INTERACTIVE COMPUTER GRAPHICAL APPROACHES

TO

SOME MAXIMIN AND MINIMAX LOCATION PROBLEMS

By

DAVID JOHN BUCHANAN, B.SC., M.B.A.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

ISBN 0-315-42353-6

MAXIMIN AND MINIMAX LOCATION PROBLEMS

DOCTOR OF PHILOSOPHY (1988)
(Management Science/Systems)

MCMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Interactive Computer Graphical Approaches to some Maximin and Minimax Location Problems

AUTHOR: David John Buchanan, B.Sc. (McMaster University)
M.B.A. (McMaster University)

SUPERVISOR: Professor G. O. Wesolowsky

NUMBER OF PAGES: ix, 187

ABSTRACT

This study describes algorithms for the solution of several single facility location problems with maximin or minimax objective functions.

Interactive computer graphical algorithms are presented for maximizing the minimum rectilinear travel distance and for minimizing the maximum rectilinear travel distance to a number of point demands when there exist several right-angled polygonal barriers to travel. For the special case of unweighted rectilinear distances with barriers, a purely numerical algorithm for the maximin location problem is described.

An interactive computer graphical algorithm for maximizing the minimum Euclidean, rectilinear, or general l_p distance to a number of polygonal areas is described. A modified version of this algorithm for location problems with the objective of minimizing the maximum cost when the costs are non-linear monotonically decreasing functions of distance is presented. Extension of this algorithm to problems involving the minimization of the maximum cost when the costs are functions of both distance and direction is discussed using asymmetric distances.

ACKNOWLEDGMENTS

I would like to thank Dr. George O. Wesolowsky, my supervisor, for his encouragement, support and guidance in the development of this research.

To the rest of my supervisory committee, Dr. M. W. Luke Chan, Dr. Robert F. Love, and Dr. George Steiner, I also wish to express my appreciation.

Most of all, my thanks to Floyd J. Buchanan.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

CHAPTER 1

INTRODUCTION

1.1 Introduction to Facility Location Problems

Location models are used in operations research to determine the best location to place one or more new facilities in order to serve demands with known positions. The best location depends in some manner on the distances from the new facilities to the demands they serve. Variations in the appropriate measure of distance, constraints on the locations of new facilities, behaviour of costs as functions of distance, and the criteria used to evaluate solutions yield a rich variety of optimization problems. These models are of interest to many disciplines including engineering, economics, management, and geography.

1.1.1 Early History

Interest in location-type problems dates back at least to the early seventeenth century. In his essay on maxima and minima Fermat wrote: "Let he who does not approve of my method attempt the solution of the following problem: Given three points in the plane find a fourth point such that the sum of its distances to the three given points is a minimum." This was partially solved by

Torricelli prior to 1640 (Love, Morris, and Wesolowsky). Also studied by Jacob Steiner in the early nineteenth century (Courant and Robbins), this problem is known as the Steiner or Fermat problem.

1.1.2 The Weber Problem

The Weber problem is a generalization of the Steiner problem; there can be more than three given points and weights representing the amount of demand at the given points. The given points are usually termed "demand points" or, more simply, "demands" but may also be called "fixed points" or "existing facilities." This problem has served as a prototype for many of the location problems found in operations research. The Weber problem can be formulated as

$$\underset{X}{\text{minimize}} \quad f(X) = \sum_{i=1}^{N} w_i d(X,P_i) \qquad (1.1.1)$$

where

$X = (x,y)$ = location of new facility,

P_i = location of i'th demand point,

w_i = weight associated with i'th demand,

N = number of demand points,

and $d(X,P_i)$ = distance from X to P_i.

For obvious reasons, this problem is called a minisum location problem.

Although the Steiner problem has a simple geometric solution (Courant and Robbins), this is not the case for the Weber problem. Weiszfeld (1936) presented an iterative algorithm for the Weber problem which was subsequently rediscovered by Kuhn and Kuenne (1962). Like the Weber problem, most location models require iterative algorithms for their solution and to solve problems of realistic size, these algorithms must usually be implemented on an electronic computer.

