,320
  260 FNO=FNO+P(ITIN)*2*D(M,ICIT,1)
C
C  CALCULATE WEIGHTS FOR HAP.
C
      WGT(1,ICIT)=WGT(1,ICIT)+2*P(ITIN)
      GO TO 330
  270 ICIT1=0
      ICIT2=0
      DO 300 II=1,N
      IF (NPOINT(II).NE.0) GO TO 300
      IF (ICIT1) 280,280,290
  280 ICIT1=II
      GO TO 300
  290 ICIT2=II
      GO TO 310
  300 CONTINUE
  310 FNO=FNO+P(ITIN)*(D(M,ICIT1,1)+D(ICIT1,ICIT2,1)+D(ICIT2,M,1))
C
C  CALCULATE WEIGHTS FOR HAP.
C
      WGT(1,ICIT1)=WGT(1,ICIT1)+P(ITIN)
      WGT(1,ICIT2)=WGT(1,ICIT2)+P(ITIN)
      GO TO 330
  320 CALL TSP (D,E,W,WN,CPEN,RPEN,IP,JP,LINK,NPOINT,BEST,ITINR)
      FNO=FNO+P(ITIN)*BEST
C
C  CALCULATE WEIGHTS FOR HAP.
C
```

```
      ICIT3=ITINR(1,NRTE)
      WGT(1,ICIT3)=WGT(1,ICIT3)+P(ITIN)
      ICIT4=ITINR(JK,NRTE)
      WGT(1,ICIT4)=WGT(1,ICIT4)+P(ITIN)
  330 ITIN=ITIN+1
  340 CONTINUE
      IF (IPRINT) 360,360,350
  350 WRITE (6,570) ITER,XN1(M),XN2(M),FNO
  360 IF (ITER.NE.0) GO TO 380
      DO 370 I=1,N
  370 WLAST(1,I)=WGT(1,I)
      CALL HAP (A,B,WGT,NSTART,KP)
      XN1(M)=X(1)
      XN2(M)=Y(1)
      ITER=ITER+1
      GO TO 120
C
C  COMPARE WEIGHT VECTORS.
C
  380 DO 390 I=1,N
      IF (WGT(1,I).NE.WLAST(1,I)) GO TO 420
  390 CONTINUE
      IF (IPRINT) 410,410,400
  400 WRITE (6,560)
      WRITE (6,520) XN1(M),XN2(M),FNO
      WRITE (6,560)
  410 GO TO 440
  420 IF (ITER.EQ.NITER) GO TO 440
C
C  SAVE MOST CURRENT WEIGHT VECTOR.
C
      DO 430 I=1,N
```

```
  430 WLAST(1,I)=WGT(1,I)
      CALL HAP (A,B,WGT,NSTART,KP)
      XN1(M)=X(1)
      XN2(M)=Y(1)
      ITER=ITER+1
      GO TO 120
  440 JSTART=JSTART+1
      IF (FNO.LT.ZBEST) ZBEST=FNO
      IF (JSTART.GT.NSTART) GO TO 450
      IF (ITER.EQ.NITER) GO TO 450
      ITER=0
      GO TO 70
  450 CONTINUE
C
C  CHECK EXECUTION TIME.
C
      CALL TIMECK (NTIME)
      TIME=NTIME/100.
      WRITE (6,530) ZBEST
      WRITE (6,540) TIME
      STOP
C
C
  460 FORMAT (1X,I4)
  470 FORMAT (//,T10,4HKP= ,I5,//)
  480 FORMAT (12X,6HXN1(I),T32,6HXN2(I))
  490 FORMAT (9X,F10.3,9X,F10.3)
  500 FORMAT (F10.5)
  510 FORMAT (T20,23HSTARTING POSITIONS ARE:,/,T25,1HX,T35,1HY)
  520 FORMAT (T20,46HA LOCAL MINIMUM HAS BEEN FOUND USING HAP:  X1=,F10.
     14,5H, X2=,F10.4,4H, Z=,F10.4)
  530 FORMAT (T10,19HMINIMUM SOLUTION IS,3X,F10.4)
```

```
540 FORMAT (T10,17HEXECUTION TIME=  ,F10.3,7H  SECS.)                   PR1  321
550 FORMAT (19X,2F10.4)                                                 PR1  322
560 FORMAT (//,110(1H*))                                                PR1  323
570 FORMAT (T10,11HITERATION #,I3,T26,25H--NEW FACILITY LOCATED AT,10HPR1  324
   1 (X1,X2)=(,F7.1,3H , ,F7.1,9H) WITH Z=,F9.3)                        PR1  325
580 FORMAT (110(1H-))                                                   PR1  326
590 FORMAT (T48,11HITERATION #,I3)                                      PR1  327
600 FORMAT (T44,19HSTARTING POSITION #,I3)                              PR1  328
610 FORMAT (//,T10,3HP( ,I2,3H )=,F7.3)                                 PR1  329
620 FORMAT (T2,7F9.3)                                                   PR1  330
    END                                                                 PR1  331
```

```
      FUNCTION RANDU(IX)
C
C
      IY=IX*65539
      IF (IY) 10,20,20
   10 IY=IY+2147483647+1
   20 YFL=IY
      YFL=YFL*.4656613E-9
      IX=IY
      RANDU=YFL
      RETURN
      END
```

```
      SUBROUTINE HAP (A,B,WGT,NSTART,KP)
C
C     HYPERBOLIC APPROXIMATION PROCEDURE (HAP)
C     DETERMINES NEW FACILITY LOCATION FOR RECTILINEAR (KP=1) OR EUCLIDEAN
C     (KP=2) DISTANCES
C
      COMMON/HYPER/X,Y,EPS,V,MLM,IPRINT
      DOUBLE PRECISION X(1),Y(1),EPS,F,FS,DABS,A(MLM),B(MLM)
      DIMENSION V(1,1),WGT(1,MLM)
      INTEGER COUNT
   10 CONTINUE
      NOUT=0
      NSTOP=6
      IE1=2
      IE2=12
      IEIND=2
      IF (KP.NE.4) KP1=0
      IF (KP.EQ.4) KP1=4
      IF (KP1.EQ.4) KP=1
      N=1
      M=MLM
      V(1,1)=0.
      IF (IPRINT) 40,40,20
   20 WRITE (6,150)
      DO 30 I=1,M
      WRITE (6,140) A(I),B(I),(WGT(J,I),J=1,N)
   30 CONTINUE
   40 CONTINUE
   50 CONTINUE
      NUMBER=0
C
C     SET HYPERBOLOID CONSTANT
```

```
C
      DO 110 IE=IE1,IE2,IEIND
      IF (KP.EQ.3) GO TO 60
      EPS=.1**IE
      IF (NOUT.NE.0) WRITE (6,130) EPS
   60 COUNT=0
C
C  CALCULATING VALUE OF OBJECTIVE FUNCTION
C
   70 CALL OBJ (A,B,WGT,F,KP)
      IF (COUNT.EQ.0) GO TO 100
C
C  CHECKING STOPPING CRITERION
C
      IF (DABS(F-FS).GT..1**NSTOP*F) GO TO 100
C
C  PRINTOUT ITERATIVE RESULTS
C
      IF (NOUT.EQ.0) GO TO 90
      WRITE (6,170) NUMBER
      DO 80 J=1,N
   80 WRITE (6,160) J,X(J),J,Y(J)
      WRITE (6,180) F
   90 CONTINUE
      IF (KP.EQ.3) GO TO 120
      GO TO 110
  100 FS=F
      NUMBER=NUMBER+1
      COUNT=COUNT+1
C
C  CALCULATING NEXT SET OF ITERATIVE VALUES
C
```

