

THE RECTILINEAR DISTANCE  
LOCATION-ALLOCATION PROBLEM

A THESIS

Presented to

The Faculty of the Division of Graduate  
Studies

By

Anis D. Serali

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in Operations Research

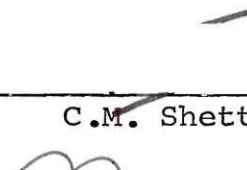
Georgia Institute of Technology

March 1976


28-3  
T-254

THE RECTILINEAR DISTANCE  
LOCATION-ALLOCATION PROBLEM

Approved :

  
\_\_\_\_\_  
C.M. Shetty, Thesis Advisor

  
\_\_\_\_\_  
J.J. Jarvis

  
\_\_\_\_\_  
J.A. White

Date approved by Thesis Advisor : Jul. 26, 1976

## ACKNOWLEDGMENTS

I wish to express my heartfelt gratitude to Dr. C.M. Shetty for all the assistance extended to me in the preparation of this thesis, and throughout my educational experience at the School of Industrial and Systems Engineering of the Georgia Institute of Technology. His constant guidance, infinite patience and deep insight were indispensable for the completion of this work, and I will always treasure the bond of friendship existing between us.

I also wish to thank Dr. J.A. White for introducing me to the wide spectrum of topics in the area of Location Theory, and both Dr. J.A. White and Dr. J.J. Jarvis for reading the dissertation and providing helpful criticism.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	ii
LIST OF TABLES	v
LIST OF ILLUSTRATIONS	vi
Chapter	
I. INTRODUCTION AND LITERATURE SURVEY . . . . .	1
Introduction	
Related Problems and Applications	
The Problem Statement	
II. PROPERTIES OF THE RECTILINEAR DISTANCE LOCATION-ALLOCATION PROBLEM . . . . .	20
Introduction	
Some Preliminary Properties : Extreme Point and Global Optimality	
III. AN ALGORITHM TO SOLVE THE RECTILINEAR DISTANCE LOCATION PROBLEM . . . . .	28
A Primal Simplex Based Algorithm for the Problem RDLP	
Properties of the Problem RDLP	
IV. THE CUTTING PLANE ALGORITHM . . . . .	41
The Cutting Plane Algorithm for Bilinear Programming Problems	
The Cutting Plane Algorithm for Problem RDLAP	
Algorithm for Cut Generation	
Determination of a Weak Pseudo Global Minimum	
Determination of a Good Starting Solution	

Chapter	Page
IV. THE CUTTING PLANE ALGORITHM (Continued)	
Solution of the Parametric Problem	
Determination of a Feasible Extreme Point	
Statement of the Complete Algorithm	
Convergence of the Algorithm	
Illustrative Examples	
The Unrestricted Capacity Problem	
V. COMPUTATIONAL RESULTS AND CONCLUSIONS . . . . .	79
Computational Results	
Summary and Conclusions	
Recommendations for Further Research	
BIBLIOGRAPHY	96

## LIST OF TABLES

Table	Page
1. Updated Starting Tableau for Problem LP - (x) . . . . .	33
2. Simplex Tableau for Problem LP-(x) . . . . .	37
3. Locations of Existing Facilities : Cooper's Problems . . . . .	81
4. Computational Results for Cooper's Problems . . . . .	82
5. Computational Results for $m = 2$ . . . . .	85
6. Computational Results for $n = 7$ . . . . .	85
7. Computational Results for an Efficient Transportation Code . . . . .	87
8. Computational Results for an Out of Kilter Code . . . . .	87
9. Computational Results for Large Size Rectilinear Distance Location-Allocation Problems . . . . .	89

## LIST OF ILLUSTRATIONS

Figure	Page
1. Locations of New and Existing Facilities for Example Problem 4.10.1 . . . . .	68
2. Locations of New and Existing Facilities for Example Problem 4.10.2 . . . . .	74

## CHAPTER I

## INTRODUCTION AND LITERATURE SURVEY

1.1 Introduction

Facility layout and location problems have interested researchers from a wide spectrum of disciplines ever since the seventeenth century (14). This diversified interest has resulted in a vast body of literature on the subject, consisting of different aspects of the problem and different approaches for their solution. The earliest known facilities location problem was formulated by Fermat, a seventeenth century mathematician. He posed the following problem: "Given three points in a plane, find a fourth point such that the sum of its distances to the three given points is a minimum." In his historical sketch, Kuhn (14) relates that before 1640 Torricelli had solved the problem. The optimum point, aptly named the Torricelli point, is to be found at the intersection of the circles which circumscribe the equilateral triangles formed on the sides of and outside the triangles of the three given points. Fermat's problem was later generalized to

include  $n$  existing facilities and was referred to as the Steiner-Weber problem. But, it was not until 1962 that Kuhn and Kuenne (15) showed that the problem could be solved by an iterative method for the location of the new facility. This represented the first practical numerical solution to the problem.

Since then, we have come a long way in the study of facilities location problems. In classifying problems in this area, Francis and White (6) mention six major elements that need to be considered, i.e., new facility characteristics, existing facility locations, new and existing facility interactions, solution space characteristics, distance measures and the objective under consideration. In light of this classification, location-allocation problems are categorized as having the following characteristics.

Location-allocation problems involve the determination of the number of new facilities as well as their locations, in addition to the interaction between new and existing facilities. The configuration of new and existing facilities is assumed to be concentrated at points in two dimensional space. Existing facility locations are static, deterministic and constitute one of the parameters of the problem. The objective is to minimize

total cost. A mathematical formulation of the location - allocation problem is as follows:

$$\begin{aligned}
 \text{LAP-1} \quad & \text{Minimize} \quad \phi = \sum_{i=1}^m \sum_{j=1}^n w_{ij} d(X_i, P_j) + g(m) \\
 & \text{subject to} \quad \sum_{i=1}^m w_{ij} = r_j \quad j=1, \dots, n \\
 & \quad \quad \quad \sum_{j=1}^n w_{ij} \leq c_i \quad i=1, \dots, m \\
 & \quad \quad \quad X_i \in L \quad i=1, \dots, m
 \end{aligned}$$

where,  $\phi$  : total cost

$m$  : number of new facilities to be located

$n$  : number of existing facilities

$X_i$  : point location of a new facility  $i$

$P_j$  : point location of an existing facility  $j$

$d(X_i, P_j)$  : distance measure between the points  $X_i$  and  $P_j$

$w_{ij}$  : units of interaction between new facility  $i$  and existing facility  $j$ .

$r_j$  : units of requirement at existing facility  $j$

$c_i$  : units of capacity at new facility  $i$

$L$  : solution space characteristics for the location of new facilities

$g(m)$  : cost of providing  $m$  new facilities.

The decision variables in the problem LAP-1 are (1)  $m$ , (2)  $X_i$ ,  $i=1, \dots, m$  and (3)  $w_{ij}$ ,  $i=1, \dots, m$  and  $j=1, \dots, n$ . Depending on which combination of these three sets of variables is fixed, we can formulate special cases of the problem LAP-1. In the next section, we will consider these special problems, their solution procedures and applications in real life situations.

### 1.2 Related Problems and Applications

We will first deal with the problem in which the variables  $m$  and  $w_{ij}$  are fixed. The implication of this restriction is that the interaction between new and existing facilities is location independent, and functions as a parameter of the problem, fixed at static and deterministic values. The mathematical model simplifies to the following:

Given the values of  $\bar{w}_{ij}$ ,

$$\begin{array}{ll} \text{LP-1} & \text{Minimize: } \phi_1 = \sum_{i=1}^m \sum_{j=1}^n \bar{w}_{ij} d(X_i, P_j) \\ & \text{subject to: } X_i \in L \quad i=1, \dots, m \end{array}$$

The above problem can be further subdivided by considering different expressions for  $d(X_i, P_j)$  depending on the

appropriate distance measure, and by defining suitable sets  $L$  representing the actual solution space characteristics. The distance measures most frequently encountered are the rectilinear and Euclidean distances. Examples of other distance measures include the squared Euclidean distance and the great circle distance. The solution space can be characterized according to whether it is single or multidimensional, discrete or continuous, constrained or unconstrained. Sometimes, the problem LP-1 is further generalized to include terms corresponding to fixed interaction between two new facilities. In this case, the objective function becomes:

$$\phi_2 = \sum_{i=1}^m \sum_{j=1}^n \bar{w}_{ij} d(X_i, P_j) + \sum_{1 \leq i < k \leq m} \bar{v}_{ik} d(X_i, X_k)$$

where,  $\bar{w}_{ij}$  and  $\bar{v}_{ik}$  are fixed at known values. We will now discuss solution procedures and applications for some of the more important problems under this classification. The first half of the discussion will be devoted to problems in which the set  $L$  is the two dimensional space  $R_2$ , and the latter half to problems in which  $L$  is a discrete subset of  $R_2$ .

The rectilinear distance location problem combines the property of being a very appropriate distance measure for a large number of location problems, and the property of being

very simple to treat analytically. Urban location analyses and warehouse and office designs are some examples where rectilinear distances are most often applicable. The ease of solution arises as a result of a suitable substitution to eliminate the absolute value signs in the expression for  $d(X_i, P_j)$  shown below.

$$d(X_i, P_j) = |x_i - d_j| + |y_i - e_j| = (x_{ij}^+ + x_{ij}^-) + (y_{ij}^+ + y_{ij}^-)$$

where,  $x_{ij}^+ - x_{ij}^- = d_j - x_i$ ,  $x_{ij}^+ \cdot x_{ij}^- = 0$ ,

$$y_{ij}^+ - y_{ij}^- = e_j - y_i, \quad y_{ij}^+ \cdot y_{ij}^- = 0,$$

and  $x_{ij}^+, x_{ij}^-, y_{ij}^+, y_{ij}^-, x_i, y_i \geq 0$ .

It can be proved, that the solution to the problem remains the same even when the constraints of the form  $x_{ij}^+ \cdot x_{ij}^- = 0$  are completely ignored. Thus, the problem reduces to a linear programming problem, whose dual can be solved with greater efficiency, using the bounded simplex algorithm. In Chapter III we will develop a primal-simplex based algorithm to solve the problem LP-1 with rectilinear distances, in the continuous space  $R_2$ . When the objective function is of the

type  $\phi_2$ , the linear programming approach is still applicable, although even less efficient. Eyster, et al (5) have developed a hyperboloid approximation procedure (HAP) which approximates the objective function at the points where its derivative is not defined. Then, use is made of a gradient procedure to yield a solution which is improved upon using fixed point iteration methods. At every stage, the approximated function is improved until a suitable stopping criterion is met. The approximation used is  $|x_i - d_j| = ((x_i - d_j)^2 + e)^{\frac{1}{2}}$  where  $e$  tends to zero. This method is applicable to problems with objective functions of the type  $\phi_1$  and  $\phi_2$ . It is found to be efficient although no proof of convergence is available.

For the Euclidean distance location problem,

$$d(X_i, P_j) = ((x_i - d_j)^2 + (y_i - e_j)^2)^{\frac{1}{2}}.$$

The HAP procedure devised by Eyster, et al is at present the best available algorithm to solve this problem. The approximation involved gives rise to the following expression:

$$d(X_i, P_j) = ((x_i - d_j)^2 + (y_i - e_j)^2 + e)^{\frac{1}{2}}.$$

Hence, the derivative of  $d(X_i, P_j)$  is defined for every

$(x_i, y_i)$  in  $R_2$ . The algorithm used is essentially the same as the one outlined for the rectilinear distance problem. Another procedure is the modification due to Kuhn (14) based on the definition of the derivative when  $(x_i, y_i) = (d_j, e_j)$ . This procedure too involves using fixed point iteration methods to obtain the optimum locations, but unlike the HAP procedure, a proof of convergence is available.

Discrete plant location problems differ from their continuous counterparts in that the locations of the potential plant sites are finite. Such problems occur frequently in the context of locating industrial plants. Efroymsen and Ray (4) formulate the discrete plant location problem as a mixed integer programming problem and initially solve it as a linear programming problem, ignoring the integer restrictions. Obviously if the solution is integer valued, the problem is solved. If not, a branch and bound approach is employed to obtain integer solutions. In addition to the Efroymsen and Ray procedure, a number of other exact and heuristic procedures have been developed for a variety of plant location problems. References (11), (12) and (22) relate to some of the research done in this area.

Another plant location problem which is frequently encountered is the covering problem. The problem of deter-

minimizing the number and locations of new facilities which are at a distance of not greater than say, 10 miles or 15 minutes from all existing facilities can be formulated as a covering problem. Such problems are encountered when locating public schools, police stations, hospitals, post offices, etc. Generally speaking there are four broad approaches reported in the literature for solving covering problems. The first of these is an implicit enumeration approach such as the branch and bound (17). The second approach uses cutting planes and solves iteratively a number of linear programming problems (1). The third approach is to employ reduction techniques (24), and the fourth deals with heuristic methods (10) for solving the covering problem.

We now consider the problem LAP-1 with variables  $m$  and  $X_i$  fixed at known values. The problem reduces to:

$$\begin{array}{ll}
 \text{LP-2} & \text{Minimize: } \phi = \sum_{i=1}^m \sum_{j=1}^n k_{ij} w_{ij} \\
 & \text{subject to: } \sum_{j=1}^n w_{ij} \leq c_i \quad i=1, \dots, m \\
 & \sum_{i=1}^m w_{ij} = r_j \quad j=1, \dots, n \\
 & w_{ij} \geq 0 \quad \text{for all } i \text{ and } j
 \end{array}$$

where,  $k_{ij} = d(\bar{X}_i, \bar{P}_j) > 0$  and is fixed at a known value. The problem LP-2 is just a transportation problem for which very efficient algorithms and computer codes (23) are available. If the constraints  $\sum_{j=1}^n w_{ij} \leq c_i$ ,  $i=1, \dots, m$  are deleted or are non-binding, the problem can be solved very easily.

We will now discuss the problem LAP-1 with regards to some of the currently available solution procedures to solve this problem, and its applications to real life problems. In the previous sections we considered the problem with variables  $m$  and  $w_{ij}$  fixed and the problem with variables  $m$  and  $X_i$  fixed. In the former case it is assumed a priori exactly which existing facilities interact with a particular new facility, and exactly how much flow takes place between existing facility  $j$  and new facility  $i$ . In the latter case, it is assumed a priori that the travel costs are explicitly fixed at known values. In actual practice, this is almost never the case, since the allocation of flow between facilities  $i$  and  $j$  depends on the relative location of these facilities, and similarly, the location of new facilities depends on the extent of interaction between new and existing facilities. It is precisely this concept that Cooper (2) employs in

a heuristic which iterates between problems LP-1 and LP-2 until it converges to a solution. We will show in Chapter IV that the above heuristic can easily converge to a point which is non-optimal.

Another heuristic proposed by Cooper (2) assumes that each existing facility can be supplied from only one new facility. This assumption is valid only if the problem LAP-1 is considered without the constraints

$$\sum_{j=1}^n w_{ij} \leq c_i \quad i=1, \dots, m.$$

It can be argued that the above problem is a more realistic one to solve since it provides additional information regarding the optimal production capacities of the new facilities to be located. But, it is obvious that the restricted capacity problem is a more general one, the unrestricted capacity problem being a special case with the  $c_i$ 's fixed at arbitrarily large values. The heuristic algorithm involves adjusting the allocations at each step of the iterative procedure by comparing the requirement at an existing facility to the surplus or deficit in the capacity of the new facility.