A minisum location problem similar to the Weber problem but defined on a network is usually called a "median" problem (Handler and Mirchandani, p. 13).

1.2 Minimax and Maximin Criterions

Although the minisum criterion, minimizing the sum of weighted distances or costs, is the most common criterion used in the literature, it is not the only possible one. The first version of a location problem with a minimax criterion, that is minimizing the maximum of a set of weighted distances, was presented in 1857 by J. J. Sylvester. He offered the one sentence problem description (Love, Morris, and Wesolowsky): "It is required to find the least circle which shall contain a given set of points in the plane." A minimax location problem defined on a network is commonly called a "center" problem (Handler and Mirchandani, p. 80).

The maximin criterion, or maximizing the minimum of a set of weighted distances, is another possibility. This research examines certain minimax and maximin single facility location problems in the plane.

1.2.1 Justification

The appropriate criterion for evaluating locations depends on the particular application involved and the goals of the decision maker. The minisum criterion, minimizing the sum of weighted distances from the new facilities to the demands they serve, is appropriate when one wishes to reduce average transportation costs. For example, if the weights in the Weber problem are the numbers of trips per year to each of the demands, then the minisum criterion will minimize the total distance traveled per year.

The minimax criterion, since it minimizes the largest component of the total cost, may be useful when costs are known to increase rapidly with distance but the costs as functions of distance cannot be determined with reasonable accuracy. (Optimization using a minisum criterion implicitly involves trading-off cost increases with respect to some demands against cost decreases with respect to other demands. When costs are difficult to determine and the marginal costs cannot be assumed to be equal for different demands and distances, minimizing the sum of the distances may not be justified.) Thus, the minimax criterion has been suggested as appropriate for choosing the location of emergency services such as fire stations and ambulances (Elzinga and Hearn 1972a). In a sense, the minimax criterion can be interpreted as a "grease the squeaky wheel" policy because it minimizes the effects of the worst situation (Francis and White, p. 379).

The minimax criterion can also be appropriate in certain types of competitive situations. Consider the problem of positioning a security force at one of a number of targets so that it can respond as quickly as possible to an attack on any one of the targets. Assume that the opponent can launch only one attack and that he knows the position of the security force when he chooses his point of attack. If the opponent desires to maximize the time available at the target in which he can cause damage, then he will choose to attack the target furthest from the security force. Thus, the minimax criterion, which minimizes the maximum distance from the security force to any target, is the appropriate criterion to use in positioning the security force.

The maximin criterion, where the minimum distance is maximized, can be appropriate when locating noxious or hazardous facilities as far as possible from a given set of points. Since it will minimize the largest component of the total cost, the maximin criterion is convenient if the costs or hazards associated with a facility are expected to decrease rapidly with increasing distance but cannot be determined with reasonable accuracy. The related maxisum criterion, in which we maximize the sum of the distances to the given points, could be used if the costs are expected to decrease linearly with increasing distance from the noxious facility.

The maximin criterion, like the minimax, can be appropriate for evaluating locations in certain competitive situations. For example, if the attacker in the preceding discussion is confronted by several security forces, then he should choose as his point of attack that target which is as far as possible from the nearest security force.

The minimax criterion has also been advocated on grounds of equity because it improves the worst case service and reduces the disparity among service recipients (often substantially increasing the cost of providing a given average level of service).

1.2.2 Formulation

A fairly general formulation of the constrained single facility minimax problem in the plane is

$$\underset{X \text{ in } F}{\text{minimize}} \quad \{ \underset{i=1,\ldots,N}{\text{maximum}} \quad f_i(X) \} \tag{1.2.1}$$

where

$f_i(X) = w_i d(X,P_i)$,

$X = (x,y)$ = location of new facility,

$P_i = (a_i,b_i)$ = location of i'th demand point,

w_i = weight associated with i'th demand,

F = set of feasible new facility locations, and

$d(X,P_i)$ = distance from X to P_i.