```
      IF (KP.EQ.1) CALL UPDATR (A,B,WGT)
      IF (KP.EQ.2) CALL UPDATE (A,B,WGT)
      GO TO 70
  110 CONTINUE
  120 CONTINUE
      IF (KP.EQ.3) GO TO 10
      IF (KP1.EQ.4) KP=KP+1
      IF (KP1.EQ.4) GO TO 50
      RETURN
  130 FORMAT (//10X,23HHYPERBOLOID CONSTANT = ,D10.3)
  140 FORMAT (/5X,10F9.2)
  150 FORMAT (64H LOCATION OF EXISTING FACILITIES AND THEIR RESPECTIVE W
     1EIGHTS(W),/)
  160 FORMAT (//15X,3HX1(,I2,3H)= ,D20.12,6H    X2(,I2,3H)= ,D20.12)
  170 FORMAT (//10X,6HAFTER ,I5,38H ITERATIONS THE OPTIMAL LOCATIONS ARE
     1:)
  180 FORMAT (//15X,21HOBJECTIVE FUNCTION = ,D20.12)
      END
```

```
      SUBROUTINE OBJ (A,B,WGT,F,KP)
C
C   EVALUATES OBJECTIVE FUNCTION
C
      COMMON/HYPER/X,Y,EPS,V,MLM,IPRINT
      DOUBLE PRECISION D,FW,FV,F,X(1),Y(1),EPS,DSQRT,R,S,A(MLM),B(MLM)
      DOUBLE PRECISION G,H,DR,FACT,FAK,DABS,DS
      DIMENSION V(1,1),WGT(1,MLM)
      D(R,S,G,H)=DSQRT((R-G)**2+(S-H)**2)
      DR(R,S,G,H)=DABS(R-G)+DABS(S-H)
      DS(R,S,G,H)=(R-G)**2+(S-H)**2
      N=1
      M=MLM
      FV=0.
      FW=0.
      DO 30 J=1,N
      JP1=J+1
      IF (JP1.GT.N) GO TO 20
      DO 10 K=JP1,N
      IF (KP.EQ.1) FACT=DR(X(J),Y(J),X(K),Y(K))
      IF (KP.EQ.2) FACT=D(X(J),Y(J),X(K),Y(K))
      IF (KP.EQ.3) FACT=DS(X(J),Y(J),X(K),Y(K))
   10 FV=FV+V(J,K)*FACT
   20 CONTINUE
      DO 30 I=1,M
      IF (KP.EQ.1) FAK=DR(X(J),Y(J),A(I),B(I))
      IF (KP.EQ.2) FAK=D(X(J),Y(J),A(I),B(I))
      IF (KP.EQ.3) FAK=DS(X(J),Y(J),A(I),B(I))
   30 FW=FW+WGT(J,I)*FAK
      F=FV+FW
      RETURN
      END
```

```
      SUBROUTINE UPDATR (A,B,WGT)
C
C  DETERMINES NEW VALUES FOR X AND Y FOR RECTILINEAR PROBLEM
C
      COMMON/HYPER/X,Y,EPS,V,MLM,IPRINT
      DIMENSION V(1,1),WGT(1,MLM)
      DOUBLE PRECISION D,SUMD,SUMA,EPS,NUMVX,NUMVY,NUMWX,NUMWY,DENOVX
      DOUBLE PRECISION DENOVY,X(1),Y(1),DSQRT,R,A(MLM),B(MLM),G
      DOUBLE PRECISION DENOWX,DENOWY
      D(R,G,EPS)=DSQRT((R-G)**2+EPS)
      SUMD(T,R,G,EPS)=T/D(R,G,EPS)
      SUMA(T,R,G,EPS)=T*G/D(R,G,EPS)
      M=MLM
      N=1
      DO 30 J=1,N
      NUMWX=0.
      NUMWY=0.
      DENOWX=0.
      DENOWY=0.
      NUMVX=0.
      NUMVY=0.
      DENOVX=0.
      DENOVY=0.
      DO 10 I=1,M
      NUMWX=NUMWX+SUMA(WGT(J,I),X(J),A(I),EPS)
      NUMWY=NUMWY+SUMA(WGT(J,I),Y(J),B(I),EPS)
      DENOWX=DENOWX+SUMD(WGT(J,I),X(J),A(I),EPS)
   10 DENOWY=DENOWY+SUMD(WGT(J,I),Y(J),B(I),EPS)
      DO 20 K=1,N
      NUMVX=NUMVX+SUMA(V(J,K),X(J),X(K),EPS)
      NUMVY=NUMVY+SUMA(V(J,K),Y(J),Y(K),EPS)
      DENOVX=DENOVX+SUMD(V(J,K),X(J),X(K),EPS)
```

```
20 DENOVY=DENOVY+SUMD(V(J,K),Y(J),Y(K),EPS)
   X(J)=(NUMVX+NUMWX)/(DENOVX+DENOWX)
30 Y(J)=(NUMVY+NUMWY)/(DENOVY+DENOWY)
   RETURN
   END
```

```
      SUBROUTINE UPDATE (A,B,WGT)
C
C  DETERMINES NEW VALUES FOR X AND Y FOR EUCLIDEAN PROBLEM
C
      COMMON/HYPER/X,Y,EPS,V,MLM,IPRINT
      DIMENSION V(1,1),WGT(1,MLM)
      DOUBLE PRECISION D,SUMD,SUMA,SUMB,EPS,NUMVX,NUMVY,NUMWX,NUMWY,H
      DOUBLE PRECISION DENOMV,DENIMW,X(1),Y(1),DSQRT,R,S,A(MLM),B(MLM),G
      DOUBLE PRECISION E
      D(R,S,G,H,E)=DSQRT((R-G)**2+(S-H)**2+E)
      SUMD(T,R,S,G,H,E)=T/D(R,S,G,H,E)
      SUMA(T,R,S,G,H,E)=T*G/D(R,S,G,H,E)
      SUMB(T,R,S,G,H,E)=T*H/D(R,S,G,H,E)
      E=EPS
      N=1
      M=MLM
      DO 30 J=1,N
      NUMWX=0.
      NUMWY=0.
      DENOMW=0.
      NUMVX=0.
      NUMVY=0.
      DENOMV=0.
      DO 10 I=1,M
      NUMWX=NUMWX+SUMA(WGT(J,I),X(J),Y(J),A(I),B(I),E)
      NUMWY=NUMWY+SUMB(WGT(J,I),X(J),Y(J),A(I),B(I),E)
   10 DENOMW=DENOMW+SUMD(WGT(J,I),X(J),Y(J),A(I),B(I),E)
      DO 20 K=1,N
      NUMVX=NUMVX+SUMA(V(J,K),X(J),Y(J),X(K),Y(K),E)
      NUMVY=NUMVY+SUMB(V(J,K),X(J),Y(J),X(K),Y(K),E)
   20 DENOMV=DENOMV+SUMD(V(J,K),X(J),Y(J),X(K),Y(K),E)
      X(J)=(NUMVX+NUMWX)/(DENOMV+DENOMW)
```