Cooper's exact algorithm (2) involves finding all the basic feasible solutions corresponding to the extreme points

of the problem LP-2 ( in the absence of degeneracy) and solving the location problem LP-1 for each one of these solutions. The optimum solution is obtained by selecting the best from among all the problems solved. A similar enumerative approach is to consider all combinations of ways in which  $m$  new facilities with the restriction that each existing facility is served by only one new facility. The number of such combination is given by the Stirling number of the second kind:

$$S(m,n) = \sum_{k=0}^m \frac{(-1)^k (m-k)^n}{k! (m-k)!}$$

Another approach to solve the location-allocation problem in continuous space was a branch and bound algorithm presented by Kuenne and Soland (13). The formulation of the problem differs slightly from the one mentioned under LAP-1 in the sense that a zero-one variable  $z_{ij}$  is introduced to indicate whether existing facility  $j$  is to be serviced by new facility  $i$ . Under the assumption mentioned above,  $\sum_i z_{ij} = 1$  for each  $j$ . The algorithm approaches optimality to within a specified level of accuracy.

An interesting approach to solve exactly the location-allocation problem with rectilinear distances was presented

by Morris (21). He identified the extreme points of the location part of the problem and formulated the problem LAP-1 using zero-one variables  $z_{ik}$  to indicate whether new facility  $i$  was assigned to the  $k^{\text{th}}$  extreme point location, and zero-one variables  $z_{ij}$  to indicate whether new facility  $i$  interacts with existing facility  $j$ . Morris too worked under the assumption that  $\sum_i z_{ij} = 1$  for all  $j$ . The approach used to solve the problem was similar to the one used by Efraymson and Ray (4), and the claim that the solutions were usually fully integer was backed by the computational experience provided by Re Velle and Swan (22).

The principal disadvantage of this formulation, if we choose to disregard the fact that solutions may not always be integer, is the dimensionality problem encountered. There are  $n^3+n^2-n$  variables and more importantly,  $n^3+1$  constraints. It is precisely this fact which limits  $n$  to be no greater than 15 when using the LP1108 code which can accommodate 4044 rows and 99000 columns. The advantage of this formulation lies in the fact that the model is computationally insensitive to the number of new facilities  $m$ , which occurs just once in the right hand side of one of the constraints. Thus, the problem of optimizing over  $m$  can be solved with great ease, using the basis inverse obtained from the previous optimal

solution.

The above problem, although formulated in continuous space yielded certain properties which restricted the locations of new facilities to a finite number of points, i.e., the extreme points of the location part of the problem. Hence, a similar approach can be used to solve location-allocation problems in discrete space. Another approach to solving problems of this type has been presented by Geoffrion (8). This algorithm utilizes the Bender's partitioning procedure to reduce the dimensionality problem encountered with mixed integer programming formulations mentioned above. It would be unfair to end the discussion on discrete space location-allocation problem formulations without mentioning the fact that a great deal of work has been done in this area to develop algorithms capable of achieving a high degree of efficiency in solving mixed integer programming problems. We refer the reader to the following list of references for further details (11, 12 and 22).

We will now address ourselves to the problem which we will be discussing in this dissertation. The rectilinear distance measure, as mentioned previously, has the property of being very appropriate for a large number of urban location problems. The location-allocation problem is a very realistic

problem which arises frequently enough to merit a great deal of attention in developing an exact solution procedure. The location-allocation problem with limitations on the capacities at new facilities is more general than its uncapacitated counterpart. Although the problem can be formulated as a mixed integer programming problem with a finite number of potential sites for locating new facilities we will formulate it in continuous space to avoid the problems mentioned previously, and attempt to solve it using an approach different from those tried already. The statement of the restricted capacity rectilinear distance location-allocation problem formulated in continuous space follows in the next section.

### 1.3 The Problem Statement

A mathematical model for the restricted capacity rectilinear distance location-allocation problem with continuous variables can be stated as follows :

$$\begin{array}{ll}
 \text{RDLAP-1} & \text{Minimize: } \sum_{i=1}^m \sum_{j=1}^n w_{ij} (|x_i - d_j| + |y_i - e_j|) \\
 & \text{subject to: } \sum_{j=1}^n w_{ij} \leq c_i \quad i=1, \dots, m \\
 & \sum_{i=1}^m w_{ij} = r_j \quad j=1, \dots, n \\
 & w_{ij} \geq 0 \quad \text{for all } i \text{ and } j.
 \end{array}$$

where,  $(d_j, e_j)$  is the coordinate of the  $j^{\text{th}}$  existing facility,  
 $(x_i, y_i)$  is the coordinate of the  $i^{\text{th}}$  new facility,  
 $m$  and  $n$  are the number of new and existing facilities resp.,  
 $c_i$  represents the capacity of the  $i^{\text{th}}$  new facility,  
 $r_j$  represents the requirement at the  $j^{\text{th}}$  existing facility,  
and  $w_{ij}$  represents the flow between new facility  $i$  and  
existing facility  $j$ .

The absolute value signs can be deleted using the  
following substitution:

$$d_j - x_i = x_{ij}^+ - x_{ij}^-$$

$$x_{ij}^+, x_{ij}^- \geq 0$$

We will assume, without loss of generality, that

$$x_i, y_i, d_j, e_j \geq 0 \quad \text{for all } i \text{ and } j.$$

This can be achieved through a simple transformation of axes.

It is readily seen that

$$|d_j - x_i| = x_{ij}^+ + x_{ij}^- \quad \text{if} \quad x_{ij}^+ \cdot x_{ij}^- = 0.$$

Using this substitution, RDLAP-1 simplifies to:

$$\text{RDLAP-2} \quad \text{Minimize: } \sum_{i=1}^m \sum_{j=1}^n w_{ij} (x_{ij}^+ + x_{ij}^- + y_{ij}^+ + y_{ij}^-)$$

$$\text{subject to: } x_{ij}^+ - x_{ij}^- = d_j - x_i \text{ for all } i \text{ and } j$$

$$y_{ij}^+ - y_{ij}^- = e_j - y_i \text{ for all } i \text{ and } j$$

$$\sum_{i=1}^m w_{ij} = r_j \quad j=1, \dots, n$$

$$\sum_{j=1}^n w_{ij} \leq c_i \quad i=1, \dots, m$$

$$x_{ij}^+ \cdot x_{ij}^- = 0 \text{ for all } i \text{ and } j$$

$$y_{ij}^+ \cdot y_{ij}^- = 0 \text{ for all } i \text{ and } j$$

$$x_i, y_i, x_{ij}^+, x_{ij}^-, y_{ij}^+, y_{ij}^-, w_{ij} \geq 0$$

for all  $i$  and  $j$ .

RDLAP-2 has  $5mn + 2m$  variables and  $2mn + m + n$  constraints in addition to the  $2mn$  constraints of the form  $x_{ij}^+ \cdot x_{ij}^- = 0$ , and the non-negativity constraints. For a problem of size  $m=3$  and  $n=7$ , we have 111 variables and 94

constraints, excluding the non-negativity constraints. We will show in Chapter III that the constraints of the form  $x_{ij}^+ \cdot x_{ij}^- = 0$  can be deleted without affecting the problem solution, and hence will not be included in the definition of the following constraint sets.

Notice that the constraints are separable in the location variables  $x_i, y_i, x_{ij}^+, x_{ij}^-, y_{ij}^+, y_{ij}^-$  and the allocation variables  $w_{ij}$ . For simplicity in presentation, and to show the relationship between RDLAP-2 and the bilinear programming problem (to be defined shortly), we will introduce the following sets.

$$Z = \left\{ z \equiv (x_{11}^+, \dots, x_{mn}^+, x_{11}^-, \dots, x_{mn}^-, y_{11}^+, \dots, y_{mn}^+, y_{11}^-, \dots, y_{mn}^-, x_1, \dots, x_m, y_1, \dots, y_m)^t : \begin{aligned} x_{ij}^+ - x_{ij}^- &= d_j - x_i, \\ y_{ij}^+ - y_{ij}^- &= e_j - y_i, \quad z \geq 0, \text{ for all } i \text{ and } j \end{aligned} \right\}$$

$$W = \left\{ w \equiv (w_{11}, \dots, w_{mn})^t : \begin{aligned} \sum_{j=1}^m w_{ij} &= r_j \quad j=1, \dots, n, \\ \sum_{j=1}^n w_{ij} &\leq c_i \quad i=1, \dots, m, \quad w_{ij} \geq 0 \text{ for all } \\ & i \text{ and } j \end{aligned} \right\}$$

Then, RDLAP-2 can be rewritten as :

RDLAP-3      Minimize:       $\phi(z,w) = z^t D w$

subject to:       $z \in Z$

$w \in W$

where,

$$D = \begin{bmatrix} I_{mn \times mn} \\ I_{mn \times mn} \\ I_{mn \times mn} \\ I_{mn \times mn} \\ O_{2m \times mn} \end{bmatrix}$$

The sets  $Z$  and  $W$  are polyhedral sets since all the constraints in each set are linear. Compare problem RDLAP-3 with the statement of the Bilinear Programming problem given below.

BLP-1      Minimize:       $\phi(z,w) = c^t z + d^t w + z^t D w$

subject to:       $z \in Z$

$w \in W$

where,  $Z$  and  $W$  are non-empty polyhedral sets.

It is clearly seen that the problem RDLAP-3 can be reduced to exactly the same form as BLP-1 by letting  $c=d=0$ .

## CHAPTER II

PROPERTIES OF THE RECTILINEAR DISTANCE  
LOCATION-ALLOCATION PROBLEM

This chapter serves to establish certain properties of the problem RDLAP-3 which will be used in subsequent chapters. Some of the results to follow are provided only to indicate the difficulties involved in solving the problem RDLAP-3.

2.1 Introduction

The first two properties discussed in this chapter deal with the nature of the objective function over the positive orthant  $R_n^+$ . Our objective in investigating these properties of quasiconcavity and quasiconvexity is to be able to use the following theorems of Martos to establish certain properties of the optimal solution to the problem RDLAP.

Theorem 2.1.1 : A continuous function  $f$  defined over a polytope  $L$  attains its minimum at an extreme point of  $L$  and all its convex polyhedral subsets if and only if it is quasiconcave on  $L$ .

Proof : See (19)

Theorem 2.1.2 : A continuous function  $f$  defined over a polytope  $L$  is such that each local minimum is also a global minimum on  $L$  and all its convex polyhedral subsets if and only if it is explicitly quasiconvex on  $L$ .

Proof : See (19)

In reference (2), the objective function has been shown to be neither convex nor concave. In this study we will investigate less stringent sufficiency conditions for global and extreme point optimality. For this purpose we will need the following definitions and theorems.

Definition 2.1.3 : A matrix  $C$  is positive(negative) subdefinite if  $y^t C y \leq 0$  ( $y^t C y > 0$ ) implies  $Cy \leq 0$  or  $Cy \geq 0$ . A quadratic form  $\phi(y) = \frac{1}{2} y^t C y$  is said to be positive (negative) subdefinite if  $C$  is positive (negative) subdefinite.

Theorem 2.1.4 : A quadratic form  $\phi(y) = \frac{1}{2} y^t C y$  is said to be quasiconvex (quasiconcave) on the non-negative orthant  $R^+$ , if and only if it is positive (negative) subdefinite.

Proof : See (18)

The problem RDLAP-3 can also be expressed in the quadratic form  $\frac{1}{2} y^t C y$  where  $y^t = (z^t, w^t)$  and



$(-1, 2, 0, 0, 0, 0, -1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)^t$ .

Therefore,  $\frac{1}{2}y^tCy = 1 > 0$ .

But,  $Cy$  is neither less than or equal to zero nor greater than or equal to zero. Hence,  $\phi(z, w) = z^tDw$  is not quasiconcave over the non-negative orthant  $R^+$ .

We have just shown, with the help of two counter-examples that  $\phi(z, w)$  is neither quasiconvex nor quasiconcave over  $R^+$ . We will now show that in spite of the fact that  $\phi(z, w)$  is not quasiconcave over  $R^+$ , the problem RDLAP has the property that the optimum will occur at an extreme point of its constraint set.

## 2.2 Some Preliminary Properties :

### Extreme Point and Global Optimality.

In this section, we will introduce the concept of extreme points for the problem RDLAP-3 defined over two polyhedral constraint sets  $Z$  and  $W$  (see Section 1.3). We will characterize a point which we shall call a local star minimum, and develop an algorithm to find such a point. And finally, we will introduce the concept of a "pseudo global minimum".

Theorem 2.2.1 :  $(\bar{z}, \bar{w})$  is an extreme point of

$$P = \{(z, w) : z \in Z \text{ and } w \in W\}$$

if and only if  $\bar{z}$  is an extreme point of  $Z$  and  $\bar{w}$  is an extreme point of  $W$ .

Proof : See (25)

Corollary 2.2.2 : Each adjacent extreme point of  $(\bar{z}, \bar{w}) \in P$  is either of the form  $(z^i, \bar{w})$  where  $z^i \in N(\bar{z})$ , or of the form  $(\bar{z}, w^i)$  where  $w^i \in N(\bar{w})$ , given that  $N(\bar{z})$  and  $N(\bar{w})$  are sets of adjacent extreme points of  $\bar{z}$  in  $Z$  and  $\bar{w}$  in  $W$  respectively.

Proof : See (25)

Definition 2.2.3 : The function  $\phi(z, w)$  defined over the polyhedral sets  $Z$  and  $W$  has a local star minimum at the point  $(\bar{z}, \bar{w})$  if  $\phi(\bar{z}, \bar{w}) \leq \phi(z, w)$  for all  $(z, w) \in N(\bar{z}, \bar{w})$  where  $N(\bar{z}, \bar{w})$  represents the set of all adjacent extreme points of  $(\bar{z}, \bar{w})$ .

Theorem 2.2.4 :  $(\bar{z}, \bar{w})$  is a local star minimum of RDLAP-3 if and only if  $\bar{z}$  is a solution to:

$$\begin{aligned} \text{P-1} \quad & \text{Minimize:} \quad \phi(z, \bar{w}) = z^t D \bar{w} \\ & \text{subject to:} \quad z \in Z \end{aligned}$$

and  $\bar{w}$  is a solution to:

P-2            Minimize:  $\phi(\bar{z}, w) = \bar{z}^t Dw$   
                  subject to:  $w \in \bar{w}$

Proof : Since the problem RDLAP-3 was shown to be of the same form as the Bilinear Programming problem BLP-1, we refer the reader to the proof of a similar theorem for the problem BLP-1 in (25).

We will use the above theorem to develop an algorithm to find a local star minimum.

Step 1. Start with an extreme point of the set  $Z$ , say  $\bar{z} = z^0$ . Solve P-2 to obtain a solution  $w^1$ . Go to step 2.

Step 2. Solve P-1 using  $\bar{w} = w^1$  and obtain a solution  $z = z^1$ . If  $z^1 = z^0$ , stop. Otherwise go to step 1 with  $z^0 = z^1$ .

The above procedure is finitely convergent since the number of extreme points is finite, the objective function value has a finite minimum and each sequence of steps from 1 to 2 yields a strict decrease in the value of the objective function so that no extreme point is revisited at any iteration.

The following discussion deals with properties of the objective function in an  $\epsilon$  neighbourhood of a point  $(\bar{z}, \bar{w})$ .

Definition 2.2.5 : The function  $\phi(z, w)$  defined over the sets  $Z$  and  $W$  has a local minimum at the point  $(\bar{z}, \bar{w})$  if  $\phi(\bar{z}, \bar{w}) \leq \phi(z, w)$  for all  $(z, w) \in B_\epsilon(\bar{z}, \bar{w}) \cap (ZXW)$  where  $B_\epsilon(z, w)$  is an  $\epsilon$  neighbourhood around  $(z, w)$ .

Definition 2.2.6 : The function  $\phi(z,w)$  defined over the sets  $Z$  and  $W$  has a global minimum at the point  $(\bar{z}, \bar{w})$  if  $\phi(\bar{z}, \bar{w}) \leq \phi(z,w)$  for each  $z$  in  $Z$  and  $w$  in  $W$ .