The distance function $d(X,P_i)$ depends on the particular application. If travel is confined to a rectangular street grid or aisles in a warehouse such that each street or aisle is parallel to either the x or y axis, then the rectangular, rectilinear or "Manhattan" distance,

$$d(X,P_i) = |x-a_i| + |y-b_i|, \qquad (1.2.2)$$

is a more appropriate measure of travel distance than the straight-line or Euclidean distance,

$$d(X,P_i) = [(x-a_i)^2 + (y-b_i)^2]^{1/2}. \qquad (1.2.3)$$

The Euclidean and rectilinear distance metrics (norms) are special cases of the more general l_p distance metric which is also used to model travel distances. The l_p distance function is

$$d(X,P_i) = [|x-a_i|^p + |y-b_i|^p]^{1/p}, \quad p \geq 1. \qquad (1.2.4)$$

A maximin problem formulation can be obtained by replacing "minimize" by "maximize" and "maximum" by "minimum" in (1.2.1)..

1.2.3 Relation to Covering Problems

Minimax and maximin location problems are closely related to covering problems. The unweighted (all w_i equal) minimax location problem with Euclidean distances can be interpreted as finding the center and radius of the smallest circle which encloses all of the fixed points P_i (see Figure

1.1). In more than two dimensions, this is also known as "the minimum covering sphere problem" (Elzinga and Hearn 1972b). The unweighted maximin location problem with Euclidean distances can be interpreted as finding the largest circle with its center in a given feasible region which does not contain any of the P_i (see Figure 1.2).

For weighted problems (not all w_i equal), these simple interpretations no longer apply. The weighted maximin location problem with Euclidean distances can be interpreted as finding the largest value of z such that the union of the set of discs with radii z/w_i and centers at the P_i does not cover all of the points in the feasible region F. Any feasible point which is not covered by at least one of the discs defined by this optimal value of z is an optimal location (see Figure 1.3). This interpretation can also be applied to unweighted maximin problems where all of the w_i are equal. When the unweighted maximin problem is interpreted as a covering problem in this manner, an extensive related literature exists in the area of the geometry of numbers, according to Dasarathy and White (1980). They note that a good discussion of the mathematical theory of this type of covering can be found in Rogers (1964).

The weighted minimax location problem with Euclidean distances can be interpreted as finding the smallest value of z such that the set of discs with radii z/w_i and centers

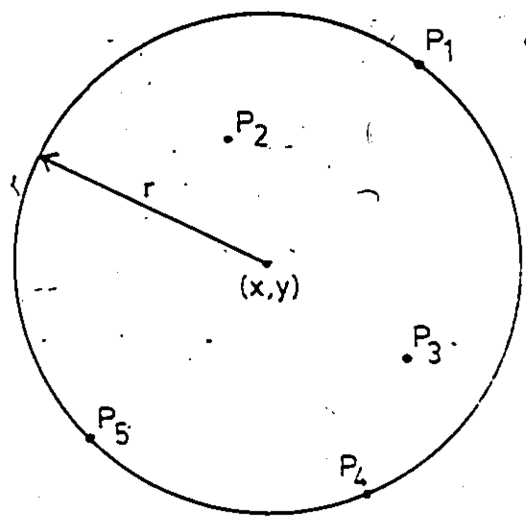

Figure 1.1. Covering interpretation of unweighted minimax.