```
30 Y(J)=(NUMVY+NUMWY)/(DENOMV+DENOMW)
   RETURN
   END
```

```
C*********************************************************************PR2
C   MAIN PROGRAM--PROCEDURE 2                                         PR2
C*********************************************************************PR2
C   EXPLANATION OF NEW VARIABLES-SUBSCRIPTED VARIABLES ARE TO BE      PR2
C   DIMENSIONED EXACTLY AS GIVEN.                                     PR2
C                                                                     PR2
C   NOB-THE NUMBER OF INITIAL OBSERVATIONS                            PR2
C                                                                     PR2
C   X1(NOB),X2(NOB)-COORDINATE POSITIONS OF RANDOMLY GENERATED NEW    PR2
C   FACILITIES                                                        PR2
C                                                                     PR2
C   Y(NOB)-OBJECTIVE FUNCTION VALUES CORRESPONDING TO RANDOMLY GENERATED PR2
C   COORDINATE POSITIONS                                              PR2
C                                                                     PR2
C   A(15)-ARRAY USED IN REGRESSION PROCEDURE (AB=C)                   PR2
C                                                                     PR2
C   B(5)-VECTOR USED IN REGRESSION PROCEDURE                          PR2
C                                                                     PR2
C   SUM(20)-USED FOR ACCUMULATING ENTRIES IN THE A ARRAY              PR2
C                                                                     PR2
C   AUX(4)-AUXILIARY ARRAY USED IN GELS                               PR2
C                                                                     PR2
C   DELTA-STOPPING CRITERIA FOR PROCEDURE 2                           PR2
C                                                                     PR2
      COMMON/TSP NL/NTSPS,NRTE                                        PR2
      COMMON/ALL/M,N,K,ITER,NOUT                                      PR2
      DIMENSION SUM(20),A(15),R(5),AUX(4)                             PR2
      DIMENSION XN1(7),XN2(7),X1(10),X2(10),Y(10),P(63)               PR2
      DIMENSION  RPEN(7),CPEN(7),E(7,7),W(6),WN(6)                    PR2
      DIMENSION D(7,7,6)                                              PR2
      INTEGER*2 LINK(2,6),NPOINT(7),ITINR(6,10),IP(7),JP(7)           PR2
C                                                                     PR2
```

```
C   IF IPRINT=1,PERTINENT INFORMATION BETWEEN ITERATIONS IS PRINTED OUT.
C   IF IPRINT=0,ONLY FINAL SOLUTION VALUES ARE PRINTED OUT.
C
       READ (5,470) IPRINT
       WRITE (6,470) IPRINT
C
C   NITER IS THE MAXIMUM NUMBER OF ITERATIONS.
C
       READ (5,470) NITER
       WRITE (6,470) NITER
C
C   M IS THE TOTAL NUMBER OF FACILITIES.(INCLUDES THE NEW FACILITY)
C
       READ (5,470) M
       WRITE (6,470) M
C
C   NOB IS THE NUMBER OF INITIAL OBSERVATIONS
C
       READ (5,470) NOB
       WRITE (6,470) NOB
C
C   IF KP=1,RECTILINEAR DISTANCE MEASURE IS USED.
C   IF KP=2, EUCLIDEAN DISTANCE MEASURE IS USED.
C
       READ (5,470) KP
       WRITE (6,480) KP
       N=M-1
       NN=N-1
       NPROB=2**N-1
       NPOINT(M)=0
       ITER=0
       BIG=10.**20
```

```
      STARTY=BIG
      BESTY=BIG
      III=1
      ISTOP=0
      DELTA=5.*10.**(-2)
      EPS=10.**(-6.5)
      NEQ=5
      NCOEF=(NEQ*(NEQ+1))/2
      MAUX=NEQ-1
      NSUM=20
      BIG=10.**20
      IX=32155
C
C   INPUT EXISTING FACILITY LOCATIONS.
C
      IF (IPRINT.EQ.1) WRITE (6,490)
      DO 20 I=1,N
      READ (5,500) XN1(I),XN2(I)
      IF (IPRINT.EQ.1) WRITE (6,500) XN1(I),XN2(I)
   20 CONTINUE
      DO 30 I=1,NPROB
      READ (5,510) P(I)
   30 WRITE (6,660) I,P(I)
C
C   BEGIN SEARCH PROCEDURE 2
C   SET CLOCK TO ZERO.
      CALL TIMEON
C
C   FIND UPPER AND LOWER LIMITS FOR X1 AND X2.
C
      X1U=XN1(1)
      X2U=XN2(1)
```

```
      X1L=XN1(1)
      X2L=XN2(1)
      DO 40 I=2,N
      IF (XN1(I).LT.X1L) X1L=XN1(I)
      IF (XN1(I).GT.X1U) X1U=XN1(I)
      IF (XN2(I).LT.X2L) X2L=XN2(I)
      IF (XN2(I).GT.X2U) X2U=XN2(I)
   40 CONTINUE
C
C   COMPUTE CONSTANT ELEMENTS OF THE DISTANCE MATRIX.
C
      DO 60 I=1,N
      IF (IPRINT.EQ.1) WRITE (6,540) I,XN1(I),I,XN2(I)
      D(I,I,1)=BIG
      IF (I.EQ.N) GO TO 60
      II=I+1
      DO 50 J=II,N
      IF (KP.EQ.1) D(I,J,1)=ABS(XN1(I)-XN1(J))+ABS(XN2(I)-XN2(J))
      IF (KP.EQ.2) D(I,J,1)=SQRT((XN1(I)-XN1(J))**2+(XN2(I)-XN2(J))**2)
      D(J,I,1)=D(I,J,1)
   50 D(J,I,1)=D(I,J,1)
   60 CONTINUE
      D(M,M,1)=BIG
C
C   RANDOMLY GENERATE THE INITIAL LOCATION OF THE NEW FACILITY AND
C   COMPUTE THE NEW DISTANCES.
C
   70 XN1(M)=(X1U-X1L)*RANDU(IX)+X1L
      XN2(M)=(X2U-X2L)*RANDU(IX)+X2L
      DO 80 J=1,N
      IF (KP.EQ.1) D(M,J,1)=ABS(XN1(M)-XN1(J))+ABS(XN2(M)-XN2(J))
      IF (KP.EQ.2) D(M,J,1)=SQRT((XN1(M)-XN1(J))**2+(XN2(M)-XN2(J))**2)
```

```
   80 D(J,M,1)=D(M,J,1)
      GO TO 200
   90 Y(III)=FND
      X1(III)=XN1(M)
      X2(III)=XN2(M)
      IF (IPRINT.EQ.1) WRITE (6,550) III,X1(III),X2(III),Y(III)
      IF (IPRINT.EQ.1) WRITE (6,600)
C
C  RETAIN BEST EVALUATION OBTAINED RANDOMLY.
C
      IF (Y(III).GT.STARTY) GO TO 100
      STARTY=Y(III)
      START1=X1(III)
      START2=X2(III)
  100 III=III+1
      IF (III.LE.NOB) GO TO 70
      IF (IPRINT.EQ.1) WRITE (6,560)
C
C  PLACE LEAST OPTIMAL POINT IN THE LAST POSITION OF THE ARRAY.
C
      WORST=0.
      X1TEMP=X1(NOB)
      X2TEMP=X2(NOB)
      YTEMP=Y(NOB)
      DO 110 I=1,NOB
      IF (Y(I).GT.WORST) WORST=Y(I)
      IF (Y(I).EQ.WORST) IDUM=I
  110 CONTINUE
      Y(NOB)=Y(IDUM)
      X1(NOB)=X1(IDUM)
      X2(NOB)=X2(IDUM)
      Y(IDUM)=YTEMP
```

```
      X1(IDUM)=X1TEMP
      X2(IDUM)=X2TEMP
C
C  ACCUMULATE SUMS FOR REGRESSION PROCEDURE.
C
      DO 120 I=1,NSUM
  120 SUM(I)=0.
      DO 130 I=1,NOB
      SUM(1)=SUM(1)+Y(I)
      SUM(2)=SUM(2)+X1(I)
      SUM(3)=SUM(3)+X1(I)**2
      SUM(4)=SUM(4)+X2(I)
      SUM(5)=SUM(5)+X1(I)*X2(I)
      SUM(6)=SUM(6)+X2(I)**2
      SUM(8)=SUM(8)+X1(I)**3
      SUM(9)=SUM(9)+X1(I)**2*X2(I)
      SUM(10)=SUM(10)+X1(I)**4
      SUM(12)=SUM(12)+X1(I)*X2(I)**2
      SUM(13)=SUM(13)+X2(I)**3
      SUM(14)=SUM(14)+(X2(I)*X1(I))**2
      SUM(15)=SUM(15)+X2(I)**4
      SUM(16)=SUM(16)+Y(I)
      SUM(17)=SUM(17)+X1(I)*Y(I)
      SUM(18)=SUM(18)+X2(I)*Y(I)
      SUM(19)=SUM(19)+Y(I)*X1(I)**2
  130 SUM(20)=SUM(20)+Y(I)*X2(I)**2
      YBAR=SUM(1)/NOB
      A(1)=NOB
      A(2)=SUM(2)
      A(3)=SUM(3)
      A(4)=SUM(4)
      A(5)=SUM(5)
```