Definition 2.2.7 : A pseudo global minimum is a local star minimum which is also a local minimum.

In the next chapter, we will develop an extremely efficient algorithm to find a pseudo global minimum and illustrate with the help of an example the property that the problem RDLAP-3 may have pseudo global minima different from the global minimum.

Property 2.2.8 : The rectilinear distance location-allocation problem has an optimal solution  $(\bar{z}, \bar{w})$  such that  $\bar{z}$  and  $\bar{w}$  are extreme points of  $Z$  and  $W$  respectively.

Proof: For an arbitrary  $w$ , consider the problem:

$$\text{Minimize: } \phi(z,w) = z^t D w$$

$$\text{subject to: } z \in Z$$

Since this is a linear program, it has an extreme point solution  $\bar{z}$  with  $\phi(\bar{z}, w) \leq \phi(z, w)$  for all  $z \in Z$ . Now consider the problem:

$$\text{Minimize: } \phi(\bar{z}, w) = \bar{z}^t D w$$

$$\text{subject to: } w \in W$$

Again, this is a linear program with an optimal extreme point solution  $\bar{w}$  such that  $\phi(\bar{z}, \bar{w}) \leq \phi(\bar{z}, w)$  for all  $w \in W$ .

But,  $\phi(\bar{z}, w) \leq \phi(z, w)$  for all  $z \in Z$ . Hence,  $\phi(\bar{z}, \bar{w}) \leq \phi(z, w)$  for all  $z \in Z$  and  $w \in W$ . Therefore,  $(\bar{z}, \bar{w})$  is an optimal solution with  $\bar{z}$  an extreme point of  $Z$  and  $\bar{w}$  an extreme point of  $W$ .

At this stage, it may appear that Martos' Theorem which claims that quasiconcavity is a necessary and sufficient condition for extreme point optimality is being violated. But, if we look closely, the theorem stipulates that quasiconcavity is a necessary condition only if the function attains its optimum at an extreme point of all convex polyhedral subsets of its constraint set  $L$ . For the problem RDLAP, it is obvious that if we consider constraints involving both  $z$  and  $w$ , the proof will no longer hold.

Using the same reasoning as above one can argue that even though  $\phi$  is not quasiconvex, yet it may be possible for each local minimum to be a global minimum. Unfortunately, the next property shows that this is not the case.

Property 2.2.9 : The rectilinear distance location-allocation problem may have local minima different from the global minimum.

We will provide two examples to illustrate this property at the end of Chapter IV.

## CHAPTER III

AN ALGORITHM TO SOLVE THE RECTILINEAR  
DISTANCE LOCATION PROBLEM

3.1 A Primal Simplex Based Algorithm for the  
Rectilinear Distance Location Problem

As mentioned earlier, the rectilinear distance location-allocation problem reduces to a rectilinear distance location problem when the allocation of resources from new to existing facilities is fixed. Denoting the known quantity shipped from new facility  $i$  to existing facility  $j$  by  $\bar{w}_{ij}$ , the mathematical model for the rectilinear distance location problem can be stated as follows:

$$\begin{aligned}
 \text{RDLP-1} \quad & \text{Minimize: } \sum_{i=1}^m \sum_{j=1}^n \bar{w}_{ij} (x_{ij}^+ + x_{ij}^- + y_{ij}^+ + y_{ij}^-) \\
 & \text{subject to: } \quad x_{ij}^+ - x_{ij}^- = d_j - x_i \quad \text{for all } i \text{ \& } j \\
 & \quad \quad \quad y_{ij}^+ - y_{ij}^- = e_j - y_i \quad \text{for all } i \text{ \& } j \\
 & \quad \quad \quad x_{ij}^+ \cdot x_{ij}^- = 0, \quad y_{ij}^+ \cdot y_{ij}^- = 0 \quad \text{for all } i \text{ and } j \\
 & \quad \quad \quad x_i, y_i, x_{ij}^+, x_{ij}^-, y_{ij}^+, y_{ij}^- \geq 0 \quad \text{for all } i \text{ \& } j
 \end{aligned}$$

Notice that the problem can be broken up into two subproblems, one corresponding to the variables  $x_{ij}^+$ ,  $x_{ij}^-$ , and  $x_i$  for all  $i$  and  $j$ , and the other corresponding to the variables  $y_{ij}^+$ ,  $y_{ij}^-$  and  $y_i$  for all  $i$  and  $j$ . Each one of the above problems can be further subdivided into  $m$  subproblems corresponding to the variables associated with the  $i^{\text{th}}$  new facility. Denoting the  $i^{\text{th}}$  subproblem associated with the  $x$ -variables by  $LP-i(x)$  and those with the  $y$ -variables by  $LP-i(y)$ , we can state the subproblems as follows :

Given  $\bar{w}_{ij} \geq 0$ ,

$$\begin{aligned}
 LP-i(x) \quad & \underline{\text{Minimize}} : \sum_{j=1}^n \bar{w}_{ij} (x_{ij}^+ + x_{ij}^-) \\
 & \underline{\text{subject to}}: \quad x_{ij}^+ - x_{ij}^- = d_j - x_i \quad \text{for all } j \\
 & \quad \quad \quad x_{ij}^+ \cdot x_{ij}^- = 0 \quad \text{for all } j \\
 & \quad \quad \quad x_i, x_{ij}^+, x_{ij}^- \geq 0 \quad \text{for all } j
 \end{aligned}$$

The subproblem  $LP-i(y)$  can be stated similarly. We will now develop an algorithm to solve  $LP-i(x)$ . All statements regarding  $LP-i(x)$  are equally applicable to problem  $LP-i(y)$  and can easily be extended to hold for all  $i$ .

The problem  $LP-i(x)$  is a linear program if we choose to disregard the constraints of the form  $x_{ij}^+ \cdot x_{ij}^- = 0$ . If

the solution to this linear program satisfies the constraints of the form  $x_{ij}^+ \cdot x_{ij}^- = 0$ , the problem LP-i(x) can be considered to be solved. Fortunately, we can guarantee that this will always be the case, since the columns for the variables  $x_{ij}^+$  and  $x_{ij}^-$  are linearly dependent and hence can never be in the basis at the same time. Also, without loss of generality, we can transfer the origin in such a way that  $(d_j, e_j) \geq 0$  for all j.

We will now develop a primal simplex based algorithm to solve the problem LP-i(x) and characterize the simplex tableau at each iteration. The objective of the development presented in this section is two fold. Firstly, we shall obtain an algorithm to solve the problem RDLP, and secondly, the characterization of the optimal tableau developed in this section will play a very important part in the chapters to follow.

The problem LP-i(x) is restated below without the subscript i in order to simplify notation. Also, constraints of the form  $x_{ij}^+ \cdot x_{ij}^- = 0$  have been deleted since the linear programming solution to the relaxed problem automatically satisfies these constraints. Note that the problem RDLP-1 can be broken up into 2m such problems, each one of which can be solved using the algorithm to be developed

in this section.

Given  $w_j \geq 0$ ,

$$\begin{array}{ll}
 \text{LP-(x)} & \text{Minimize: } \sum_{j=1}^n w_j (x_j^+ + x_j^-) \\
 & \text{subject to: } x_j^+ - x_j^- = d_j - x \quad j=1, \dots, n \\
 & x_j^+, x_j^-, x \geq 0 \quad j=1, \dots, n
 \end{array}$$

It is assumed without loss of generality that the indices of the variables  $x_j^+$ ,  $x_j^-$  and the quantity  $w_j$  are ordered such that  $d_j \leq d_{j+1}$  for  $j=1, \dots, n-1$  and that  $d_j \geq 0$  for all  $j$ .

The characterization of the simplex tableau and the algorithm to solve problem LP-(x) are based on the following lemmas and theorems.

Lemma 3.1.1 : The problem LP-(x) has a finite optimum.

Proof : The dual of the problem LP-(x) is as follows:

$$\begin{array}{ll}
 \text{Maximize:} & \sum_{j=1}^n d_j v_j \\
 \text{subject to:} & -w_j \leq v_j \leq w_j \quad j=1, \dots, n \\
 & \sum_{j=1}^n v_j = 0.
 \end{array}$$

Since  $w_j \geq 0$  for all  $j$ ,  $v_j = 0$  for all  $j$  is a feasible solution to the dual problem. Clearly, the primal also has a feasible solution. Hence, the statement of the

lemma follows from the duality theorem of linear programming.

Theorem 3.1.2 : Any basic feasible solution will have the variable  $x$  basic except when the variables  $x_1^+, \dots, x_n^+$  are basic.

Proof : Since there are  $n$  linearly independent constraints, the basis at any iteration will consist of  $n$  variables. Suppose the variable  $x$  is not basic and, without loss of generality, the variables  $x_1^-, \dots, x_r^-, x_{r+1}^+, \dots, x_n^+$  are basic for some  $r=1, \dots, n$ . Then, the basis matrix  $B$  can be obtained from the following starting tableau. Note that an initial basis is readily available, and the zeroth row has been updated so that the coefficients corresponding to the basic variables are all zero. In the starting tableau we have let,

$$A_1 = \sum_{j=1}^n w_j > 0$$

The starting tableau is presented in Table 1. Therefore,

$$B = \begin{matrix} & x_1^-, \dots, x_r^-, & x_{r+1}^+, \dots, x_n^+ \\ \begin{matrix} x_1^- \\ \cdot \\ x_r^- \\ x_{r+1}^+ \\ \cdot \\ x_n^+ \end{matrix} & \begin{bmatrix} & & \\ & -I & 0 \\ & & \\ & 0 & I \end{bmatrix} & = B^{-1} \end{matrix}$$

Table 1. Updated Starting Tableau for the Problem LP-(x)

	$z_0$	$x_1^+$	$x_1^-$	$x_2^+$	$x_2^- \dots x_k^+$	$x_k^- \dots x_n^+$	$x_n^-$	$x$	RHS	
$z_0$	1	0	$-2w_1$	0	$-2w_2$	0	$-2w_k$	$-2w_n$	$A_1$	-
$x_1^+$	0	1	-1	0	0...0	0...0	0	1		$d_1$
$x_2^+$	0	0	0	1	-1...0	0...0	0	1		$d_2$
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
$x_k^+$	0	0	0	0	0...1	-1...0	0	1		$d_k$
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
$x_n^+$	0	0	0	0	0...0	0...1	-1	1		$d_n$

Hence the RHS for the tableau in which the variables  $x_1^-, \dots, x_r^-, x_{r+1}^+, \dots, x_n^+$  are basic is given by  $B^{-1}d$  where,

$$d = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} .$$

But,

$$B^{-1}d = \begin{bmatrix} -d_1 \\ \vdots \\ -d_r \\ d_{r+1} \\ \vdots \\ d_n \end{bmatrix} \not\geq 0$$

Since  $d_j \geq 0$  for  $j=1, \dots, n$  the above solution is not feasible and hence, we have obtained a contradiction.

The only remaining possibility is when  $r=0$ , i.e., when  $x_1^+, \dots, x_n^+$  are basic, corresponding to the situation in the starting tableau.

Before stating the next theorem, we will introduce some notation. Let  $z_k - c_k$  represent the zeroth row coefficient for the  $k^{\text{th}}$  variable at any fixed iteration. For a minimization problem, at optimality,  $z_k - c_k \leq 0$  for each  $k$ .

Theorem 3.1.2 : Every non-optimal tableau has precisely one  $(z_k - c_k) > 0$ .

Proof: From Table 1, it is obvious that the theorem is true for the starting tableau. Now consider an intermediate tableau. By Theorem 3.1.1, the variable  $x$  has to be basic. Again, without loss of generality we can let the

other basic variables be represented by  $x_1^-, \dots, x_{r-1}^-, x_{r+1}, \dots, x_n^+$ . Then the basis matrix is given by :

$$B = \begin{array}{c} x_1^- \\ \vdots \\ x_{r-1}^- \\ x \\ x_{r+1}^+ \\ \vdots \\ x_n^+ \end{array} \begin{bmatrix} x_1^- & \dots & x_{r-1}^- & x & x_{r+1}^+ & \dots & x_n^+ \\ & & & 1 & & & \\ & & -I & \vdots & & 0 & \\ & & & \vdots & & & \\ & & & 1 & & & \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ & & & \vdots & & & \\ & & 0 & \vdots & & I & \\ & & & \vdots & & & \\ & & & 1 & & & \end{bmatrix}$$

Therefore,

$$B^{-1} = \begin{array}{c} x_1^- \\ \vdots \\ x_{r-1}^- \\ x \\ x_{r+1}^+ \\ \vdots \\ x_n^+ \end{array} \begin{bmatrix} x_1^- & \dots & x_{r-1}^- & x & x_{r+1}^+ & \dots & x_n^+ \\ & & & 1 & & & \\ & & -I & \vdots & & 0 & \\ & & & \vdots & & & \\ & & & 1 & & & \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ & & & -1 & & & \\ & & 0 & \vdots & & I & \\ & & & \vdots & & & \\ & & & -1 & & & \end{bmatrix}$$

The updating matrix  $U$  is given by :

$$U = \begin{bmatrix} 1 & c_B B^{-1} \\ 0 & B^{-1} \end{bmatrix}$$

where,  $c_B = (2w_1, \dots, 2w_{r-1}, -A_1, 0, \dots, 0)$

Hence, the tableau at an intermediate iteration can be obtained by premultiplying the starting tableau in Table 1 by  $U$ . The resulting tableau is shown in Table 2. In the tableau,

$$A^+ = -\sum_{j=1}^n w_j + \sum_{j=1}^{n-1} w_j \quad \text{and} \quad A^- = -A^+ - 2w_r$$

Consider the column of a non-basic variable  $x_k$  other than  $x_r^+$  or  $x_r^-$ . It is non-positive. Hence, if  $(z_k - c_k) > 0$ , then the problem is unbounded, violating Lemma 3.1.1. Hence  $(z_k - c_k) \leq 0$  for all variables other than  $x_r^+$  or  $x_r^-$ . The only columns for which  $(z_k - c_k)$  can possibly be greater than zero are the ones corresponding to the variables  $x_r^+$  and  $x_r^-$ , with  $(z_k - c_k)$  values given by  $A^+$  and  $A^-$  respectively. Now,  $A^+ + A^- = -2w_r < 0$ . Hence, either  $A^+$  or  $A^-$  is negative. If both are  $< 0$ , the tableau is optimal since we have shown that all other

Table 2. Simplex Tableau for the Problem LP-(x)

	$z$	$x_1^+$	$\dots$	$x_{r-1}^+$	$x_r^+$	$x_{r+1}^+$	$\dots$	$x_n^+$	$x_1^-$	$\dots$	$x_{r-1}^-$	$x_r^-$	$x_{r+1}^-$	$\dots$	$x_n^-$	$x$	RHS
$z$	1	$-2w_1$	$\dots$	$-2w_{r-1}$	$A^+$	0	$\dots$	0	0	$\dots$	0	$A^-$	$-2w_{r+1}$	$\dots$	$-2w_n$	0	-
$x_1^-$	0				1							-1				0	$d_r - d_1$
$\vdots$	$\vdots$				$\vdots$							$\vdots$				$\vdots$	$\vdots$
$\vdots$	$\vdots$		$-I$		$\vdots$	0			$I$			$\vdots$		0		$\vdots$	$\vdots$
$\vdots$	$\vdots$				$\vdots$							$\vdots$				$\vdots$	$\vdots$
$x_{r-1}^-$	0				1							-1				0	$d_r - d_{r-1}$
$x$	0	0	$\dots$	0	1	0	$\dots$	0	0	$\dots$	0	-1	0	$\dots$	0	1	$d_r$
$x_{r+1}^+$	0				-1							1				0	$d_{r+1} - d_r$
$\vdots$	$\vdots$				$\vdots$							$\vdots$				$\vdots$	$\vdots$
$\vdots$	$\vdots$		0		$\vdots$	$I$			0			$\vdots$		$-I$		$\vdots$	$\vdots$
$\vdots$	$\vdots$				$\vdots$							$\vdots$				$\vdots$	$\vdots$
$x_n^+$	0				-1							1				0	$d_n - d_r$

$(z_k - c_k) \leq 0$ . Hence, either  $A^+$  or  $A^-$  is  $\leq 0$ , and the other is  $\geq 0$ , which proves the theorem.