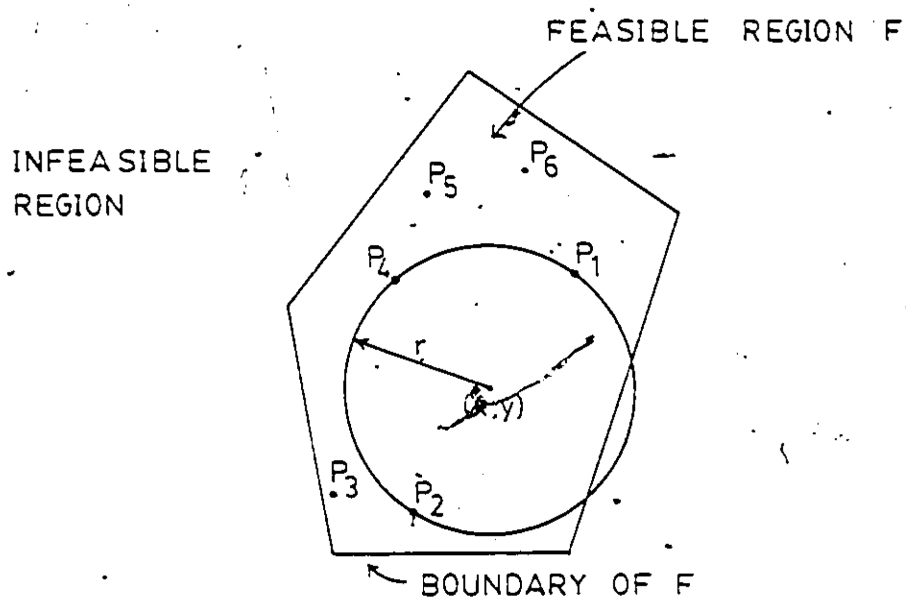

Figure 1.2. Covering interpretation of unweighted maximin.

at the P_i has a non-empty intersection (see Figure 1.4). The optimal location is the smallest possible non-empty intersection that can be obtained by varying z (Francis 1967). This geometric interpretation can be converted to a more convenient covering interpretation similar to the interpretation of the weighted maximin problem by using an identity from set theory: namely, that the intersection of several sets is identical to the complement of the union of their complements. Thus, the weighted minimax location problem with Euclidean distances can also be interpreted as finding the smallest value of z such that the union of the complements of the discs with radii z/w_i and centers at the P_i does not cover the entire feasible region. Any feasible point which is not covered by the union of the complements of the discs defined by this optimal value of z is an optimal location.

Discrete location problems with a finite set of possible locations for the new facility or facilities are also closely related to set covering problems. A discussion of both can be found in Francis and White (1974).

Since this research is concerned with certain single facility weighted minimax and maximin problems in continuous space, only the interpretations of weighted minimax and maximin problems as covering problems are particularly relevant.

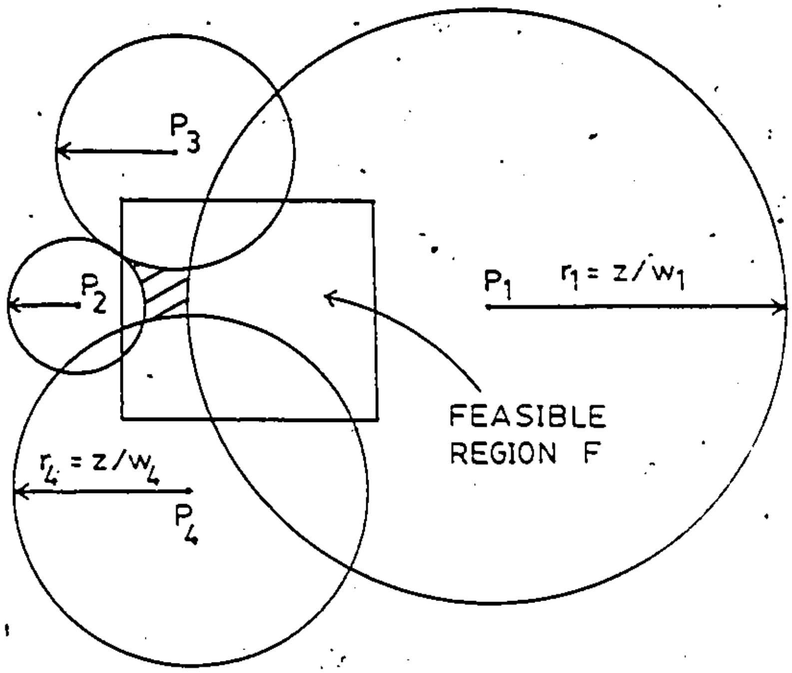

Figure 1.3. Covering interpretation of weighted maximin.