```
      A(6)=SUM(6)
      A(7)=A(3)
      A(8)=SUM(8)
      A(9)=SUM(9)
      A(10)=SUM(10)
      A(11)=A(6)
      A(12)=SUM(12)
      A(13)=SUM(13)
      A(14)=SUM(14)
      A(15)=SUM(15)
      R(1)=SUM(16)
      R(2)=SUM(17)
      R(3)=SUM(18)
      R(4)=SUM(19)
      R(5)=SUM(20)
C
C  SOLVE FOR THE REGRESSION COEFFICIENT ESTIMATES.
C
  140 CALL GELS (R,A,NEQ,1,EPS,IER,AUX,NCOEF,MAUX)
      ITER=ITER+1
      IF (ITER.EQ.NITER) GO TO 460
C
C  FIND COEFFICIENT OF DETERMINATION.
C
      SST=0.
      SSR=0.
      DO 150 I=1,NOB
      SST=SST+(Y(I)-YBAR)**2
  150 SSR=SSR+(R(1)+R(2)*X1(I)+R(3)*X2(I)+R(4)*X1(I)**2+R(5)*X2(I)**2-YB
     1AR)**2
      R2=SSR/SST
      IF (IPRINT) 170,170,160
```

```
  160 WRITE (6,570)
      WRITE (6,580) ITER
      WRITE (6,640) (R(I),I=1,5)
      WRITE (6,670) SSR,SST
      WRITE (6,700) YBAR
      WRITE (6,680) R2
  170 X1OUT=X1(NOB)
      X2OUT=X2(NOB)
      YOUT=Y(NOB)
C
C   DETERMINE THE MINIMUM POINT ON THE REGRESSION SURFACE AND REPLACE THE
C   LEAST OPTIMAL POINT IN SOLUTION.
C
      X1(NOB)=-R(2)/(2.*R(4))
      X2(NOB)=-R(3)/(2.*R(5))
C
C   CONSTRAIN THE COORDINATES OF NEW FACILITY LOCATION TO BE WITHIN THE
C   LIMITS ESTABLISHED FOR X1 AND X2.
C
      IF (X1(NOB).GT.X1U) X1(NOB)=X1U
      IF (X1(NOB).LT.X1L) X1(NOB)=X1L
      IF (X2(NOB).GT.X2U) X2(NOB)=X2U
      IF (X2(NOB).LT.X2L) X2(NOB)=X2L
      YY=R(1)+R(2)*X1(NOB)+R(3)*X2(NOB)+R(4)*X1(NOB)**2+R(5)*X2(NOB)**2
      IF (IPRINT.EQ.1) WRITE (6,650) X1(NOB),X2(NOB),YY
C
C   CALCULATE NEW DISTANCE MATRIX ELEMENTS.
C
      DO 180 I=1,N
      IF (KP.EQ.1) D(M,I,1)=ABS(X1(NOB)-XN1(I))+ABS(X1(NOB)-XN2(I))
      IF (KP.EQ.2) D(M,I,1)=SQRT((X1(NOB)-XN1(I))**2+(X2(NOB)-XN2(I))**2
     1)
```

```
  180 D(I,M,1)=D(M,I,1)
      D(M,M,1)=BIG
      IF (IPRINT.NE.1) GO TO 200
      DO 190 I=1,M
  190 WRITE (6,690) (D(I,J,1),J=1,M)
C
C  THIS SECTION GENERATES ALL (N,R) COMBINATIONS, R=1,2,...,N.
C
  200 NTRIV=0
      FND=0
      ITIN=1
      DO 370 JK=1,N
      NOUT=N-JK
      RA=N+1
      RB=N-JK+1
      RC=JK+1
      NC=GAMMA(RA)/(GAMMA(RB)*GAMMA(RC))
      IF (JK.LT.3) GO TO 210
      NTRIV=NTRIV+JK*NC
C
C  GENERATE ALL COMBINATIONS OF R OBJECTS.
C
  210 DO 360 I=1,NC
      IF (JK.EQ.1) ICIT=I
      IF (I.GT.1) GO TO 230
      DO 220 J=1,N
      NPOINT(J)=0
  220 IF (J.GT.JK) NPOINT(J)=1
C
C  IF NPOINT(J) EQUALS 1, THEN FACILITY J IS NOT VISITED.
C
      GO TO 280
```

```
C
C  INTERCHANGE LEFTMOST (0,1) PAIR.
C
  230 DO 240 J=1,NN
      IF (NPOINT(J).NE.0.OR.NPOINT(J+1).NE.1) GO TO 240
      JG=J-1
      NPOINT(J)=1
      NPOINT(J+1)=0
      GO TO 250
  240 CONTINUE
C
C  MOVE ALL ZEROES TO THE LEFT OF THE INTERCHANGE POSITION AS FAR
C  TO THE LEFT AS POSSIBLE.
C
  250 IF (JG.LT.1) GO TO 280
      DO 270 J=1,JG
      IF (NPOINT(J).EQ.0) GO TO 270
      IF (J.EQ.JG) GO TO 270
      JH=J+1
      DO 260 JJ=JH,JG
      IF (NPOINT(JJ).NE.0) GO TO 260
      NPOINT(JJ)=1
      NPOINT(J)=0
      GO TO 270
  260 CONTINUE
  270 CONTINUE
C
C  IF NPOINT(J) EQUALS 1, THEN FACILITY J IS NOT VISITED.
C
  280 IF (JK-2) 290,300,350
  290 FNO=FNO+P(ITIN)*2*D(M,ICIT,1)
      GO TO 360
```

```
  300 ICIT1=0
      ICIT2=0
      DO 330 II=1,N
      IF (NPOINT(II).NE.0) GO TO 330
      IF (ICIT1) 310,310,320
  310 ICIT1=II
      GO TO 330
  320 ICIT2=II
      GO TO 340
  330 CONTINUE
  340 FNO=FNO+P(ITIN)*(D(M,ICIT1,1)+D(ICIT1,ICIT2,1)+D(ICIT2,M,1))
      GO TO 360
  350 CALL TSP (D,E,W,WN,CPEN,RPEN,IP,JP,LINK,NPOINT,BEST,ITINR)
      FNO=FNO+P(ITIN)*BEST
      ICIT3=ITINR(1,NRTE)
      ICIT4=ITINR(JK,NRTE)
  360 ITIN=ITIN+1
  370 CONTINUE
      IF (III.LE.NOB) GO TO 90
      Y(NOB)=FNO
      IF (IPRINT) 400,400,380
  380 WRITE (6,710)
      DO 390 I=1,NOB
  390 WRITE (6,720) I,X1(I),X2(I),Y(I)
      WRITE (6,590) ITER,X1(NOB),X2(NOB),Y(NOB)
      WRITE (6,600)
C
C  RETAIN BEST EVALUATION OBTAINED IN SEARCH.
C
  400 IF (Y(NOB).GT.BESTY) GO TO 410
      BESTY=Y(NOB)
      BEST1=X1(NOB)
```