### 3.2 Properties of the Rectilinear

#### Distance Location Problem

We will now develop certain properties of the rectilinear distance location problem based on the Lemma and Theorems just proved.

Property 3.2.1 : The variable  $x$  is basic at all iterations except in the starting tableau. Also, for any iteration  $t \geq 1$ , its value is given by  $d_j$  for some  $j \in \{1, \dots, n\}$ .

This follows directly from Theorem 3.1.2 and Table 2.

Property 3.2.2 : The tableau at the  $t^{\text{th}}$  iteration can be obtained by replacing  $r$  by  $t$  in Table 2. If the tableau at the  $r^{\text{th}}$  iteration is not optimal, then the variable  $x_r^-$  will enter the basis, the variable  $x_{r+1}^+$  will leave the basis and the value of the basic variable  $x$  will change from  $d_r$  to  $d_{r+1}$ . At the first iteration, the variable  $x$  enters the basis at a value  $d_1$  and the variable  $x_1^+$  leaves the basis. This follows directly from the pivoting operation of the simplex method. The proof for this property can be developed using an induction argument. We have not outlined a detailed proof since we will not be using this property directly.

Property 3.2.3 : The only two adjacent extreme points corresponding to the solution  $x = d_r$  are those for which  $x = d_{r-1}$  and  $x = d_{r+1}$ . This property can be shown to be true as follows. An adjacent extreme point can be characterized by pivoting in a non-basic variable. If any one of the non-basic variables  $x_1^+, \dots, x_{r-1}^+, x_{r+1}^-, \dots, x_n^-$  is pivoted into the basis, the resulting solution is infeasible. The only other non-basic variables remaining are  $x_r^+$  and  $x_r^-$ . If  $x_r^+$  is pivoted into the basis, the resulting solution has  $x = d_{r-1}$ . If  $x_r^-$  is pivoted into the basis the resulting solution has  $x = d_{r+1}$ . Hence, the only two adjacent extreme points corresponding to the solution  $x = d_r$  are those corresponding to the solutions where  $x = d_{r-1}$  and  $x = d_{r+1}$ .

Property 3.2.4 : There exists an  $r \in \{1, \dots, n\}$  such that  $A_r^+ = -\sum_{j=r}^n w_j + \sum_{j=1}^{r-1} w_j$  and  $A_r^- = \sum_{j=r+1}^n w_j - \sum_{j=1}^r w_j$  are both  $\leq 0$ .

Proof: When  $r=1$ ,  $A_1^+ = -\sum_{j=1}^n w_j < 0$  and  $A_1^- = \sum_{j=1}^1 w_j - 2w_1$ .  
When  $r=n$ ,  $A_n^+ = \sum_{j=1}^n w_j - 2w_n$  and  $A_n^- = -\sum_{j=1}^n w_j < 0$ .

Also,  $A_r^+ + A_r^- \leq 0$  for each  $r \in \{1, \dots, n\}$ .

Note that the sequence  $A_r^+$  is monotonically increasing while the sequence  $A_r^-$  is monotonically decreasing.

Since  $A_1^+ < 0$ ,  $A_n^- < 0$  and  $A_r^+ + A_r^- \leq 0$  for all  $r$ , it follows

that for some  $r \in \{1, \dots, n\}$   $A_r^+$  and  $A_r^-$  are  $\leq 0$ .

The algorithm to solve the problem LP-(x) involves finding  $r$  such that  $A_r^+$  and  $A_r^-$  are  $\leq 0$ , in which case the tableau in Table 2 represents the optimal solution to the problem. The following procedure can be used to find  $r$ .

Step 1 : Set  $r = 1$ . Compute  $W_1 = \sum_{j=1}^n w_j - 2w_1$ . If  $W_1 \leq 0$ , stop with  $x^* = d_1$ . Otherwise go to step 3.

Step 2 : Increment  $r$  by one and go to step 3.

Step 3 : Compute  $W_{r+1} = W_r - 2w_{r+1}$ . If  $W_{r+1} \leq 0$ , stop with  $x^* = d_{r+1}$ . Otherwise go to step 2.

Property (4) ensures that a solution to the problem LP-(x) exists.

## CHAPTER IV

## THE CUTTING PLANE ALGORITHM

In this chapter we will develop a cutting plane algorithm to solve exactly, the rectilinear distance location-allocation problem. Cutting plane algorithms in general have been found to be computationally inefficient because they tend to destroy any special structure of the problem. However, in this case, we will show that the ease of generation and depth of cut more than offset the above mentioned disadvantage. Further, we will show that the cutting planes need to be introduced in either the set  $Z$  or the set  $W$ , and the judicious choice of the set  $Z$  for the introduction of the cutting planes reduces this disadvantage even further.

In Section 4.3 through 4.6 we will develop algorithms for cut generation, determination of a pseudo global minimum and solution to the parametric problem. Some of the later sections will be devoted to discussing certain modifications and special topics. Lastly, we will provide two illustrative examples and an efficient algorithm for the determination of a good starting solution.

#### 4.1 The Cutting Plane Algorithm for Bilinear Programming Problems

The cutting plane to be developed in this chapter is based on the theory of polar cuts for Bilinear Programming Problems (25) . In this section, we will discuss without proof some of the fundamental concepts involved in developing this generalized cutting plane algorithm.

Definition 4.1.1 : Let  $A \subseteq R_m$ . Given a function  $f: R_n \times R_m$  to  $R_1$ , and a scalar  $k$ , the generalized polar of  $A$  is defined as :

$$A^{\circ}(k) = \{x \in R_n : f(x,y) \leq k \text{ for all } y \in A\}$$

We wish to construct polar sets with respect to the objective function  $\phi(z,w)$  using as the scalar  $k$ , the current best solution of the problem, and find the intersection of the polar with as large a subset of the feasible region as is computationally tractable. Then, the optimum over this subset would be known. If the entire feasible region can be broken up into a finite number of such subsets, the problem will have been solved.

In order to maintain the separable structure of the constraint set of the problem RDLAP-3, the cut defined by the intersection of the polar set with the feasible region will

be defined in terms of the variables included in the vector  $z$ . The choice for introducing the cut in the Z-set instead of the W-set will shortly become clear when we illustrate the ease with which a cut can be generated in the Z-set.

The basic idea behind the generation of a cutting plane is as follows. Let  $(\bar{z}, \bar{w})$  be a pseudo global minimum. Consider the polyhedral cone  $C$  with vertex at  $\bar{z}$  and whose  $r$  extreme rays are given by  $\xi^j = \{z : z = \bar{z} - \bar{e}^j \lambda_j, \lambda_j \geq 0\}$  for all  $j \in J$ , where  $J$  is the set of  $r$  indices corresponding to the non-basic variables, and the vectors  $\bar{e}^j$  are the extended non-basic columns obtained from the optimal simplex tableau for the point  $\bar{z}$ . Suppose  $Q$  is any closed convex set that has the property that  $\phi(z, w) \geq \phi(\bar{z}, \bar{w}) \geq k$  for each  $z \in Q$  and  $w \in W$ . Since  $(\bar{z}, \bar{w})$  is also a local star minimum,  $\phi(\bar{z}, w) \geq \phi(\bar{z}, \bar{w})$  for each  $w \in W$ . Hence  $\bar{z} \in Q$ . Let us consider the set  $C \cap Q$ . Since  $\phi(z, w) \geq \phi(\bar{z}, \bar{w})$  for each  $z$  in  $C \cap Q$  and  $w$  in  $W$ , the global minimum over  $C \cap Q$  is known. In general, the set  $C \cap Q$  is very difficult to define. Hence, we construct a subset of  $C \cap Q$  by finding the unique hyperplane  $H$  passing through the intersection of the  $r$  rays  $\xi^j$  with the boundary of  $Q$ . If  $H^-$  is the closed half-space containing  $\bar{z}$ , then the global minimum over  $C \cap H^-$  is known, and hence  $H^+$  will be a valid cutting plane.

Let  $\bar{z}$  be an extreme point of  $Z$  corresponding to the optimum solution to problems  $LP-i(x)$  and  $LP-i(y)$ ,  $i=1, \dots, m$ .

Let  $p_j$ ,  $j$  in  $J$  represent the non-basic variables in the above optimal solution. Then, the following conditions

are necessary for  $\sum_{j \in J} p_j / \bar{\lambda}_j \geq 1$  to be a valid cutting plane.

(i)  $Q$  is a closed convex set such that  $\bar{z} \in Q$

(ii)  $Q \cap$  relative interior of  $\xi^j \neq$  empty set for each  $j$  in  $J$ .

(iii)  $Q \cap \{z \in Z : z^t D w < k \text{ for some } w \text{ in } W\} =$  empty set.

(iv)  $\bar{\lambda}_j = \max (\lambda_j : \bar{z} - \bar{e}^j \lambda_j \in Q)$  if part of the ray  $\xi^j$  is in  $Q$ ,

= infinity, if the entire ray  $\xi^j$  is in  $Q$  for all

$\lambda_j \geq 0$ .

We will now introduce the set  $W^0(k)$  which has all the properties of the set  $Q$ . Let,

$$W^0(k) = \{z : \phi(z, w) = z^t D w \geq k \text{ for each } w \text{ in } W\}$$

Hence, it follows from the definition of  $\bar{\lambda}_j$  that,

$$\bar{\lambda}_j = \text{Max} (\lambda_j : \bar{z} - \bar{e}^j \lambda_j \in W^0(k) )$$

$$= \text{Max}_{\lambda_j > 0} ( \text{Min}_{w \in W} (\bar{z} - \lambda_j \bar{e}^j)^t D w \geq k )$$

From the definition of  $\bar{\lambda}_j$  it is obvious that the range of values for  $\bar{\lambda}_j$  lies in the interval  $(0, +\infty]$ . Note that the computation involved in computing  $\bar{\lambda}_j$  amounts to solving a parametric linear programming problem over the set  $W$ . Hillier and Lieberman (9) have shown that the function  $\Psi(\lambda_j) = \text{Min} (\bar{z} - \lambda_j \bar{e}^j)^t D w$  is concave and hence, unimodal. Therefore,  $\bar{\lambda}_j$  can be easily determined by conducting a search over a finite range of values for  $\lambda_j$ . Section 4.4 is devoted to outlining precisely this procedure.

Once a cutting plane is determined, it is introduced into the  $Z$  set. Denote this incremented set by  $Z^+$ . If there is no point feasible to the  $Z^+$  then the problem is solved and the solution is the one corresponding to the current best solution. If a feasible point exists, a pseudo global minimum for the set  $Z^+$  is determined, and the procedure repeated until no point feasible to the incremented set can be found.

In the sections to follow, we will specialize this cutting plane algorithm to take advantage of the special structure of the problem. We will develop an extremely efficient algorithm for cut generation, outline a method to find a "weak pseudo global minimum", and devise a

procedure either to obtain a feasible point to the set  $Z^+$ , or an indication that no such point exists.

#### 4.2 The Cutting Plane Algorithm for the Problem RDLAP

Although the cutting plane algorithm discussed in Section 4.1 has the property that the separable nature of the constraint set of the problem RDLAP is not destroyed, yet, the special structure of one of these sets is not retained. We will demonstrate below how this disadvantage can be overcome by defining the cutting planes in terms of the variables of the set  $Z$ . It is worth noting at this stage that the specialized algorithm overcomes this drawback without resorting to decomposition techniques, which generally converge slowly.

The algorithms for cut generation, and the determination of a weak pseudo global minimum also exploit the special structure of the set  $Z$ . Both of these algorithms are based on the properties developed in section 3.1, and have an intuitive appeal.

Section 4.6 deals with a solution procedure for the parametric transportation problem, while Section 4.7 discusses algorithms to find an extreme point of the set  $Z$  which is feasible to the current set of cutting planes.

The algorithm of Section 4.1 generates a cutting plane only at a pseudo global minimum in order to ensure that  $\bar{\lambda}_j \in (0, \infty]$ . We will show that due to the special properties of the set  $Z$ , this condition can be relaxed to a certain extent. We will characterize the special properties of such a point and prove that a valid cutting plane can be generated starting from this point.

And lastly, we will discuss the convergence of the algorithm as a whole, and make certain statements comparing the generalized cutting plane algorithm to other cutting plane algorithms.

### 4.3 Algorithm for Cut Generation

As outlined in Section 4.1 a valid cutting plane is given by  $\sum_{j \in J} p_j / \bar{\lambda}_j \geq 1$  where  $\bar{\lambda}_j$  is obtained by solving a parametric problem. The cardinality of the set  $J$  is equal to  $2m(n+1)$  for the problem RDLP-1. This quantity represents the total number of non-basic variables in the optimal simplex tableau corresponding to the solution  $\bar{z}$  of the problem :

$$\text{Minimize : } \phi(z, \bar{w}) = z^t D \bar{w}$$

$$\text{subject to: } z \in Z .$$

In section 3.1, we have shown that the above problem can be

broken up into  $2m$  subproblems denoted by  $LP-i(x)$  and  $LP-i(y)$  for  $i=1, \dots, m$ . We have also shown how to characterize the simplex tableau corresponding to each subproblem. Hence, the vectors  $\bar{e}^j$  can be determined from the extended simplex tableau which includes the vector equation :

$$z_{NB} + (-I) z_{NB} = 0 ,$$

where  $z_{NB}$  is a column vector of non-basic variables.

Although it is now relatively easy to solve the parametric linear programming problem for  $\bar{\lambda}_j$  (as will be shown in Section 4.4), the very task of solving  $4m(n+1)$  such problems for each cut makes this algorithm extremely unattractive. Fortunately, the special structure of the optimal simplex tableau is such that we can directly set  $\bar{\lambda}_j = \infty$  for  $2m(n-1)$  of these non-basic variables. Hence, we will now show that any cut can have at most  $4m$  non-basic variables with  $\bar{\lambda}_j \neq \infty$ , or alternately, for each subproblem  $LP-i(x)$  and for each subproblem  $LP-i(y)$  at most only 2 non-basic variables will have  $\bar{\lambda}_j \neq \infty$ .

The simplex tableau corresponding to the solution  $x_i = d_r$  is given in Table 2. To obtain the extended simplex tableau, the vector equation  $z_{NB} + (-I) z_{NB} = 0$  is attached to this table.

Consider the parametric problem corresponding to any of the non-basic variables  $x_1^-, \dots, x_{r-1}^-, x_{r+1}^+, \dots, x_n^+$ . Note that the subscript  $i$  has been dropped for notational simplification. Since the vector  $\bar{e}^j$  is  $\leq 0$  for any of these non-basic variables, every component of  $\bar{z} - \bar{e}^j \lambda_j$  is greater than or equal to the corresponding component of  $\bar{z}$ . Note that the cost associated with some of the  $w_j$ 's is increased by an amount  $\lambda_j$  and therefore the solution to the problem :

$$\text{Min}_{w \in W} (\bar{z} - \lambda_j \bar{e}^j)^t D w$$

will be  $d_k$  for all  $\lambda_j$ . Hence,  $\bar{\lambda}_j = \infty$  for all non-basic variables, except  $x_r^+$  and  $x_r^-$  in the simplex tableau corresponding to the solution  $x = d_r$ . We will now give a physical interpretation of the parametric problem for the variables  $x_r^+$  and  $x_r^-$ .