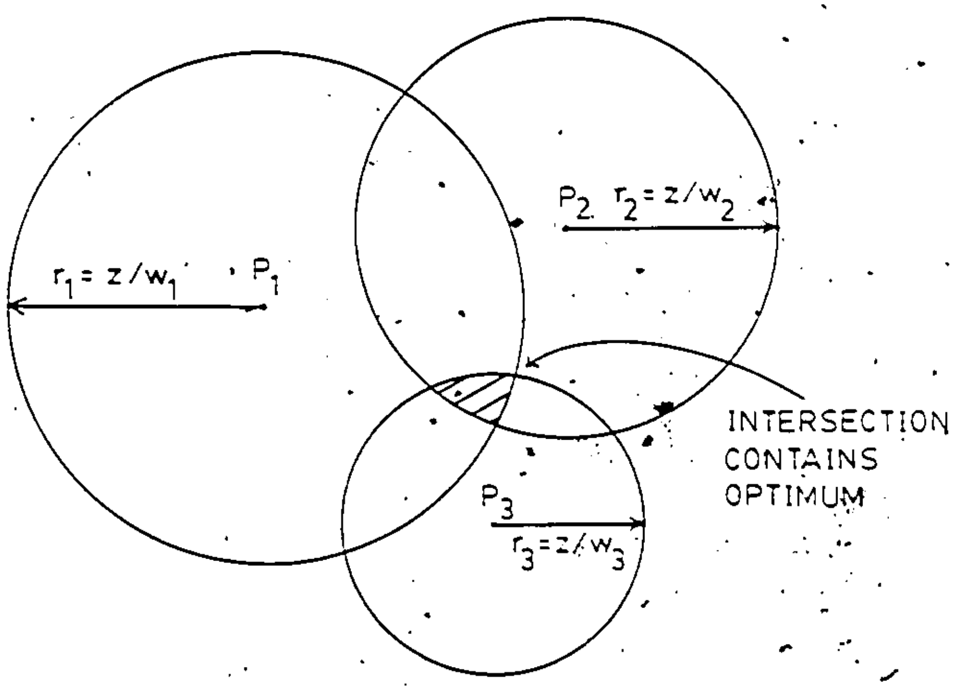

Figure 1.4. Covering interpretation of weighted minimax.

1.3 Barriers to Travel

Ironically, Larson and Li (1981) noted that the rectilinear or "Manhattan" distance metric is inadequate as a measure of travel distances in Manhattan due to barriers to travel like Central Park. They pointed out that such barriers to travel are often found in transportation and plant layout problems and they presented an algorithm (POLYPATH) to find the minimum rectilinear distance path in the presence of polygonal barriers.

1.3.1 Location with Barriers

Although barriers to travel occur frequently in practical problems they have received very little attention in the literature of location problems in continuous space. Network location problems can take barriers into consideration when the distances between the nodes of a network are defined. Larson and Sadiq (1983) considered a continuous space minisum problem with rectilinear distances and polygonal barriers to travel which they reduced to a p-median problem on a finite network.

The only other work on location problems with barriers to travel in continuous space is that of Katz and Cooper (1979a, 1979b, and 1981), Muralimohan and Babu (1983a, 1983b), and Ravindranath, Vrat, and Singh (1985). Katz and Cooper studied single facility minisum problems with Euclidean distances and one or more circular barriers.

Muralimohan and Babu studied single facility minisum problems with Euclidean distances and at most two non-intersecting convex barriers. Ravindranath et al. considered a single facility location problem with rectilinear distances and only one rectangular or circular barrier to travel.