```
      BEST2=X2(NOB)
      IBEST=ITER
C
C  TERMINATE IF DIFFERENCE BETWEEN SUCCESSIVE FUNCTIONAL EVALUATIONS
C  IS LESS THAN THE SPECIFIED DELTA.
C
  410 IF (ITER.EQ.1) GO TO 420
      SAVE=Y(NOB)
C
C  CHECK TO SEE IF STATIONARY POINT IS A MAXIMUM.
C
      IF (ABS(SAVE-YTEMP).LE.DELTA.AND.YTEMP.NE.YOUT) GO TO 460
      IF (YTEMP.NE.YOUT) GO TO 420
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.3) GO TO 460
C
C  MOVE AWAY FROM MAXIMUM POINT AND EVALUATE F(X,Y)--THIS IS
C  ONLY ALLOWED TWICE.
C
      X1(NOB)=(RANDU(IX)-.5)*20.
      X2(NOB)=(RANDU(IX)-.5)*20.
      GO TO 200
C
C  REPOSITION WORST OBSERVATION TO LAST STORAGE AREA IN EACH ARRAY.
C
  420 NM=NOB-1
      X1TEMP=X1(NOB)
      X2TEMP=X2(NOB)
      YTEMP=Y(NOB)
      WORST=0.
      DO 430 I=1,NM
      IF (Y(I).GT.WORST) WORST=Y(I)
```

```
      IF (Y(I).EQ.WORST) IDUM=I
  430 CONTINUE
      X1(NOB)=X1(IDUM)
      X2(NOB)=X2(IDUM)
      Y(NOB)=Y(IDUM)
      X1(IDUM)=X1TEMP
      X2(IDUM)=X2TEMP
      Y(IDUM)=YTEMP
C
C  DETERMINE NEW COEFFICIENTS IN THE A MATRIX AND THE R VECTOR.
C
      A(1)=NOB
      A(2)=SUM(2)-X1OUT+X1TEMP
      A(3)=SUM(3)-X1OUT**2+X1TEMP**2
      A(4)=SUM(4)-X2OUT+X2TEMP
      A(5)=SUM(5)-X1OUT*X2OUT+X1TEMP*X2TEMP
      A(6)=SUM(6)-X2OUT**2+X2TEMP**2
      A(7)=A(3)
      A(8)=SUM(8)-X1OUT**3+X1TEMP**3
      A(9)=SUM(9)-X1OUT**2*X2OUT+X1TEMP**2*X2TEMP
      A(10)=SUM(10)-X1OUT**4+X1TEMP**4
      A(11)=A(6)
      A(12)=SUM(12)-X1OUT*X2OUT**2+X1TEMP*X2TEMP**2
      A(13)=SUM(13)-X2OUT**3+X2TEMP**3
      A(14)=SUM(14)-(X1OUT*X2OUT)**2+(X1TEMP*X2TEMP)**2
      A(15)=SUM(15)-X2OUT**4+X2TEMP**4
      R(1)=SUM(16)-YOUT+YTEMP
      R(2)=SUM(17)-YOUT*X1OUT+YTEMP*X1TEMP
      R(3)=SUM(18)-YOUT*X2OUT+YTEMP*X2TEMP
      R(4)=SUM(19)-YOUT*X1OUT**2+YTEMP*X1TEMP**2
      R(5)=SUM(20)-YOUT*X2OUT**2+YTEMP*X2TEMP**2
      YBAR=R(1)/NOB
```

```
      DO 440 I=1,15
  440 SUM(I)=A(I)
      DO 450 I=16,20
  450 SUM(I)=R(I-15)
      GO TO 140
C
C  TERMINATE SEARCH.
C  CHECK EXECUTION TIME
C
      CALL TIMECK (NTIME)
  460 CONTINUE
      WRITE (6,610) IBEST,BESTY,BEST1,BEST2
      WRITE (6,620) STARTY,START1,START2
      IF (STARTY.LT.BESTY) BESTY=STARTY
      WRITE (6,520) BESTY
      TIME=NTIME/100.
      WRITE (6,530) TIME
      WRITE (6,630)
      STOP
C
C
  470 FORMAT (1X,I4)
  480 FORMAT (//,T10,4HKP= ,I5,//)
  490 FORMAT (12X,6HXN1(I),T32,6HXN2(I))
  500 FORMAT (9X,F10.3,9X,F10.3)
  510 FORMAT (F10.5)
  520 FORMAT (T10,20HMINIMUM SOLUTION IS ,3X,F10.4)
  530 FORMAT (T10,16HEXECUTION TIME= ,F10.3,6H SECS.)
  540 FORMAT (//,T10,3HX1(,I2,3H )=,F7.3,T30,3HX2(,I2,3H )=,F7.3)
  550 FORMAT (//,T10,13HOBSERVATION #,I3,T30,41HNEW FACILITY LOCATED AT
     1     (X1,   X2)=( ,F7.2,4H ,   ,F7.2,11H ) WITH Y=,,F8.2,//)
  560 FORMAT (//////,T70,22HBEGIN SEARCH PROCEDURE,////)
```

```
570 FORMAT (T10,100(1H*))
580 FORMAT (T50,11HITERATION #,I6)
590 FORMAT (//,T10,11HITERATION #,I3,T30,34HNEW FACILITY LOCATED AT (X
   11,X2)=( ,F7.2,4H ,  ,F7.2,26H ) WITH ACTUAL VALUE OF Y=,F8.2)
600 FORMAT (//,T10,100(1H-))
610 FORMAT (T10,62HBEST OBSERVATION OBTAINED DURING SEARCH WAS AT
   1ITERATION #,I4,9H  WITH Y=,F10.3,5H ,X1=,F10.3,5H ,X2=,F10.3)
620 FORMAT (T10,42HBEST OBSERVATION OBTAINED AT RANDOM WAS Y=,F10.3,T6
   15,6HAT X1=,F10.3,T85,4H,X2=,F10.3)
630 FORMAT (1H1)
640 FORMAT (//,T10,4HY=  ,F10.3,T25,3H+  ,F10.3,T40,6HX1 +  ,F10.3,T58
   1,6HX2 +  ,F10.3,T76,9HX1**2 +  ,F10.3,T97,5HX2**2)
650 FORMAT (//,T10,24HWHOSE MINIMUM IS AT X1= ,F15.4,6H  ,X2=,F15.4,9H
   1WITH Y=  ,F10.3)
660 FORMAT (//,T10,3HP( ,I2,3H )=,F7.3)
670 FORMAT (//,T10,4HSSR=,F12.3,T30,4HSST=,F12.3)
680 FORMAT (//,T10,10HR SQUARED=,F7.3)
690 FORMAT (//,T10,5F10.4)
700 FORMAT (//,T10,5HYBAR=,F10.4)
710 FORMAT (//,T10,3HNOB,T23,7HX1(NOB),T40,7HX2(NOB),T55,6HY(NOB))
720 FORMAT (T10,I3,T21,F10.4,T37,F10.4,T52,F10.4)
    END
```

```
      FUNCTION RANDU(IX)
C
C
      IY=IX*65539
      IF (IY) 10,20,20
   10 IY=IY+2147483647+1
   20 RANDU=IY*.4656613E-9
      IX=IY
      RETURN
      END
```

```
      SUBROUTINE GELS (R,A,M,N,EPS,IER,AUX,NCOEF,MAUX)
C
C     ..................................................................
C
C        SUBROUTINE GELS
C
C        PURPOSE
C           TO SOLVE A SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS WITH
C           SYMMETRIC COEFFICIENT MATRIX UPPER TRIANGULAR PART OF WHICH
C           IS ASSUMED TO BE STORED COLUMNWISE.
C
C        USAGE
C           CALL GELS(R,A,M,N,EPS,IER,AUX)
C
C        DESCRIPTION OF PARAMETERS
C           R      - M BY N RIGHT HAND SIDE MATRIX.  (DESTROYED)
C                    ON RETURN R CONTAINS THE SOLUTION OF THE EQUATIONS.
C           A      - UPPER TRIANGULAR PART OF THE SYMMETRIC
C                    M BY M COEFFICIENT MATRIX.  (DESTROYED)
C           M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.
C           N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.
C           EPS    - AN INPUT CONSTANT WHICH IS USED AS RELATIVE
C                    TOLERANCE FOR TEST ON LOSS OF SIGNIFICANCE.
C           IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS
C                    IER=0  - NO ERROR,
C                    IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR
C                             PIVOT ELEMENT AT ANY ELIMINATION STEP
C                             EQUAL TO 0,
C                    IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-
C                             CANCE INDICATED AT ELIMINATION STEP K+1,
C                             WHERE PIVOT ELEMENT WAS LESS THAN OR
C                             EQUAL TO THE INTERNAL TOLERANCE EPS TIMES
```