The expression  $(\bar{z} - \bar{e}^j \lambda_j)$  for the variable  $x_r^+$  represents an increase in the value of the variables  $x_1^+, \dots, x_r^+$  and a decrease in the value of the variables  $x_1^-, x_{r+1}^-, \dots, x_n^-$  by an amount  $\lambda_j$ . This exactly corresponds to moving the new facility  $i$  from its location at the  $x$  coordinate  $d_r$ , in the negative  $x$ -direction by a distance  $\lambda_j$ . The physical interpretation is valid only for those values of  $\lambda_j$  lying in the interval  $[0, d_r - d_k]$ , where

$d_k$  is the closest x-coordinate of some existing facility strictly less than  $d_r$ . Similarly, for the variable  $x_r$  the expression  $(\bar{z} - \bar{e}^j \lambda_j)$  can be interpreted as moving the new facility  $i$  from its location at  $d_r$  in the positive x-direction by a distance  $\lambda_j$ . This interpretation can be extended to the variables of problem LP-i(y) too. Hence, we have identified four directions coinciding with the coordinate axes, along which 4 parametric linear programming problems have to be solved for each new facility  $i$ .

An interesting property of this cutting plane algorithm which is valid only if all new facilities have the same capacity is that once a cutting plane is generated,  $m!$  additional cutting planes can be written down corresponding to the  $m!$  different permutations obtained by relocating the  $m$  new facilities among the  $m$  points at which the initial cutting plane was developed. Obviously, interchanging the locations of new facilities with the same capacities has no effect on the solution, but the new cutting planes generated do cut off extreme points which the original cutting plane was not deep enough to reach. The problem with this scheme is that as  $m$  becomes large, the storage required increases tremendously fast, limiting  $m$  to be no greater than 6 or 7. This property is illustrated in Example 3 of section 4.3.

#### 4.4 Determination of a Weak Pseudo

##### Global Minimum

In this section, we will introduce a special point which we shall call a weak pseudo global minimum, and prove that a valid cutting plane can be developed from this point. Next, we will present an algorithm to find a weak pseudo global minimum and show that in the absence of cutting planes, such a point is precisely a pseudo global minimum.

Our objective in this development is to overcome the handicap of having to solve the location problem with the cutting planes as additional constraints in order to find a local star minimum, and eventually a pseudo global minimum. According to the cutting plane algorithm of Section 4.1, a valid cutting plane can only be developed from a pseudo global minimum. We will show that due to the special structure of the set  $Z$ , this condition can be relaxed to a weak pseudo global minimum. Also, the algorithm which we will develop to find a weak pseudo global minimum will not involve the above mentioned handicap. Note that decomposition techniques are not applicable since there is no guarantee that constraints of the type  $x_{ij}^+ \cdot x_{ij}^- = 0$  will be satisfied at optimality, unless some form of restricted

basis entry is imposed.

Let  $N_m(\bar{z})$  denote the set of adjacent extreme points of  $\bar{z}$ . Also, at stage  $s$ , let  $g^s(z) \geq 1$  denote the  $s$  cutting planes  $g_i(z) \geq 1, i=1, \dots, s$  to be introduced into the set  $Z$ .

Definition 4.4.1 : At stage  $s$ , let  $\bar{z}$  be an extreme point of  $Z$  and  $\bar{w}$  be an extreme point of  $W$  such that  $g^s(\bar{z}) \geq 1$  and  $\text{Min}_{w \in W} \phi(\bar{z}, w) = \phi(\bar{z}, \bar{w})$ . Then  $(\bar{z}, \bar{w})$  is said to be a weak pseudo global minimum of  $\phi(z, w)$  with respect to the cuts  $g^s(z) \geq 1$  if for each  $z$  in  $N(\bar{z})$  either  $g^s(z) \not\geq 1$  or  $\text{Min}_{w \in W} \phi(z, w) \geq \phi(\bar{z}, \bar{w})$ .

Lemma 4.4.2 :  $(\bar{z}, \bar{w})$  is a weak pseudo global minimum implies that a valid cutting plane can be generated with  $k =$  value of the objective function at the current best solution.

Proof : Let  $(\bar{z}, \bar{w})$  be a weak pseudo global minimum.

Case (i) : Let  $(z, \bar{w})$  be such that  $\text{Min}_{w \in W} \phi(z, w) \geq \phi(\bar{z}, \bar{w})$  for

a  $z$  in  $N_m(\bar{z})$ . Since  $z$  belongs to  $N_m(\bar{z})$ , then,

$z = \bar{z} - \lambda'_j e^j$  for some  $\lambda'_j \geq 0$ . Also,

$$\text{Min}_{w \in W} \phi(z, w) \geq \phi(\bar{z}, \bar{w}) \geq k.$$

Substituting for  $z$  in the above expression we have,

$$\text{Min}_{w \in W} \phi(z, w) = \text{Min}_{w \in W} z^t D w = \text{Min}_{w \in W} (\bar{z} - \lambda'_j e^j)^t D w \geq k.$$

Therefore,  $\Psi(\lambda'_j) \geq k$  for  $\lambda'_j > 0$ .

Since  $\Psi(\lambda_j)$  is concave and  $\Psi(0) \geq k$  because  $k$  corresponds to the current best solution,  $\Psi(\lambda_j) \geq k$  for all  $0 < \lambda_j \leq \lambda'_j$ . Therefore,  $\bar{\lambda}_j = \text{Max}_{\lambda_j > 0} (\Psi(\lambda_j) \geq k) > 0$ , and hence a valid cut can be generated with  $k =$  the current best solution.

Case (ii) : Let  $(\bar{z}, \bar{w})$  be such that  $g^s(z) < 1$  for a  $z$  in  $N_m(\bar{z})$ . Since  $g^s(z) < 1$ ,  $g_i(z) < 1$  for some  $i \in \{1, \dots, s\}$ . Let  $g_p(z) < 1$  for some  $p$  and  $k^p$  be the current best solution at stage  $p \leq s$ . Then  $k \leq k^p$ .

Further, the point  $z$  satisfies  $g_p(z) < 1$  i.e., violates the cut at the  $p^{\text{th}}$  stage. This implies  $\text{Min}_{w \in W} \phi(z, w) \geq k^p$ .

But,  $k^p \geq k$ . Hence  $\text{Min}_{w \in W} \phi(z, w) \geq k$ .

Now using the same argument as in Case (i), we have  $\bar{\lambda}_j > 0$ . Since  $\bar{\lambda}_j > 0$  in both cases, a valid cutting plane can be generated with  $k =$  the current best solution starting from a weak pseudo global minimum.

The algorithm to find a weak pseudo global minimum is directly based on the definition of this point. We will first give a step by step statement of the algorithm and then show that the algorithm converges finitely.

Step 1 : At stage  $s$  find an extreme point  $\bar{z}^s$  of set  $Z$  feasible to the cuts  $g^s(z) \geq 1$ . If no such point exists,

terminate. The problem has been solved and the solution corresponds to the point with the current best objective function value  $k$ . Otherwise go to step 2. (A detailed algorithm for this step is presented in the Section 4.7, later)

Step 2 : Find a  $\bar{z}$  in  $N_m(\bar{z}^s)$  such that  $g^s(\bar{z}) \geq 1$  and ,

$$\text{Min}_{w \in W} \phi(\bar{z}, w) < \text{Min}_{w \in W} \phi(\bar{z}^s, w) .$$

If no such point exists, terminate with  $(\bar{z}^s, \bar{w})$  as a weak pseudo global minimum, where  $\bar{w}$  solves  $\text{Min}_{w \in W} \phi(\bar{z}^s, w)$  .

Otherwise go to step 3.

Step 3 : Replace  $\bar{z}^s$  by  $\bar{z}$  and go to step 2.

Convergence of the Algorithm : Note that the cardinality of set  $N_m(z)$  is finite, the number of extreme points of set  $Z$  is also finite, and the objective function has a finite minimum. Since each sequence of steps from 2 to 3 involves a strict decrease in the value of the objective function, an extreme point  $\bar{z}$  can never be revisited. Also, because the objective function has a finite minimum, the process must terminate finitely. Hence the algorithm is finitely convergent.

We will now discuss the algorithm with respect to the case where the set of cutting planes is empty ( $s = 0$ ).

In step 1,  $\bar{z}^s$  can be selected to be any extreme point of  $Z$ . (In the next section we will justify the selection of a particular extreme point to guarantee a good feasible solution at the end of the first stage.) In step 2, we know that every point  $\bar{z}$  in  $N_m(\bar{z}^s)$  is feasible and hence the check for  $g^s(\bar{z}) \geq 1$  is ignored. The remaining portion of the algorithm is retained as it is. We will now show that  $\bar{z}^0$ , the weak pseudo global minimum obtained at the end of stage  $s = 0$  is also a pseudo global minimum.

Since  $\bar{z}^0$  is a weak pseudo global minimum, we know from Lemma 4.4.2 that a valid cutting plane can be generated from  $\bar{z}^0$ . Hence,  $k = \min_{w \in W} \phi(\bar{z}^0, w) \leq \phi(z, w)$  for each  $z$  in  $N(\bar{z}^0)$  and  $w$  in  $W$ . Also,

$$\min_{w \in W} \phi(\bar{z}^0, w) \leq \phi(\bar{z}, w) \quad \text{for each } w \text{ in } W$$

and  $\bar{z}$  in  $N_m(\bar{z}^0)$ . Hence,  $(\bar{z}^0, \bar{w})$  is a local minimum and

a local star minimum, where  $\bar{w}$  solves  $\min_{w \in W} \phi(\bar{z}^0, w)$ .

From Definition 3.2.7 we conclude that  $(\bar{z}^0, \bar{w})$  is a pseudo global minimum.

#### 4.5 Determination of a Good

##### Starting Feasible Solution

Although the process of finding a pseudo global minimum at stage  $s = 0$  can be initiated from any extreme

point of set  $z$ , it is felt that a better starting solution will tend to reduce the total amount of computation required to find the global minimum. Thus, any reasonable extra effort expended at this stage will perhaps be worthwhile. The algorithm that we will present is based on computational experience, and has a great deal of intuitive appeal, as we shall shortly see. Our objective in introducing this step into the cutting plane algorithm is to avoid pseudo global minima which may be different from the global minimum. We will now characterize two solutions where an inappropriate starting solution may lead to such a point. A new facility  $i$  may be trapped at a pseudo global minimum different from the global minimum if its capacity  $c_i$  is not large enough to meet the requirements  $r_j$  of existing facilities closer to it than to any other new facility, and yet not small enough to be displaced by another new facility. The same situation occurs when new facilities are not located close enough to existing facilities with the largest requirements. We have provided examples in Section 4.3 to illustrate both the above mentioned defects.

The algorithm presented below overcomes the second drawback in step 1 and the first in the sequence of steps from 2 to 3.

Step 1 : Reorder the capacities of the  $m$  new facilities and the requirements at the  $n$  existing facilities such that  $c_{i+1} \leq c_i$ ,  $i=1, \dots, m-1$  and  $r_{i+1} \leq r_i$ ,  $i=1, \dots, n-1$ . If  $m \leq n$ , locate  $(x_i, y_i) = (d_i, e_i)$   $i=1, \dots, m$ . If  $m > n$  locate  $(x_i, y_i) = (d_i, e_i)$   $i=1, \dots, n$  and  $(x_i, y_i) = (d_j, e_j)$  for  $i=n, \dots, m$  and  $j=1, \dots, m-n$ .

Step 2 : Start with the location in step 1. Set the capacities of all new facilities =  $\text{Max} \{c_1, \dots, c_m\}$ . Find a pseudo global minimum  $(\bar{z}, \bar{w})$  using the algorithm of the previous section.

Step 3 : Compute  $\bar{c}_i = \sum_{j=1}^n \bar{w}_{ij}$ ,  $i=1, \dots, m$ . Set the capacity of each new facility  $i$  equal to its original capacity  $c_i$ , for  $i=1, \dots, m$ . If  $\bar{c}_i > c_i$  for any  $i \in \{1, \dots, m\}$ , resolve the problem with original  $c_i$ 's to obtain a new pseudo global minimum  $(\bar{z}, \bar{w})$ . (When resolving the problem, start the solution procedure from the point  $(\bar{z}, \bar{w})$ .) Otherwise, the pseudo global minimum is given by the point  $(\bar{z}, \bar{w})$ .

Computational results indicate that the pseudo global minimum obtained by using this algorithm to generate a starting solution for the algorithm of Section 4.4 almost always turned out to be the global minimum.

#### 4.6 Solution of the Parametric Problem

To determine the quantity  $\bar{\lambda}_j$ , we have to solve

the following parametric linear programming problem :

$$\text{Max}_{\lambda_j > 0} \left( \text{Min}_{w \in W} (\bar{z} - \lambda_j \bar{e}^j)^t Dw \cdot k \right) = \text{Max}_{\lambda_j > 0} (\Psi(\lambda_j) \geq k)$$

But once the expression  $(\bar{z} - \lambda_j \bar{e}^j)$  is simplified to a known quantity,  $\text{Min}_{w \in W} (\bar{z} - \lambda_j \bar{e}^j)$  is just a transportation problem. Hence, computing  $\bar{\lambda}_j$  involves solving a parametric transportation problem. We know that the parametric transportation problem is piecewise linear, with breakpoints occurring each time a change in allocation takes place. Computational experience shows that most breakpoints of the function  $\Psi(\lambda_j)$  coincide with some coordinate of the existing facilities. The difficulty arises when this is not the case, thus making a procedure which tries to take advantage of the piecewise linearity of  $\Psi(\lambda_j)$  to find  $\bar{\lambda}_j$  relatively unattractive. In addition, we found that a direct search procedure such as the Bolzano or Bisection search is extremely efficient in solving this problem. At each step, we were able to eliminate half the current interval, as compared to only 0.38 when using the Golden Section search technique. Hence, all computational results were obtained using the following algorithm based on this search technique.

Step 1 : Define a large number  $L \gg 0$  and a permissible error  $e > 0$ .

Step 2 : Solve the following transportation problem with

$$\lambda_j = L.$$

P-3 : Minimize :  $\Psi(\lambda_j) = (\bar{z} - e^{-j}\lambda_j)^t D_w$

subject to :  $w \in W$

If  $\Psi(L) \geq k$ , terminate with  $\bar{\lambda}_j = L$ .

Step 3 : Define  $\lambda_h = L$ ,  $\lambda_l = 0$  and  $\lambda_r = L/2$ .

Step 4 : Solve P-3 with  $\lambda_j = \lambda_r$ . If  $0 \leq \Psi(\lambda_r) - k \leq e$ , terminate with  $\bar{\lambda}_j = \lambda_r$ .

Step 5 : If  $\Psi(\lambda_r) > k$ , set  $\lambda_l = \lambda_r$ ,  $\lambda_r = (\lambda_l + \lambda_h)/2$  and go to step 4.

Step 6 : If  $\Psi(\lambda_r) < k$ , set  $\lambda_h = \lambda_r$ ,  $\lambda_r = (\lambda_l + \lambda_h)/2$  and go to step 4.

The above algorithm is very simple to program and is found to be very efficient.