Barriers to travel which are right-angled polygons aligned with the travel directions (streets or aisles) are a type of barrier which is quite common in practical applications. For example, the boundaries of parks and zones where truck traffic is prohibited usually follow a street grid and hence are right-angled polygons aligned with the travel directions. Interpreting the $d(X,P_i)$ in (1.2.1) as the minimum rectilinear distance in the presence of such right-angled barriers yields one of the location problems studied in this research.

1.3.2 Barriers versus Infeasible Regions

A facility located in the interior of a barrier cannot be reached and thus a barrier to travel is for practical purposes also an infeasible region for facility location. In a given application, there may also exist regions where it is infeasible to locate a facility (zoning restrictions etc.) but travel through the region is allowed. Location problems having this type of infeasible region for facility placement are known as constrained location problems and, in contrast to barrier problems, have received considerable attention.

1.4 A Graphical Approach to Some Location Problems

Although mechanical and electrical analogues have been used, most location models are solved using numerical methods. (For discussions of analogue methods for planar problems see Francis and White or Eilon et al. and see Litwhiler for applications to problems on the sphere.) This section discusses a graphical method of solving some minimax and maximin problems. In order to illustrate the graphical method, a simple single facility location problem with Euclidean distances is discussed. Neither the method nor this particular application are new but they provide an introduction to the methodology developed in the succeeding chapters.

1.4.1 Previous Graphical Approaches

Francis (1967) gave a geometrical characterization of the optimum for minimax location problems and remarked that since the level sets in the Euclidean and rectilinear cases can be constructed graphically the optimum can be determined by graphical methods. (The level set of a function for the value t is the set of points for which the value of the function is less than or equal to t. The level sets of Euclidean and rectilinear distance functions are circular discs and squares, respectively.)

Brady and Rosenthal (1980) presented an interactive computer graphical algorithm for optimally solving constrained minimax location problems with rectilinear, Euclidean, or mixed (rectilinear and Euclidean) distance norms. Brady, Rosenthal, and Young (1983) extended this method to the multifacility case by relying more heavily on the pattern recognition ability of the operator.

Hansen, Peeters, and Thisse (1981) presented a graphical method (BLACK AND WHITE), similar to Brady and Rosenthal (1980), for solving the problem of locating one obnoxious facility when the costs incurred by the fixed points (demand points) decrease with increasing distance from the facility and the objective is to minimize the maximum cost. Their method was implemented manually.

Melachrinoudis and Cullinane (1986) presented a simple interactive computer graphic method using slowly expanding circles for locating an undesirable facility in a non-convex feasible region. The noxious effects of the facility were assumed to be proportional to the inverse of the square of the Euclidean distance from the facility to the fixed points.

This study extends the single facility graphical approach to several unsolved location problems. In these problems the determination of the level curves of the $d(X,P_i)$, required by the graphical approach, is a non-trivial problem by itself.

1.4.2 A Simple Example

Consider the problem of locating a facility in a bounded region of the plane so that the minimum weighted Euclidean distance from the facility to any of a finite number of demand points is maximized. This problem can be formulated as

$$\underset{X \text{ in } F}{\text{maximize}} \quad z$$

subject to

$$f_i(X) = w_i d(X,P_i) \geq z; \quad i=1,\ldots,N$$

where

$X = (x,y)$ = location of new facility,

$P_i = (a_i,b_i)$ = location of i'th demand point,

w_i = weight associated with i'th demand,

F = a bounded region of the plane,

and

$$d(X,P_i) = [(x-a_i)^2 + (y-b_i)^2]^{1/2}.$$

The set of points which satisfy the i'th constraint for a given value of z is the complement of a disc with center P_i and radius z/w_i. The disc itself is the level set of the function $f_i(X)$ for the value z and is the set of points which violate the i'th constraint for the current value of z. The circle with center P_i and radius z/w_i is the level curve or contour line for the value z. The union of the N discs is the set of points which do not simultaneously satisfy all of the constraints for the current value of z. The union of the N discs is a "dominated region" in the terminology of Hansen et al. (1981) since all points in it have an objective function value less than the current feasible value of z. The optimum location or locations is the smallest set of feasible points lying outside the dominated region which we can obtain by varying z.