```
C                        ABSOLUTELY GREATEST MAIN DIAGONAL
C                        ELEMENT OF MATRIX A.
C        AUX    - AN AUXILIARY STORAGE ARRAY WITH DIMENSION M-1.
C
C     REMARKS
C        UPPER TRIANGULAR PART OF MATRIX A IS ASSUMED TO BE STORED
C        COLUMNWISE IN M*(M+1)/2 SUCCESSIVE STORAGE LOCATIONS, RIGHT
C        HAND SIDE MATRIX R COLUMNWISE IN N*M SUCCESSIVE STORAGE
C        LOCATIONS. ON RETURN SOLUTION MATRIX R IS STORED COLUMNWISE
C        TOO.
C        THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS
C        GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS
C        ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN -
C        INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL
C        SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE
C        INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS
C        GIVEN IN CASE M=1.
C        ERROR PARAMETER IER=-1 DOES NOT NECESSARILY MEAN THAT
C        MATRIX A IS SINGULAR, AS ONLY MAIN DIAGONAL ELEMENTS
C        ARE USED AS PIVOT ELEMENTS. POSSIBLY SUBROUTINE GELG (WHICH
C        WORKS WITH TOTAL PIVOTING) WOULD BE ABLE TO FIND A SOLUTION.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     METHOD
C        SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH
C        PIVOTING IN MAIN DIAGONAL, IN ORDER TO PRESERVE
C        SYMMETRY IN REMAINING COEFFICIENT MATRICES.
C
C     ..................................................................
C
```

```
C
C
      DIMENSION A(NCOEF),R(M  ),AUX(MAUX)
      IF (M) 240,240,10
C
C     SEARCH FOR GREATEST MAIN DIAGONAL ELEMENT
   10 IER=0
      PIV=0.
      L=0
      DO 30 K=1,M
      L=L+K
      TB=ABS(A(L))
      IF (TB-PIV) 30,30,20
   20 PIV=TB
      I=L
      J=K
   30 CONTINUE
      TOL=EPS*PIV
C     MAIN DIAGONAL ELEMENT A(I)=A(J,J) IS FIRST PIVOT ELEMENT.
C     PIV CONTAINS THE ABSOLUTE VALUE OF A(I).
C
C
C     START ELIMINATION LOOP
      LST=0
      NM=N*M
      LEND=M-1
      DO 180 K=1,M
C
C     TEST ON USEFULNESS OF SYMMETRIC ALGORITHM
      IF (PIV) 240,240,40
   40 IF (IER) 70,50,70
   50 IF (PIV-TOL) 60,60,70
```

```
   60 IER=K-1
   70 LT=J-K
      LST=LST+K
C
C     PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R
      PIVI=1./A(I)
      DO 80 L=K,NM,M
      LL=L+LT
      TB=PIVI*R(LL)
      R(LL)=R(L)
   80 R(L)=TB
C
C     IS ELIMINATION TERMINATED
      IF (K-M) 90,190,190
C
C     ROW AND COLUMN INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A.
C     ELEMENTS OF PIVOT COLUMN ARE SAVED IN AUXILIARY VECTOR AUX.
   90 LR=LST+(LT*(K+J-1))/2
      LL=LR
      L=LST
      DO 140 II=K,LEND
      L=L+II
      LL=LL+1
      IF (L-LR) 120,100,110
  100 A(LL)=A(LST)
      TB=A(L)
      GO TO 130
  110 LL=L+LT
  120 TB=A(LL)
      A(LL)=A(L)
  130 AUX(II)=TB
  140 A(L)=PIVI*TB
```

```
C
C     SAVE COLUMN INTERCHANGE INFORMATION
      A(LST)=LT
C
C     ELEMENT REDUCTION AND SEARCH FOR NEXT PIVOT
      PIV=0.
      LLST=LST
      LT=0
      DO 180 II=K,LEND
      PIVI=-AUX(II)
      LL=LLST
      LT=LT+1
      DO 150 LLD=II,LEND
      LL=LL+LLD
      L=LL+LT
  150 A(L)=A(L)+PIVI*A(LL)
      LLST=LLST+II
      LR=LLST+LT
      TB=ABS(A(LR))
      IF (TB-PIV) 170,170,160
  160 PIV=TB
      I=LR
      J=II+1
  170 DO 180 LR=K,NM,M
      LL=LR+LT
  180 R(LL)=R(LL)+PIVI*R(LR)
C
C
C     BACK SUBSTITUTION AND BACK INTERCHANGE
  190 IF (LEND) 240,230,200
  200 II=M
      DO 220 I=2,M
```

```
      LST=LST-II
      II=II-1
      L=A(LST)+.5
      DO 220 J=II,NM,M
      TB=R(J)
      LL=J
      K=LST
      DO 210 LT=II,LEND
      LL=LL+1
      K=K+LT
  210 TB=TB-A(K)*R(LL)
      K=J+L
      R(J)=R(K)
  220 R(K)=TB
  230 RETURN
C
C
C     ERROR RETURN
  240 IER=-1
      RETURN
      END
```

```
      SUBROUTINE TSP (D,E,W,WN,CPEN,RPEN,IP,JP,LINK,NPOINT,BEST,ITINR)
C
C  SOLVES NONTRIVIAL TRAVELING SALESMAN PROBLEMS,RETURNING MINIMUM
C  DISTANCE SOLUTION AND OPTIMAL ROUTE.
C
      COMMON/TSP NL/NTSPS,NRTE
      COMMON/ALL/M,N,K,ITER,NOUT
      DIMENSION D(M,M,N),E(M,M),W(N),WN(N),CPEN(M),RPEN(M),
     1RET(4),IRET(4),JRET(4),TEST(10)
      INTEGER*2 LINK(2,N),NPOINT(M),ITINR(N,10),IP(M),JP(M)
      NRTE=0
      NTSPS=0
      ISW=0
      KSW=0
      BIG=10.**20
      DELETE=10.**10
      BEST=BIG
      MM=M-NOUT
      NN=N-NOUT
      DO 20 I=1,4
   20 RET(I)=-4.
C
C  PRESERVE INITIAL DISTANCE MATRIX FOR RETURN TO MAIN AND FOR POSSIBLE
C  USE ON RETURN TO NODE LEVEL ONE.
C
      DO 40 I=1,M
      DO 30 J=1,M
   30 E(I,J)=D(I,J,1)
   40 CONTINUE
      K=1
   50 CALL REDUCE (D,H,NPOINT)
C
```

```
C   DETERMINE LOWER BOUND AT NODE LEVEL K.
C
      IF (K.NE.1) GO TO 60
      W(K)=H
      GO TO 70
   60 W(K)=W(K-1)+H
   70 CONTINUE
      IF (ISW.EQ.0) GO TO 80
C
C   COMPARE BEST LOWER BOUND WITH BEST FEASIBLE SOLUTION OBTAINED
C   THUS FAR.
C
      IF (W(K).GT.BEST) GO TO 400
   80 CALL PENALT (D,IP,JP,RPEN,CPEN,NPOINT,PENMAX)
      IR=IP(K)
      IC=JP(K)
      CALL NULINK (D,IR,IC,LINK)
C
C   DETERMINE LOWER BOUND IF LINK(IR,IC) IS NOT FORMED.
C
      WN(K)=W(K)+PENMAX
      IF (KSW.EQ.1) WN(K)=BIG
      KSW=0
      K=K+1
      IF (K.GT.NN) GO TO 110
C
C   FORM APPROPRIATE DISTANCE MATRIX FOR NEXT NODE LEVEL.
C
      DO 100 I=1,M
      DO 90 J=1,M
      D(I,J,K)=D(I,J,K-1)
   90 IF (I.EQ.IP(K-1).OR.J.EQ.JP(K-1)) D(I,J,K)=BIG
```