#### 4.7 Determination of a Feasible Extreme Point

We will now address ourselves to the task of deve-

loping an algorithm to find an extreme point  $\bar{z}$  of the set  $Z$ , at stage  $s$ , such that  $g^s(\bar{z}) \geq 1$ . Recall that such a point  $\bar{z}$  is required at each stage  $s \geq 1$ , to initiate the search for a weak pseudo global minimum. It was shown in Section 3.1 that the extreme points of the set  $Z$  correspond to the location of new facility  $i$  with coordinates  $(x_i, y_i)$ , and belong to the set  $A_1 \times A_2$  where,

$$A_1 = \{d_j, j=1, \dots, n\} \quad \text{and} \quad A_2 = \{e_j, j=1, \dots, n\}.$$

Denote this set  $A_1 \times A_2$  by  $S$ . Then, the above problem can be stated as :

$$\begin{aligned} \text{P-4} : \quad & \text{Maximize } 0 \\ & \text{subject to : } g^s(z) \geq 1 \\ & z \in S \end{aligned}$$

or equivalently (see 23),

$$\text{P-5} : \quad \begin{array}{ll} \text{Min} & \text{Max} \\ u \geq 0 & z \in S \end{array} \quad u^t (g^s(z) - 1).$$

We will first develop a solution procedure for the following problem with  $u$  known, i.e.,

$$\text{P-6} : \quad \text{Max}_{z \in S} u^t (g^s(z) - 1).$$

Since  $u$  is fixed, the solution  $\bar{z}$  to :

$$\text{P-7 :} \quad \text{Maximize} \quad u^t g^s(z) \\ z \in S$$

will be the same as that of P-6. Note that  $u$  is  $\geq 0$  and  $g_i(z)$  is of the linear form :

$$\sum_{j \in J} p_j / \bar{\lambda}_j, \text{ where } \bar{\lambda}_j > 0 \text{ and the } p_j \text{'s are}$$

the non-basic variables at this point, some of which correspond to  $x_{ij}^+$  and others to  $x_{ij}^-$ . Hence,

$$u^t g^s(z) = \sum_{i=1}^s u_i g_i(z) \text{ is separable in the variables}$$

associated with each new facility and also separable in the variables  $x_{ij}^+$ ,  $x_{ij}^-$  and  $y_{ij}^+$ ,  $y_{ij}^-$ . Therefore, in general,  $u^t g^s(z)$  can be broken up into  $2m$  separable parts each one of which can be written in the form :

$$\sum_{j=1}^m (a_{ij} x_{ij}^+ + b_{ij} x_{ij}^-) \text{ where } a_{ij} \text{ and } b_{ij} \text{ are } \geq 0.$$

Corresponding to each separable part of the objective function  $u^t g^s(z)$ , the constraints  $z \in S$  can be split into  $x_i \in A_1$  and  $y_i \in A_2$  for each  $i \in \{1, \dots, m\}$ . Hence, the problem P-7 can be solved by solving  $2m$  subproblems of the form :

$$\text{P-8} \quad \text{Maximize} \quad \sum_{j=1}^n (a_{ij}x_{ij}^+ + b_{ij}x_{ij}^-)$$

$$x_i \in A_1$$

where  $x_{ij}^+$  and  $x_{ij}^-$  can be computed for a given  $x_i$  from the equations :

$$x_{ij}^+ - x_{ij}^- = d_j - x_i, \quad x_{ij}^+ \cdot x_{ij}^- = 0, \quad x_{ij}^+ \text{ and } x_{ij}^- \geq 0.$$

We will now introduce a Lemma to prove a special property of the solution to the problem P-8. Assume, without loss of generality, that the  $d_j$ 's are ordered such that  $d_j \leq d_{j+1}$ ,  $j=1, \dots, n-1$ .

Lemma 4.7.1 : Either  $x_i^* = d_1$  or  $x_i^* = d_n$  is a solution to the problem P-8.

Proof : Let an optimal solution be  $x_i^* = d_r$  for some  $r \in \{1, \dots, n\}$ . Then,

$$\sum_{j=1}^n (a_{ij}x_{ij}^+ + b_{ij}x_{ij}^-) = \sum_{j=r}^n a_{ij} (d_j - d_r) + \sum_{j=1}^r b_{ij} (d_r - d_j).$$

Let,  $p^r = \sum_{j=r}^n a_{ij} (d_j - d_r)$  and  $q^r = \sum_{j=1}^r b_{ij} (d_r - d_j)$ .

Now consider the relaxed problem where  $x_i \in [d_1, d_n]$ .

$$\text{Let } z(x_i) = \sum_{x_i \leq d_j} a_{ij} (d_j - x_i) + \sum_{x_i \geq d_j} b_{ij} (x_i - d_j), \text{ and let}$$

$$z_1(x_i) = \sum_{x_i \leq d_j} a_{ij} (d_j - x_i) \text{ and } z_2(x_i) = \sum_{x_i \geq d_j} b_{ij} (x_i - d_j).$$

Notice that both  $z_1$  and  $z_2$  are piecewise linear functions

of  $x_i$ , with breakpoints at  $x_i = d_1, \dots, d_n$ . Also, both  $z_1$  and  $z_2$  are convex functions because the slope of the piecewise linear portions is monotonically increasing for both  $z_1$  and  $z_2$ . Note that  $a_{ij}$  and  $b_{ij}$  are  $\geq 0$  for all  $i$  and  $j$ . It is clear that the functions  $z_1$  and  $z_2$  can be obtained by joining the points  $p^r, r=1, \dots, n$ , and the points  $q^r, r=1, \dots, n$  by line segments, respectively.

Since  $z_1$  and  $z_2$  are convex,  $z = z_1 + z_2$  is also convex. The maximum value of  $z$  over  $x_i \in [d_1, d_n]$  will therefore occur at either  $x_i^* = d_1$  or  $x_i^* = d_n$ .

Now, if the solution to the relaxed problem with  $x_i \in [d_1, d_n]$  is such that  $x_i^* = d_1$  or  $x_i^* = d_n$ , surely the solution to the problem where  $x_i \in \{d_1, \dots, d_n\}$  will also be the same. Hence the lemma is true in general.

To find whether the solution to problem P-8 is  $x_i^* = d_1$  or at  $x_i^* = d_n$ , the quantities  $p^1$  and  $q^n$  are computed. If  $p^1 > q^n$  then  $x_i^* = d_1$ , and if  $p^1 < q^n$  then  $x_i^* = d_n$ . It is extremely unlikely that  $p^1 = q^n$  since the quantities  $a_{ij}$  and  $b_{ij}$  which are obtained from the  $u_i$ 's and the  $\bar{v}_j$ 's are totally unrelated.

Corollary 4.7.2 : The solution to problem P-8 is unique if  $p^1 \neq q^n$ .

Proof : Consider the relaxed problem with  $x_i \in [d_1, d_n]$  .

By the Lemma 4.7.1, either  $x_i^* = d_1$  or  $x_i^* = d_n$  solves

P-8. For  $x_i^* = d_1$ ,

$$z(d_1) = \sum_{j=1}^n a_{ij} (d_j - d_1) \quad \text{and,}$$

for  $x_i^* = d_n$

$$z(d_n) = \sum_{j=1}^n b_{ij} (d_n - d_j).$$

By the assumption,  $p^1 \neq q^n$  or  $z(d_1) \neq z(d_n)$ . Hence,

both  $x_i^* = d_1$  and  $x_i^* = d_n$  cannot be optimal. Without

loss of generality, suppose  $x_i = d_1$  is optimal, i.e.,

$z(d_n) < z(d_1)$  . Now, since  $z$  is convex,

$$z(x_i) \leq \lambda z(d_1) + (1 - \lambda) z(d_n)$$

where,  $x_i = \lambda d_1 + (1 - \lambda) d_n$  for  $\lambda \in (0, 1)$ .

Since  $z(d_1) > z(d_n)$ ,  $z(x_i) < z(d_1)$  for  $x_i \in (d_1, d_n]$  .

Therefore,  $d_1$  is the unique solution to problem P-8.

Recall that our primary interest is to solve the problem P-5 which is restated below :

$$\text{P-5 :} \quad \begin{array}{ll} \text{Min} & \text{Max} \\ u \geq 0 & z \in S \end{array} \quad u^t (g^s(z) - 1)$$

Given a  $u \geq 0$ , we can solve the problem  $\text{Max}_{z \in S} u^t (g^s(z) - 1)$

using Lemma 4.7.1. Let  $\theta(u) = \text{Max}_{z \in S} u^t (g^s(z) - 1)$ .

Now, we are interested in solving  $\text{Min}_{u \geq 0} \theta(u)$ . If  $\theta(u)$  is

differentiable, a reduced gradient algorithm can be readily used, which is known to be convergent. (see Chapter 8, (16)).

But, differentiability of  $\theta(u)$  is assured by the Corollary 4.7.2 and Corollary 1 of Theorem 5 in Chapter 8 of (16).

Hence, to solve P-8 use is made of the following procedure based on the reduced gradient algorithm to update the  $u$  vector at each step.

Step 1: Initially set  $u_i = 1$  for  $i=1, \dots, s$ . Solve P-6 to obtain  $\bar{z}$  using Lemma 4.7.1. If  $g^s(\bar{z}) \not\geq 1$ , terminate with  $\bar{z}$  as a feasible extreme point of the set  $Z$ . If  $\theta(u) < 0$ , terminate with the conclusion that no feasible extreme point of the set  $Z$  remains. Otherwise go to step 2.

Step 2: Set  $v_i = g_i(\bar{z}) - 1$  for  $i = 1, \dots, s$ , but if at any step  $u_i = 0$  and  $g_i(\bar{z}) \geq 1$ , then  $v_i = 0$ . Replace  $u_i$  by  $u_i - kv_i$  where  $k = \text{Min}(Q, u_i/v_i)$  for all  $i$  such that  $v_i > 0$ , where  $Q$  is an arbitrary upper bound on  $k$ . Go to step 1.

In step 2, we have increased the component  $u_i$  for which  $g_i(z) < 1$  and decreased the component  $u_i$  for which

$g_i(z) \geq 1$ , taking care to ensure that  $u_i \geq 0$  for all  $i$ .

This algorithm is found to be very efficient, with

$u = (1, \dots, 1)$  as the initial  $u$  vector.

#### 4.8 Statement of the Complete Algorithm

Step 1 : Start with an extreme point of the set  $Z$ . Use the algorithm of Section 4.5 to generate a good starting extreme point.

Step 2 : Initially set  $s = 0$ . Find a weak pseudo global minimum  $(\bar{z}^s, \bar{w}^s)$  using the algorithm of Section 4.4. Let  $k =$  current best solution.

Step 3 : Replace  $s$  by  $s+1$ . Develop the cutting plane  $g_s(z) \geq 1$ , using the algorithm of Section 4.3.

Step 4 : Find a feasible extreme point of the set  $Z$  satisfying the cuts  $g_i(z) \geq 1$ ,  $i=1, \dots, s$ . If no such point exists, terminate. The problem is solved and the solution corresponds to the one for which the current best value  $k$  was obtained. Use the algorithm of section 4.7. Otherwise go to step 2.

#### 4.9 Convergence of the Algorithm

The cutting plane algorithm developed in this research can be split into three major sections. The first section deals with finding a weak pseudo global

minimum, the second with generating a cutting plane at this point and the third with finding a feasible extreme point of the set  $Z$ . We have shown that each one of these sections is convergent. We will now argue why only a finite number of iterations involving these three major sections will be required. Since, the number of extreme points is finite, and at each stage a valid cutting plane is developed which cuts off at least one extreme point, the number of cutting planes required will always be finite. Hence, the number of iterations, which corresponds to the number of cutting planes will always be finite. Thus, the entire algorithm is convergent.

#### 4.10 Illustrative Examples

In the two examples presented in this section, the algorithm to generate a good feasible solution has not been used in order to enable us to illustrate the existence of a pseudo global minimum different from the global minimum. If the above mentioned algorithm had been used, the first weak pseudo global minimum would have been the global minimum. Example 1 is worked out in greater detail than Example 2.

Example 4.10.1 : Locate two new facilities of capacity 70 and 80 units, and determine the allocation of resources

to six existing facilities located at  $(0,0)$ ,  $(1,0)$ ,  $(2,0)$ ,  $(3,0)$ ,  $(4,0)$  and  $(5,0)$ , with requirements of 10, 40, 30, 20, 20 and 30 units respectively.

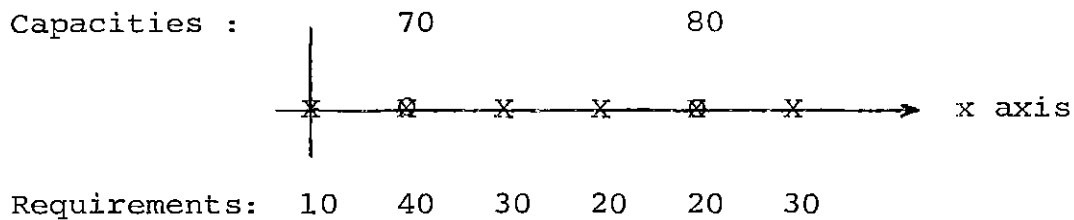


Figure 1. Location of New and Existing Facilities for Example Problem 4.10.1

Solution Procedure:

Step 1 (a) : Start at an extreme point of set  $Z$ , say  $(x_1, y_1) = (1, 0)$  and  $(x_2, y_2) = (4, 0)$ .

Step 1 (b) : Solve the transportation problem P-2 of Section 3.2 . The optimal tableau is given below, with the cost  $c_{ij} = |x_i - d_j| + |y_i - e_j|$  shown in the top left hand corner of each cell.

							$c_i$ 's
1	0	1	2	3	4		
10	40	20					70
4	3	2	1	0	1		
		10	20	20	30		80

$r_j$ 's : 10 40 30 20 20 30

The objective function value  $\phi = 100$ .

Step 1 (c) : Solve problem P-1 of Section 3.2 for  $(x_1, y_1)$ .

$$\begin{array}{rcccccc}
 d_j & : & 0 & 1 & 2 & 3 & 4 & 5 \\
 w_{ij} & : & 10 & 40 & 20 & 0 & 0 & 0 & \sum_{j=1}^6 w_{ij} = 70. \\
 T_j & : & 20 & 100 > 70. & & & & & \text{Hence, } x_1^* = 1.
 \end{array}$$

Since  $e_j = 0$  for all  $j$ ,  $y_1^* = 0$ .

Solve problem P-1 of Section 3.2 for  $(x_2, y_2)$ .

$$\begin{array}{rcccccc}
 d_j & : & 0 & 1 & 2 & 3 & 4 & 5 \\
 w_{ij} & : & 0 & 0 & 10 & 20 & 20 & 30 & \sum_{j=1}^6 w_{ij} = 80. \\
 T_j & : & 0 & 0 & 20 & 60 & 100 > 80. & & \text{Hence, } x_2^* = 4.
 \end{array}$$

Again, since  $e_j = 0$  for all  $j$ ,  $y_2^* = 0$ .

Step 1 (d) : Since the solution obtained in Step 1 (c) is the same as the one in step 1 (a), a local star minimum for this problem is given by the points obtained in Steps 1 (b) and (c) with  $\phi = 100$ .

Step 2 (a) : To find a pseudo global minimum, solve problem P-2 for each adjacent extreme point of the solution obtained in step 1 (c). If at any stage the objective

function value goes below  $\phi = 100$ , repeat the entire process with this point. The following adjacent extreme points of  $(x_1^*, y_1^*) = (1, 0)$  and  $(x_2, y_2^*) = (4, 0)$  were inspected.

Point # 1.  $(x_1, y_1) = (0, 0)$  and  $(x_2, y_2) = (4, 0)$   
 $\phi = 150.$

Point # 2.  $(x_1, y_1) = (2, 0)$  and  $(x_2, y_2) = (4, 0)$   
 $\phi = 130.$

Point # 3.  $(x_1, y_1) = (1, 0)$  and  $(x_2, y_2) = (3, 0)$   
 $\phi = 120.$

Point # 4.  $(x_1, y_1) = (1, 0)$  and  $(x_2, y_2) = (5, 0)$   
 $\phi = 120.$

Since the objective function value for problem P-2 is less than 100 at each adjacent extreme point of the solution obtained in steps 1 (b) and (c), that solution is a pseudo global minimum for this example.