To determine the optimum location for the facility one could apply the following algorithm using paper and pencil or, more conveniently, computer generated graphics.

Step 1. <u>Initialization</u>

Choose the region of the plane which will be represented on the screen or graph paper in such a way that it contains the optimum location. In this example, the maximin solution must lie in the bounded feasible region so that any rectangle which encloses all of the feasible region can be used. (See Figure 1.5.)

Choose z_{int}, an initial value for the objective function, so that for all $i=1,\ldots,N$ and all points X in the region under consideration $f_i(X) \geq z_{int}$. In this example, let $z_{int} = 0$.

Set $k = 1$, $z_0 = z_{int}$.

Step 2. <u>Improve Value of Objective Function</u>

If the remaining unshaded region on the screen or graph paper has been reduced to a few points (pixels), then go to Step 4.

Otherwise, pick any unshaded point X_k, preferably one which lies near the center of an unshaded region. (See Figure 1.6.) Calculate $z_k = \min\{ f_i(X_k);\ i=1,\ldots,N\}$.

Step 3. Eliminate Dominated Regions

Shade the area between the level curves of $f_i(X)$ corresponding to the values z_{k-1} and z_k for each $i=1,\ldots,N$. In this example, shade the annulus with center P_i, radii z_{k-1}/w_i and z_k/w_i for each $i=1,\ldots,N$.

Set $k = k + 1$.

Go to Step 2.

Step 4. Accept or Expand Scale

If the size of the remaining unshaded area is sufficiently small so that the uncertainty in the location of the optimum is acceptable, then go to Step 5.

Otherwise, choose a small rectangle enclosing the unshaded region and expand the scale so that the rectangle fills the screen or graph paper.

Go to Step 2.

Step 5. Get Optimum

Pick a point in the remaining unshaded area as a near-optimal solution X^*. Calculate $z^* = \min\{ f_i(X^*);\ i=1,\ldots,N\}$. (See Figure 1.7.)

Stop.

In effect, one colours or shades every point X in the area under consideration for which at least one of the $f_i(X)$ is less than or equal to the current value of z. By

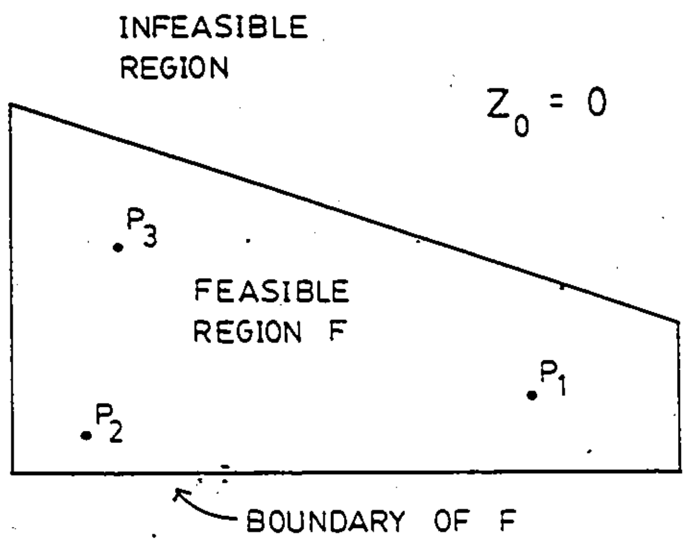

Figure 1.5. A simple example - Initialization stage.