```
  100 CONTINUE
      IA=IP(K-1)
      JA=JP(K-1)
      D(JA,IA,K)=BIG
      IF (K.LE.NN) GO TO 50
C
C  FIND LINK NECESSARY TO MAKE SOLUTION FEASIBLE.
C
  110 DO 130 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 130
      DO 120 J=1,NN
      IF (I.EQ.IP(J)) GO TO 130
  120 CONTINUE
      IB=I
      GO TO 140
  130 CONTINUE
  140 DO 160 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 160
      DO 150 I=1,NN
      IF (J.EQ.JP(I)) GO TO 160
  150 CONTINUE
      JB=J
      GO TO 170
  160 CONTINUE
  170 IP(K)=IB
      JP(K)=JB
      NRTE=NRTE+1
      DO 180 I=1,MM
      IF (IP(I).NE.M) GO TO 180
      INDIC=JP(I)
      ITINR(1,NRTE)=INDIC
      GO TO 190
```

```
  180 CONTINUE
  190 DO 210 K=2,NN
      DO 200 I=1,MM
      IF (IP(I).NE.INDIC) GO TO 200
      INDIC=JP(I)
      ITINR(K,NRTE)=INDIC
      GO TO 210
  200 CONTINUE
  210 CONTINUE
      IF (NRTE.LE.1) GO TO 260
      NRT=NRTE-1
      DO 250 J=1,NRT
      NTEMP=NN
      DO 220 I=1,NN
      IF (ITINR(I,NRTE).NE.ITINR(NTEMP,NRTE-J)) GO TO 230
  220 NTEMP=NTEMP-1
      GO TO 430
  230 DO 240 I=1,NN
      IF (ITINR(I,NRTE).NE.ITINR(I,NRTE-J)) GO TO 250
  240 CONTINUE
      GO TO 430
  250 CONTINUE
C
C COMPARE CURRENT FEASIBLE SOLUTION WITH BEST SOLUTION OBTAINED
C THUS FAR.
C
  260 TEST(NRTE)=W(NN)
      IF (TEST(NRTE).LT.BEST) BEST=TEST(NRTE)
      ISW=ISW+1
      IF (ISW.EQ.1) GO TO 270
      K=MM
      GO TO 400
```

```
C
C  LOOK FOR UNBRANCHED NODES WITH LOWER BOUND LESS THAN BEST FEASIBLE
C  SOLUTION OBTAINED THUS FAR.
C
  270 DO 280 K=1,NN
      IF (WN(MM-K).LT.BEST) GO TO 290
  280 CONTINUE
      GO TO 430
  290 K=MM-K
  300 CONTINUE
      IT=IP(K)
      JT=JP(K)
C
C  RETURN TO NODE LEVEL K.
C
      IF (K.NE.1) GO TO 350
      DO 320 J=1,M
      DO 310 I=1,M
  310 D(I,J,1)=E(I,J)
  320 CONTINUE
C
C  RETAIN ORIGINAL ELEMENT IN DISTANCE MATRIX FOR PROPER RETURN TO MAIN.
C
      DO 330 I=1,4
      IF (RET(I).GE.0.) GO TO 330
      RET(I)=E(IT,JT)
      IRET(I)=IT
      JRET(I)=JT
      GO TO 340
  330 CONTINUE
  340 NTSPS=0
C
```

```
C  UPDATE DISTANCE MATRIX BY INSERTING INFINITY IN D(IT,JT) TO PREVENT
C  LINK PREVIOUSLY SELECTED AT THIS NODE LEVEL FROM BEING SELECTED
C  AGAIN.
C
      D(IT,JT,1)=BIG
      E(IT,JT)=BIG
      KSW=1
      GO TO 50
  350 D(IT,JT,K)=BIG
      KK=K-1
      NTSPS=0
C
C  UPDATE MATRIX LINK TO INCLUDE ONLY LINKS SELECTED PRIOR TO
C  NODE LEVEL K.
C
      DO 370 I=1,M
      DO 360 J=1,M
  360 D(I,J,K)=D(I,J,KK)
  370 CONTINUE
      DO 390 I=1,KK
      IR=IP(I)
      IC=JP(I)
      CALL NULINK (D,IR,IC,LINK)
      D(IC,IR,K)=BIG
      DO 380 J=1,M
      D(IR,J,K)=BIG
  380 D(J,IC,K)=BIG
  390 CONTINUE
C
C  UPDATE DISTANCE MATRIX BY INSERTING INFINITY IN D(IT,JT)
C  TO PREVENT LINK PREVIOUSLY SELECTED AT THIS NODE LEVEL FROM BEING
C  SELECTED AGAIN.
```

```
C
      D(IT,JT,K)=BIG
      KSW=1
      GO TO 50
  400 DO 410 I=1,N
      IF ((K-I).EQ.0) GO TO 430
      IF (WN(K-I).LT.BEST) GO TO 420
  410 CONTINUE
      GO TO 430
  420 K=K-I
      GO TO 300
  430 CONTINUE
      DO 440 I=1,10
      IF (TEST(I).EQ.BEST) GO TO 450
  440 CONTINUE
  450 CONTINUE
      NRTE=I
C
C  RETURN ORIGINAL DISTANCE MATRIX.
C
      DO 470 I=1,M
      D(I,I,1)=E(I,I)
      IF (I.EQ.M) GO TO 470
      II=I+1
      DO 460 J=II,M
      D(I,J,1)=E(I,J)
  460 D(J,I,1)=D(I,J,1)
  470 CONTINUE
      DO 480 I=1,4
      IF (RET(I).LT.0.) GO TO 490
      L1=IRET(I)
      L2=JRET(I)
```

```
      D(L1,L2,1)=RET(I)                                                 TSP  225
      D(L2,L1,1)=RET(I)                                                 TSP  226
      E(L1,L2)=RET(I)                                                   TSP  227
  480 E(L2,L1)=RET(I)                                                   TSP  228
  490 CONTINUE                                                          TSP  229
      RETURN                                                            TSP  230
C                                                                       TSP  231
      END                                                               TSP  232
```

```
      SUBROUTINE REDUCE (D,H,NPOINT)
C
C
      COMMON/ALL/M,N,K,ITER,NOUT
      DIMENSION D(M,M,N)
      INTEGER*2 NPOINT(M)
       REAL MIN
C
C   FIND MINIMUM ELEMENT OF EACH ROW AND SUBTRACT THAT ELEMENT FROM
C   EACH ELEMENT IN THAT ROW.
C
      DELETE=10.**10
      R=0.
      DO 50 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 50
      MIN=D(I,1,K)
      IF (NPOINT(1).NE.1) GO TO 20
      DO 10 JJ=2,M
      IF (NPOINT(JJ).EQ.1) GO TO 10
      IF (JJ.EQ.I) GO TO 10
      MIN=D(I,JJ,K)
      GO TO 20
   10 CONTINUE
   20 DO 30 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 30
      IF (MIN.GE.D(I,J,K)) MIN=D(I,J,K)
   30 CONTINUE
      IF (MIN.GE.DELETE) GO TO 50
      IF (MIN.EQ.0.) GO TO 50
      R=R+MIN
      DO 40 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 40
```