Step 3 : The 4m non-basic variables for which  $\bar{\lambda}_j$ 's have to be computed are :  $x_{12}^+, x_{12}^-, y_{12}^+, y_{12}^-, x_{25}^+, x_{25}^-, y_{25}^+$  and  $y_{25}^-$ . Obviously, the  $\bar{\lambda}_j$ 's for the y variables is equal to infinity.

To compute  $\bar{\lambda}_j$  for  $x_{12}^+$

Solve : 
$$\lambda_j > 0 \quad \text{Max} \quad \text{Min}_{w \in W} \left[ (\bar{z} - \lambda_j \bar{e}^j)^t D w \geq 100 \right]$$

Making use of the physical interpretation of the parametric problem (see Section 4.3) the following transportation tableau is set up to compute  $\bar{\lambda}_j$ .

	$1-\lambda$	$0+\lambda$	$1+\lambda$	$2+\lambda$	$3+\lambda$	$4+\lambda$	$c_i$ 's
							70
	4	3	2	1	0	1	80
$r_j$ 's	10	40	30	20	20	30	

For  $\lambda_j = \infty$ , it is clearly seen that  $\phi = \infty$ , and hence  $\bar{\lambda}_j = \infty$ . Had the objective function value been less than 100, the Bolzano search procedure would have been used to compute  $\bar{\lambda}_j$ .

Similarly,  $\bar{\lambda}_j = 3.2$  for the variable  $x_{12}^-$ ,  
 $= 3.6$  for the variable  $x_{25}^+$  and  
 $= \dots$  for the variable  $x_{25}^-$ .

Hence, the first cutting plane is given by :

$$\frac{x_{12}^-}{3.2} + \frac{x_{25}^+}{3.6} \geq 1$$

Step 4 : Using the algorithm proposed in Section 4.7,  
and a starting u vector = ( 1 ),

Solve :            Maximize :  $1( x_{12}^-/3.2 + x_{25}^+/3.6)$   
                         subject to:  $z \in S$ .

The solution is given by  $(x_1, y_1) = (4, 0)$  and  $(x_2, y_2) = (0, 0)$ . Hence,  $x_{12}^- = 4$  and  $x_{25}^+ = 4$ , which satisfies the cut.

Step 5 : Starting from the point obtained in Step 4, find a weak pseudo global minimum using the algorithm proposed in section 4.4. The steps involved are essentially the same as those for finding a pseudo global minimum ( see step 2) except that at each point, the feasibility of all cuts must be maintained. Thus, the weak pseudo global minimum obtained at this step is at the points  $(x_1, y_1) = (4, 0)$  and  $(x_2, y_2) = (1, 0)$  , with an objective function value = 90.

Step 6 : Develop a cutting plane at the above weak pseudo global minimum. The following values for  $\bar{\lambda}_j$  were obtained :

$$\begin{aligned} \bar{\lambda}_j &= 3.4 \text{ for the variable } x_{15}^+, \\ &= 3.25 \text{ for the variable } x_{22}^- \text{ and} \\ &= \infty \text{ for the variables } x_{15}^-, x_{22}^+ \text{ and all } y\text{'s.} \end{aligned}$$

Hence, the second cutting plane is given by :

$$x_{15}^+ / 3.4 + x_{22}^- / 3.25 \geq 1.$$

Step 7 : Find an extreme point of set Z feasible to the two cuts obtained in steps 3 and 6 respectively. Again, it is found that a starting vector  $u = (1,1)$  yields a solution feasible to both cuts. This solution is given by :  $(x_1, y_1) = (5,0)$  and  $(x_2, y_2) = (5,0)$ .

Step 8 : Repeat the sequence of steps from 5 to 7. The following weak pseudo global minima and the corresponding cutting planes were obtained:

$$(a) \quad (x_1, y_1) = (5,0) \text{ and } (x_2, y_2) = (5,0)$$

$$x_{16}^+ / 3.9 + x_{26}^+ / 3.4 \geq 1$$

$$(b) \quad (x_1, y_1) = (0,0) \text{ and } (x_2, y_2) = (0,0)$$

$$x_{11}^- / 4.3 + x_{21}^- / 3.75 \geq 1$$

Step 9 : If the infeasibility criterion of the algorithm presented in Section 4.7 holds, terminate with the current best solution as the optimal solution. At this step, the above mentioned criterion holds, and hence the solution

$$(x_1^*, y_1^*) = (4,0) \text{ and } (x_2^*, y_2^*) = (1,0) \text{ along with the corres-}$$

ponding transportation tableau with an objective function value = 90, represents the optimal solution to this problem.

Example 4.10.2 : Locate two new facilities of capacity 100 units each and determine the allocation of resources to four existing facilities located at  $(0,0)$ ,  $(0,1)$ ,  $(1,1)$  and  $(1,0)$  with requirements 30,60,40 and 20 units respectively.

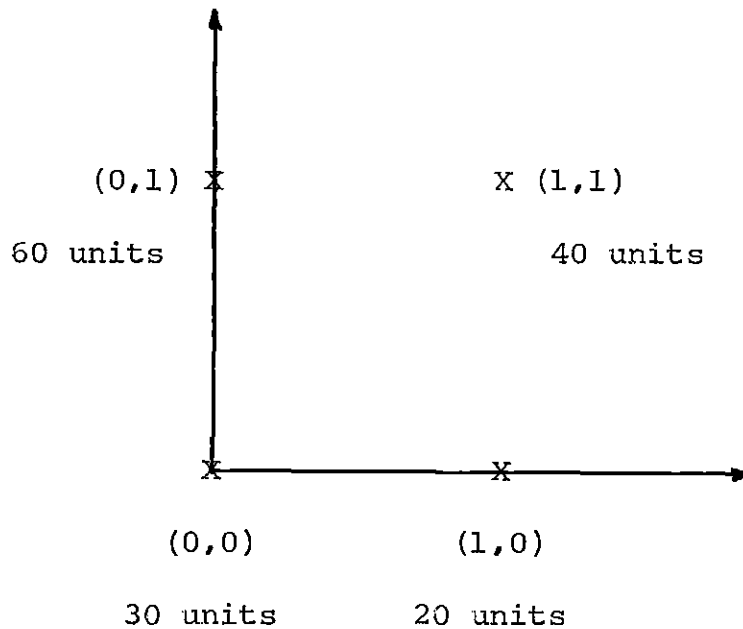


Figure 2. Locations of Existing Facilities  
for Example Problem 4.10.2

Solution Procedure :

Step 1 : Start with  $(x_1, y_1) = (0,1)$  and  $(x_2, y_2) = (0,0)$ .

Step 2 : The first weak pseudo global minimum which is also a pseudo global minimum is given by  $(x_1, y_1) = (0, 1)$  and  $(x_2, y_2) = (0, 0)$  with objective function value = 60.

Step 3 : The cutting plane at the above point is as follows:

$$x_{11}^-/1.4 + y_{13}^+/3 + x_{21}^-/1.2 + y_{21}^-/1.5 \geq 1$$

Since the capacities of the new facilities are the same, we can use the property mentioned at the end of Section 4.3 to obtain the following cut :

$$x_{21}^-/1.4 + y_{23}^+/3 + x_{11}^-/1.2 + y_{11}^-/1.5 \geq 1$$

Step 4 : The weak pseudo global minimum at this stage is given by  $(x_1, y_1) = (0, 1)$  and  $(x_2, y_2) = (1, 1)$  with an objective function value = 50.

Step 5 : The cutting plane at the above point is :

$$x_{11}^-/2.5 + y_{13}^+/1.6 + x_{23}^+/1.67 + y_{23}^+/1.4 \geq 1$$

Using the same property as in step 3, we can write down another cut as follows :

$$x_{21}^-/2.5 + y_{23}^+/1.6 + x_{13}^+/1.67 + y_{13}^+/1.4 \geq 1$$

Step 6 : Using the reduced gradient method of Section 4.7, the infeasibility criterion for the four cuts defined in steps 3 and 5 is found to be satisfied. Hence, the solution corresponds to the one obtained in step 4.

#### 4.11 The Unrestricted Capacity Problem.

A special case of the problem RDLAP occurs when the capacity constraints corresponding to the allocation part of the problem have the form :

$$\sum_{j=1}^n w_{ij} \leq c_i, \text{ where } c_i \geq \sum_{j=1}^n r_j \text{ for each } i=1, \dots, m.$$

In this case, the solution to the allocation part of the problem is greatly simplified.

Consider the problem RDLAP-2 of section 1.3 in which the location of new facilities is fixed. Let,

$$c_{ij} = x_{ij}^+ + x_{ij}^- + y_{ij}^+ + y_{ij}^-.$$

Then, RDLAP-2 can be written as :

TP-1 :            Minimize :  $\sum_{i=1}^m \sum_{j=1}^n c_{ij} w_{ij}$

subject to :         $\sum_{j=1}^n w_{ij} \leq c_i, \quad i=1, \dots, m$

$\sum_{i=1}^m w_{ij} = r_j, \quad j=1, \dots, n$

and  $w_{ij} \geq 0$  for all  $i$  and  $j$ .

If  $c_i \geq \sum_{j=1}^n r_j$  for each  $i \in \{1, \dots, m\}$ , we will show that the solution to the following relaxed problem is the same as that for TP-1.

$$\begin{aligned} \text{TP-2} \quad & \text{Minimize: } \sum_{i=1}^m \sum_{j=1}^n c_{ij} w_{ij} \\ & \text{subject to: } \sum_{i=1}^m w_{ij} = r_j, \quad j=1, \dots, n \\ & w_{ij} \geq 0 \text{ for all } i \text{ and } j. \end{aligned}$$

Let  $w_{ij}^*$  be an optimal solution to the problem TP-2.

$$\text{Then } \sum_{i=1}^m w_{ij} = r_j, \quad j=1, \dots, n$$

Therefore,  $w_{ij}^* \leq r_j$  for all  $i$  and  $j$ ,

$$\text{and } \sum_{j=1}^n w_{ij}^* \leq \sum_{j=1}^n r_j \leq c_i, \quad i=1, \dots, m.$$

Hence,  $w_{ij}^*$  solves TP-1.

To solve problem TP-2 we notice that it can be split into  $n$  subproblems corresponding to  $j=1, \dots, n$ . Each one of these subproblems can be solved as a knapsack problem. A typical subproblem has the form :

$$\begin{aligned}
 \text{TP-3(j)} : \quad & \text{Minimize : } \sum_{i=1}^m c_{ij} w_{ij} \\
 & \text{subject to } \sum_{i=1}^m w_{ij} = r_j \\
 & w_{ij} \geq 0, \quad i=1, \dots, m.
 \end{aligned}$$

$$\text{Let, } c_{kj} = \underset{i \in \{1, \dots, m\}}{\text{Minimize}} c_{ij}$$

$$\begin{aligned}
 \text{Then, } w_{ij}^* &= 0 \quad \text{if } i \neq k \\
 &= r_j \quad \text{if } i = k.
 \end{aligned}$$

Computational results for the rectilinear distance location-allocation problem with unrestricted capacity are given in the next chapter.

## CHAPTER V

## COMPUTATIONAL RESULTS AND CONCLUSIONS

Before giving computational results for the problem RDLAP, it may be worthwhile discussing the size of the Bilinear problem we are dealing with. For the problem with  $m$  new facilities and  $n$  existing facilities, we have seen in Section 1.3 that the number of variables were  $5mn + 2m$  whereas the number of constraints were  $4mn + m + n$ . We have also seen in Chapter III that the extreme points of the set  $Z$  are given by  $(x_i, y_i)$   $i=1, \dots, m$  such that  $(x_i, y_i)$  belongs to the set  $A_1 \times A_2$ , where  $A_1 = \{d_1, \dots, d_n\}$  and  $A_2 = \{e_1, \dots, e_n\}$ . If one was to attempt to solve the problem RDLAP by total enumeration, one would have to solve  $n^{2m}$  transportation problems corresponding to  $n^{2m}$  extreme points of the set  $Z$ . Even for small values of  $m$  and  $n$ , it is seen that the number of extreme points is very large and hence, such problems cannot be considered to be trivial.

5.1 Computational Results

In this research, we have developed an algorithm to solve exactly the rectilinear distance location-

allocation problem. In reporting our computational results, we first compare our algorithm with the one proposed by Morris (21), using as the basis of comparison, examples provided by Cooper in (3). Morris solves these problems using rectilinear distances, but his solution procedure is based on a discrete space formulation of the problem RDLAP. This fact enables us to compare the advantages and disadvantages of the discrete and continuous formulations and their respective solution procedures.

Table 3 gives the coordinates of the seven existing facilities for the six problems solved by Cooper in (3). Note that Cooper solves these problems using Euclidean distances. His solutions have been reproduced in columns 4 and 5 of Table 4. Cooper does not report computational times required to solve these problems. The requirement at each existing facility is 1 unit and the capacity at each new facility is 7 units. Table 4 presents computational results for the six problems of Table 3. Columns 4,5 and 8 have been reproduced from Morris' dissertation (21), and execution times reported in columns 6, 7 and 8 are in seconds on the Univac 1108. The transportation problems were solved using the simplified transportation algorithm of Section 4.4.

Table 3. Locations of Existing Facilities : Cooper's Problems.

( m=2 , n=7 )

Problem # 1	(15,15)	( 5,10)	(10,27)	(16, 8)	(25,14)	(31,23)	(22,29)
Problem # 2	( 6, 8)	( 6,32)	(20, 8)	(20,20)	(20,32)	(36, 8)	(36,32)
Problem # 3	( 8,12)	( 5,19)	( 5,26)	( 5,32)	(35,20)	(35,26)	(35,31)
Problem # 4	( 5,23)	( 9,32)	(15,23)	(21,32)	(26,23)	(31,32)	(16,12)
Problem # 5	( 6,31)	(13,24)	(13,31)	(20,24)	(20,17)	(27,17)	(27, 1)
Problem # 6	( 8,10)	( 8,26)	(11,20)	(17,15)	(17,22)	(24,17)	(31,19)

Table 4. Computational Results for Cooper's Problems

1	2	3	4	5	6	7	8
Problem #	Solution to Problem RDLAP	$\phi(z^*, w^*)$	Solution to Problem EDLAP	$\phi(z^*, w^*)$	$T_a$	$T_b$	T (Morris)
1	(22,27) (15,10)	59	(22,29) (15.2,12)	50.5	0.2	4.96	142.0
2	(20,32) (20, 8)	72	(20,32) (20,8.2)	71.5	0.15	4.07	99.0
3	(35,26) ( 5,20)	41	(35,26) (5.2,24.8)	38.3	0.11	2.07	156.0
4	(15,23) (21,32)	55	(12.1,23) (26,29.1)	48.9	0.22	6.14	211.0
5	(13,31) (20,17)	51	(13,27.5) (25.5,15.5)	36.2	0.12	3.0	149.0
6	(24,17) ( 8,22)	48	(24,17) (11,20)	38	0.21	5.77	213.0

$T_a$  : Time in seconds to obtain the global minimum for the problem RDLAP

$T_b$  : Time in seconds to cut off all the extreme points of the set Z in RDLAP

Columns 2 and 3 give the optimal location of the 2 new facilities and the

objective function value for RDLAP. Columns 4 and 5 give the same for EDLAP.