```
      D(I,J,K)=D(I,J,K)-MIN
   40 CONTINUE
   50 CONTINUE
C
C  FIND MINIMUM ELEMENT IN EACH COLUMN AND SUBTRACT THAT ELEMENT FROM
C  ALL ELEMENTS IN THAT COLUMN.
C
      C=0.
      DO 100 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 100
      MIN=D(1,J,K)
      IF (NPOINT(1).NE.1) GO TO 70
      DO 60 II=2,M
      IF (NPOINT(II).EQ.1) GO TO 60
      IF (II.EQ.J) GO TO 60
      MIN=D(II,J,K)
      GO TO 70
   60 CONTINUE
   70 DO 80 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 80
      IF (MIN.GE.D(I,J,K)) MIN=D(I,J,K)
   80 CONTINUE
      IF (MIN.GE.DELETE) GO TO 100
      IF (MIN.EQ.0.) GO TO 100
      C=C+MIN
      DO 90 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 90
      D(I,J,K)=D(I,J,K)-MIN
   90 CONTINUE
  100 CONTINUE
      H=R+C
      RETURN
```

TSP 297

END

```
      SUBROUTINE PENALT (D,IP,JP,RPEN,CPEN,NPOINT,PENMAX)
C
C
      COMMON/ALL/M,N,K,ITER,NOUT
      DIMENSION D(M,M,N),CPEN(M),RPEN(M)
      INTEGER*2 IP(M),JP(M),NPOINT(M)
      REAL MIN
C
C  FIND SECOND SMALLEST ELEMENT IN EACH ROW.
C
      DO 30 I=1,M
      MIN=10.**20
      IF (NPOINT(I).EQ.1) GO TO 30
      NCOUNT=0
      DO 10 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 10
      IF (D(I,J,K).LT.MIN.AND.D(I,J,K).NE.0.) MIN=D(I,J,K)
      IF (D(I,J,K).EQ.0.) NCOUNT=NCOUNT+1
      IF (NCOUNT.EQ.2) GO TO 20
      RPEN(I)=MIN
   10 CONTINUE
      GO TO 30
   20 RPEN(I)=0.
   30 CONTINUE
C
C  FIND SECOND SMALLEST ELEMENT IN EACH COLUMN.
C  FIND MAXIMUN PENALTY FOR NOT TAKING A BRANCH.
C
      DO 60 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 60
      MIN=10.**20
      NCOUNT=0
```

```
      DO 40 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 40
      IF (D(I,J,K).LT.MIN.AND.D(I,J,K).NE.0.) MIN=D(I,J,K)
      IF (D(I,J,K).EQ.0.) NCOUNT=NCOUNT+1
      IF (NCOUNT.EQ.2) GO TO 50
      CPEN(J)=MIN
   40 CONTINUE
      GO TO 60
   50 CPEN(J)=0.
   60 CONTINUE
C
C   FIND MAXIMUM PENALTY FOR NOT TAKING A BRANCH.
C
      PENMAX=0.
      DO 90 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 90
      DO 80 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 80
      IF (D(I,J,K).NE.0.) GO TO 80
      A=RPEN(I)+CPEN(J)
      IF (A.GT.PENMAX) GO TO 70
      GO TO 80
   70 PENMAX=A
      IP(K)=I
      JP(K)=J
   80 CONTINUE
   90 CONTINUE
      IF (PENMAX.GT.0.) GO TO 150
      DO 110 I=1,M
      IF (NPOINT(I).EQ.1) GO TO 110
      DO 100 J=1,M
      IF (NPOINT(J).EQ.1) GO TO 100
```

```
      IF (D(I,J,K).EQ.0.) GO TO 120
  100 CONTINUE
  110 CONTINUE
  120 IP(K)=I
      JP(K)=J
      IF (I.LE.M.OR.J.LE.M) GO TO 150
      KK=K-1
      DO 140 L=1,M
      IF (NPOINT(L).NE.1) GO TO 140
      IP(K)=L
      JP(K)=L
      DO 130 LL=1,KK
      IF (IP(LL).EQ.L) GO TO 140
  130 CONTINUE
      GO TO 150
  140 CONTINUE
  150 RETURN
      END
```

```
      SUBROUTINE NULINK (D,IR,IC,LINK)
C
C
      COMMON/TSP NL/NTSPS,NRTE
      COMMON/ALL/M,N,K,ITER,NOUT
      DIMENSION D(M,M,N)
      INTEGER*2 LINK(2,N)
C
C   ADD ON TO LINKS ALREADY FORMED IN MATRIX LINK.
C
      NR=0
      NC=0
      BIG=10.**20
      IF (NTSPS) 90,90,10
   10 DO 50 I=1,NTSPS
      IF (IC-LINK(1,I)) 30,20,30
   20 NR=I
   30 IF (IR-LINK(2,I)) 50,40,50
   40 NC=I
   50 CONTINUE
C
C   FORM NEW PARTIAL SOLUTIONS IN MATRIX LINK.
C
      IF ((NR+NC).EQ.0) GO TO 90
      IF (NR.GT.0.AND.NC.GT.0) GO TO 60
      IF (NR.GT.0.AND.NC.EQ.0) GO TO 70
      IF (NR.EQ.0.AND.NC.GT.0) GO TO 80
   60 LINK(1,NR)=IR
      LINK(2,NC)=IC
      GO TO 100
   70 LINK(1,NR)=IR
      JA=LINK(2,NR)
```

```
      D(JA,IR,K)=BIG
      GO TO 100
   80 LINK(2,NC)=IC
      IA=LINK(1,NC)
      D(IC,IA,K)=BIG
      GO TO 100
   90 NTSPS=NTSPS+1
      LINK(1,NTSPS)=IR
      LINK(2,NTSPS)=IC
      D(IR,IC,K)=BIG
      D(IC,IR,K)=BIG
      GO TO 140
C
C  REDUCE MATRIX LINK IF IT CONTAINS IR (FROM) ROW 1,COLUMN J,AND
C  IC (TO)IN ROW 2,COLUMN I, I NOT EQUAL TO J.
C
  100 DO 130 I=1,NTSPS
      DO 120 J=1,NTSPS
      IF (LINK(2,I).EQ.IC.AND.LINK(1,J).EQ.IR.AND.I.NE.J) GO TO 110
      GO TO 120
  110 LINK(2,I)=LINK(2,J)
      IU=LINK(1,I)
      IV=LINK(2,I)
      D(IV,IU,K)=BIG
      IF (I.LT.J) GO TO 150
      LINK(1,J)=LINK(1,I)
      LINK(2,J)=LINK(2,I)
      GO TO 150
  120 CONTINUE
  130 CONTINUE
  140 RETURN
  150 NTSPS=NTSPS-1
```

```
      RETURN                                                          TSP  444
      END                                                             TSP  445
```

VITA

Robert Currie Burness was born on February 22, 1947 in Salem, Massachusetts. He graduated from Scotch Plains-Fanwood High School, Scotch Plains, New Jersey, in 1965.

In June of 1969 Mr. Burness received a B.S. degree in Economics from Hampden-Sydney College. After serving two years as a high school math teacher in Appomattox, Virginia, he entered the graduate school at Virginia Polytechnic Institute and State University. He is a member of AIIE and Alpha Pi Mu. Currently he is an instructor in the Department of Industrial Engineering and Operations Research on the Blacksburg campus.

Robert Currie Burness

CENTRAL FACILITIES LOCATION PROBLEMS
INVOLVING TRAVELING SALESMAN TOURS AND EXPECTED DISTANCES

by

Robert C. Burness

(ABSTRACT)

The problem of locating a single new facility relative to m existing facilities has been studied extensively under the assumption that trips are always made between the new facility and a single existing facility each time a trip occurs. A variation of the problem involves a traveling salesman who services m customers. Each time a trip occurs the salesman visits one or more customers during the trip. Thus, 2^m-1 different itineraries can be formed, each occuring with a given probability. The new facility, which is the starting and ending point for each itinerary, is to be located such that the expected distance traveled per unit time is minimized. Computational experience in solving the problem is presented for both Euclidean and rectilinear distance measures.