It is seen that the Euclidean solution forms a lower bound for the solution to the rectilinear distance problem, and in most cases, the location of the two new facilities is very close for the two problems. Also, the cutting plane algorithm is found to be very much more efficient than Morris' algorithm. Also, the time to obtain the global minimum lies in the range of about 5% of the total time required to cut off all the extreme points of the set  $Z$ . We will now study the latter aspect in greater detail.

An essential ingredient of this cutting plane algorithm was to "approximate" the feasible polytope by a cone formed by the rays incident on an extreme point. It has been conjectured on the basis of computational results in (20) and (26) that in general, although the "approximation" of the feasible region by the cone becomes poor as the dimensions of the problem increases, yet the global minimum of the problem is usually obtained in the early stages of the implementation of the algorithm. This is seen to be true for Cooper's examples, and we will further investigate this property for randomly generated problems. Also, we will now focus some attention on the unrestricted capacity problem.

Table 5 summarizes computational results corresponding to changes in the value of  $n$  for fixed  $m$ , where as

Table 4 exhibits computational results corresponding to changes in  $m$  for a fixed  $n$ . All times quoted are in seconds on the Univac 1108, excluding input and output. The data was randomly generated using the following program with the corresponding "seed" numbers.

```

SUBROUTINE RANDG(ISEED,NRAND)
ISEED = ISEED*131075
IF(ISEED.LT.0) ISEED = -ISEED
NRAND = ISEED*(.2910383E-10)*20
RETURN
END

```

Problem Size ( m X n )	ISEED
2 X 5	4563217
2 X 7	8654237
2 X 9	5464237
3 X 7	4562327
4 X 7	5643321

The requirement at each existing facility was fixed at 1 unit and the capacity of each new facility at  $n$  units.

Each problem was solved twice, once using the simplified code of section 4.4 and the second time using an Out of Kilter code to solve the same transportation problems. Our objectives in doing so is to stress the fact that the efficiency of the algorithm developed in this research depends to a large extent on the transportation code used.

Table 5. Computational Results for  $m = 2$

1	2	3	4	5 *		6 *	
n	Number of variables $5mn + 2m$	Number of constraints $4mn + m + n$	Number of extreme points $\frac{2m}{n}$	$T_a$	$T_b$	$T_a$	$T_b$
5	54	47	625	0.107	4.3	1.5	30.3
7	74	65	2401	0.121	7.5	3.2	95.8
9	94	83	6561	0.153	9.8	3.4	132.1

Table 6. Computational Results for  $n = 7$

1	2	3	4	5 *		6 *	
m	Number of variables $5mn + 2m$	Number of constraints $4mn + m + n$	Number of extreme points $\frac{2m}{n}$	$T_a$	$T_b$	$T_a$	$T_b$
2	74	65	2401	0.121	7.5	3.2	95.8
3	111	94	117649	0.176	33.8	3.5	370.1
4	148	123	5764801	0.231	59.7	3.7	480

\* See page 86

Execution times reported in column 5 of Tables 5 and 6 are in seconds and were obtained when using the unrestricted capacity transportation code. Execution times reported in column 6 of Tables 5 and 6 are also in seconds and were obtained when using an Out of Kilter code to solve the transportation problem.  $T_a$  denotes the execution time to obtain the global minimum, whereas  $T_b$  denotes the time to cut off all the extreme points of set  $Z$ .

To indicate the state of the art in solving transportation problems and to illustrate the fact that the Out of Kilter code used in this research performed very poorly as compared to the codes outlined in (23), we present the following tables of computational times for solving transportation problems of various sizes.

Table 7 summarizes execution times from (26) for restricted capacity transportation problems with  $0.2 < F \leq 0.5$  where,

$$F = \left( \sum_{i=1}^m c_i - \sum_{j=1}^n r_j \right) / \sum_{i=1}^m c_i$$

represents the "oversupply" for the problems.

Table 8 summarizes execution times obtained for

an Out of Kilter code.

Table 7. Computational Results for an  
Efficient Transportation Code.

Size of the problem ( m X n )	Execution times reported in (26) for $0.2 < F \leq 0.5$ in seconds
100 X 100	4.012
120 X 120	5.986
140 X 140	4.733
160 X 160	7.685

Table 8. Computational Results for an  
Out of Kilter Code

Size of problem ( m X n )	Execution time using an Out of Kilter code, in seconds with $F = 0.5$
5 X 10	1.58
10 X 10	2.51
15 X 10	4.63
20 X 10	6.02

As a consequence of the relatively large execution times required to solve the transportation problems when using the Out of Kilter code, computational results for the problem RDLAP with restricted capacities have not been provided. Note that the algorithm developed in this research is perfectly capable of solving such problems and we expect that the execution times for the restricted capacity problems will be as attractive as those obtained for the unrestricted problem, if an efficient transportation code of the type given in (23) is used.

We will now study computational times obtained for large size rectilinear distance location-allocation problems with unrestricted capacity. The program was terminated after six iterations. Table 9 shows the value of the current best solution at every iteration and the objective function value for the starting solution. Computational times at each iteration have also been indicated.

### 5.2 Summary and Conclusions

In this research, we have developed an algorithm to solve exactly the rectilinear distance location-allocation problem. The main trend of thinking in this study was guided by the cutting plane method to solve Bilinear Programming problems (25) . In proceeding towards this goal, we have

Table 9. Computational Results for Large Size Rectilinear  
Distance Location - Allocation Problems

Problem Size : m X n	1		2		3		4		5		6		
	$k_0$	$k_1$	$t_1$	$k_2$	$t_2$	$k_3$	$t_3$	$k_4$	$t_4$	$k_5$	$t_5$	$k_6$	$t_6$
5 X 20	106	87	1.27	81	7.03	81	13.48	81	21.78	81	28.35	81	38.94
10 X 30	98	63	2.4	63	10.3	63	16.5	63	28.1	63	39.5	63	50.6
20 X 50	156	92	4.2	80	15.1	74	29.7	74	39.1	74	50.9	74	72.6

$k_0$  : objective function value for the starting solution

$k_i$  : objective function value at the current best solution for iteration i

$t_i$  : execution time in seconds to iteration i ( cumulative).

obtained several worthwhile results related to the area of primary investigation. The main accomplishments and results are elaborated below.

First, we developed a primal simplex based algorithm to solve the rectilinear distance location problem and demonstrated how to characterize the simplex tableau at any iteration. Later we also characterized the extreme points and the adjacent extreme points of the location part of the problem. The developments were then effectively utilized to achieve tremendous simplification in some of the algorithms to follow.

Next, working along the lines of the cutting plane algorithm for Bilinear programming problems, we formed the feasible polytope over which the global minimum is known. We showed that only  $4m$  parametric problems have to be solved as compared to the  $4m(n+1)$  problems required in general to define the cutting plane. We used the Bolzano search to solve fairly efficiently these parametric transportation problems.

To take advantage of the special structure of the location part of the problem, we defined a "weak pseudo global minimum" and proved that although, as the name suggests, this point did not have all properties of a

pseudo global minimum, yet a valid cutting plane could be generated from it.

To initiate the algorithm to find a weak pseudo global minimum, or to obtain an indication that all the extreme points of the location set have been cut off, we developed a method to find a feasible extreme point of the location set. Starting from such a point, we directly used the definition of a weak pseudo global minimum to devise a procedure to obtain a weak pseudo global minimum. And lastly, we introduced an algorithm to find a good starting solution. Computational results show that in nine out of ten problems for which this algorithm is implemented, the first weak pseudo global minimum which is also a pseudo global minimum turns out to be the global minimum for the problem.

On comparing our algorithm to the one proposed by Morris (21) on the basis of the computational results presented in Table 4, it is clearly seen that our algorithm performed 20 to 40 times more efficiently than the algorithm proposed by Morris. What is more interesting is the fact that the time required to find the global minimum was less than 5% of the time to cut off all the extreme points. It is interesting to note that preliminary computational experience on the use of cones to form the

feasible polytope, reported in (20) and (26), indicate that in general, the global minimum of a nonconvex minimization problem is usually obtained in the early stages of the implementation of the algorithm.

Computational results obtained in Table 9 clearly indicate that for large size problems, the cutting plane algorithm may be terminated prematurely so as to conserve the computational time required and yet not sacrifice the exactness of the solution obtained.

We will now discuss the effect of  $m$ , the number of new facilities and  $n$ , the number of existing facilities on computational time required to solve exactly this problem, which depends to some extent on the number of extreme points  $n^{2m}$ . The effect of doubling  $m$  increases the number of extreme points by a factor of  $n^{2m}$ , whereas doubling  $n$  increases it by a factor of  $2^{2m}$ . Hence, one would expect the algorithm to be more sensitive to an increase in  $m$  rather than  $n$  for  $n > 2$ . Computational results show that this conjecture is generally true.

And lastly, we will characterize certain problems as "easy" or "hard" to solve when using the algorithm developed in this dissertation. Problems with a number of existing facilities having the same  $x$  or  $y$  coordinate are

"easy" since the number of adjacent extreme points are greatly reduced. Also, problems with new facilities having the same capacity are "easy" to solve because, at every stage a few additional cuts can be defined without having to solve any parametric problems (see end of Section 4.3). Problems which do not have either one of these properties are "hard" to solve exactly in a reasonable amount of time. Obviously, the unrestricted capacity problem is "easier" to solve as compared to the restricted capacity problem.

### 5.3 Recommendations for Further Research

The computational time required to solve the problem RDLAP depends to a very large extent on the efficiency of the code used to solve the allocation or transportation problem. In reporting some of our computational results, we have made use of an Out of Kilter code to solve the transportation problem. In light of the fact that computational times are very sensitive to the efficiency of the transportation code, we recommend that even if a code which is just marginally better than the Out of Kilter code is available, use be made of it.

Since location-allocation problems are generally "one-shot" problems in the sense that they do not have to

be solved repeatedly as say, scheduling or inventory problems, one can rationalize the use of a code which requires a considerable amount of time to obtain the exact solution to the problem. On the basis of computational results presented in Section 5.1, we strongly recommend to the user who is willing to take a very small chance of not obtaining the exact solution, that he terminate the process after 3 to 4 iterations, which typically represents 2 to 5 % of the total time required to solve the problem exactly. Even if the solution obtained after 3 to 4 iterations is not the exact solution, it will probably be very close to it.

A very interesting aspect of the algorithm we have developed is that no special property of the allocation part of the problem aside from the fact that its constraints are linear and separable has been used. This enables us to substitute directly any other appropriate problem with the above property, in place of the allocation problem without having to modify any other part of the algorithm developed in this research. If the substituted problem has variables which are restricted to integer values only, a double cutting plane algorithm may be devised, where at each stage, the variables of the substi-

tuted problem are no longer restricted to integer values.

As part of our recommendation for further research, we suggest that some of the other methods outlined in (25) such as polytope generation be studied and the execution times compared with the ones obtained in this research for the polar cuts. Also, we recommend that further research be carried out in order to find an efficient lower bound for the rectilinear distance location-allocation problem so that the cutting plane algorithm can be terminated when the current solution lies within a fraction of this lower bound.

The location-allocation problem with interaction between new facilities can also be solved using the algorithm of Section 4.1. For simplifications resulting from the special structure of the problem, we recommend that an approach similar to the one adopted in this research be attempted.

## BIBLIOGRAPHY

1. Bellmore, H. and H.D. Ratcliff, "Set Covering and Involutory Bases," Mgt. Sci. Vol. 18, No. 3, 1971, pp. 194 - 206.
2. Cooper, L., "Location-Allocation Problems," Operations Research, Vol. 11, No. 1, 1963, pp 331 - 343.
3. Cooper, L., "The Transportation-Location Problem," Operations Research, Vol. 20, No. 1, 1972, pp 94 - 108.
4. Efroymsen, M.A. and T.L. Ray, "A Branch and Bound Algorithm for Plant Location," Operations Research, Vol.14, No. 3, 1966, pp 361 - 368.
5. Eyster, J.W., J.A. White and W.W. Wierwille, "On Solving Multifacility Location Problems Using a Hyperboloid Approximation Procedure," AIIE Trans., Vol. 5, No. 1, 1973, pp 1 - 6.
6. Francis, R.L. and J.A. White, "Facility Layout and Location - An Analytical Approach," Prentice Hall, Inc. 1974.
7. Geoffrion, A.M., "Primal Resource - Directive Approach for Optimizing Nonlinear Decomposable Systems," Working Paper 141, The RAND Corp., 1968.
8. Geoffrion, A.M. and G.W. Graves, "Multicommodity Distribution System Design by Bender's Decomposition," Mgt. Sci., Vol. 20, No. 5, Jan. 1974, pp 872 - 838.
9. Hillier, F.S. and Lieberman, G.T., "Introduction to Operations Research," Holden-Day, 1972.
10. Ignizio, J.P., "A Heuristic Solution to the Generalized Covering Problem," Unpublished Ph.D. dissertation, Virginia Polytechnic Institute and State Univ., Blacksburg, Va., 1971.

11. Khumawala, B.M., "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem," Mgt. Sci., Vol. 18, No. 12, 1972, pp 718 - 731.
12. Kuehn, A.A. and M.J. Hamburger, "A Heuristic Program for Locating Warehouses," Mgt. Sci., Vol. 9, No. 4, 1963, pp 643 - 666.
13. Kuenne, R.E. and R.M. Soland, "The Multisource Weber Problem : Exact Solutions by Branch and Bound," IDA Economic Papers (H.Williams, ed.), Program Analysis Division, Arlington, Va., 1971.
14. Kuhn, H.W., "On a Pair of Dual Nonlinear Problems," J. Abadie (ed.), Nonlinear Programming, John Wiley, Interscience, N.Y., 1967.
15. Kuhn, H.W. and R.E. Kuenne, "An Efficient Algorithm for the Numerical Solution of the Generalized Weber Problem in Spatial Economics," Journal of Regional Science 4, 1962, pp 21 - 34.
16. Lasdon, L.S., "Optimization Theory for Large Systems," The Macmillan Company, 1970, pp 406 - 435.
17. Lawler, E.L., "Covering Problems : Duality Relations and a New Method of Solution," Journal of the SIAM on Applied Math, Vol. 14, No. 5, 1966, pp 1115 - 1132.
18. Martos, B., "Subdefinite Matricies and Quadratic Forms," SIAM Journal on Applied Math, Vol. 17, 1969, pp 1212 - 1223.
19. Martos, B., "The Direct Power of Adjacent Vertex Programming Methods," Mgt. Sci., Vol. 17, 1965, pp 241 - 252.
20. Moreno, C., "Minimization of Concave Functions subject to Linear Constraints," Operations Research Center Report ORC 72-3, 1972, University of California, Berkeley.
21. Morris, J.G., "Facilities Location and Location-Allocation Using Rectangular Distances," Unpublished Ph.D. dissertation, Univ. of Wisconsin, Graduate School of Business, 1972.

22. Revelle, C. and R. Swain, "Central Facilities Location," Geographical Analysis, Vol. 2, No. 1, 1970, pp 30 - 42.
23. Shetty, C.M., R.W. Langley and J. Kennington, "Efficient Computational Devices for the Capacitated Transportation Problem," Naval Research Logistics Quarterly, Vol. 21, No. 4, Dec. 1974, pp 637 - 647.
24. Toregas, C. and C. Revelle, "Optimal Location under Time or Distance Constraints," Paper of the Regional Science Association, Vol. 19, No. 6, 1971, pp 1364-1373.
25. Vaish, H., "Nonconvex Programming with applications to Production and Location Problems," Unpublished Ph.D. dissertation, Georgia Institute of Technology, 1974.
26. Zwart, P.B., "Computational Aspects on the Use of Cutting Planes in Global Optimization," Proceedings 1971 Annual Conference of the Association of Computing Machinery, Chicago, 1971, pp 457 - 465.