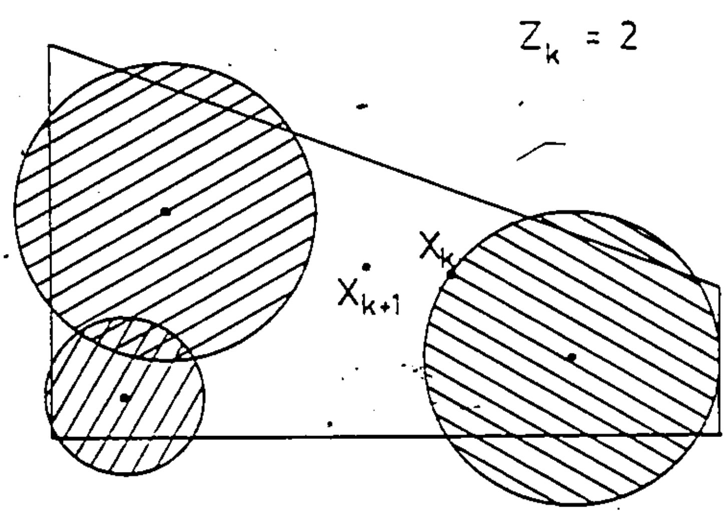

Figure 1.6. A simple example - Intermediate stage.

increasing z one is eventually left with only the optimum location (or locations) unshaded.

Since most bit-mapped computer graphic systems can quickly and accurately plot lines and circles there are obvious advantages to implementing this procedure using computer generated graphics rather than pencil and paper. When regions must be expanded to gain additional accuracy, a computer implementation is almost essential.

1.4.3 Advantages

Although much faster mathematical programming based algorithms are available for unconstrained minimax problems (Drezner and Wesolowsky 1980a) as well as for constrained minimax and maximin problems with special types of feasible regions (Drezner 1983), this simple graphical procedure has several inherent advantages when compared to strictly numerical methods; it is also extremely flexible.

Standard mathematical programming search methods are difficult to apply to optimization problems with non-convex or disjoint feasible regions since searches can be trapped in local optima at boundaries even when the objective function is convex. (Because the Euclidean, rectilinear, and general l_p distance functions are convex, the maximum of a set of such distance functions is convex.) These search methods are even more difficult to apply to maximin location problems, because the minimum of a set of convex functions

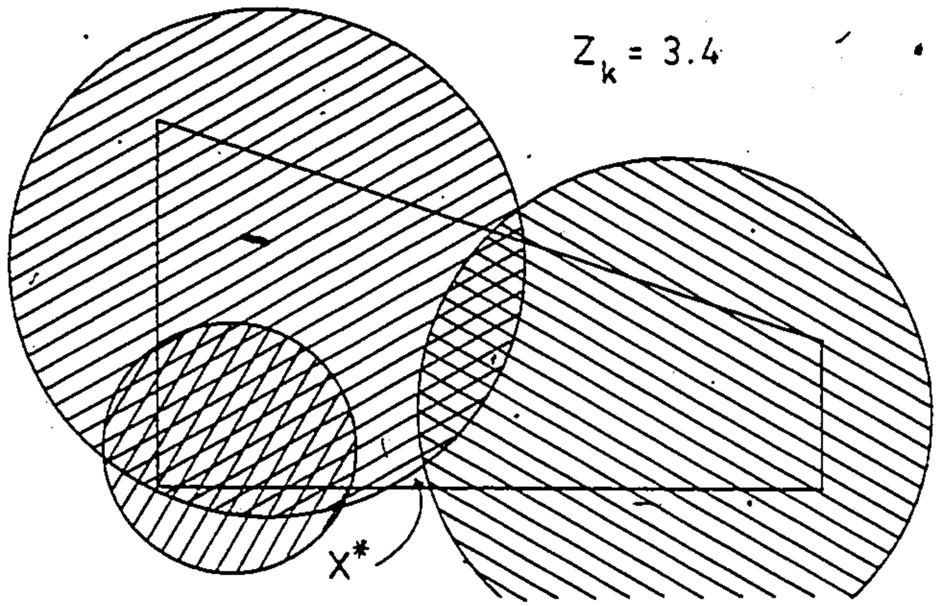

Figure 1.7. A simple example - Final stage.

is usually not convex. When there exist barriers to travel, the distance function itself is not convex since it is not defined over a convex region.

Specialized numerical algorithms usually avoid the problems associated with standard search techniques by adopting a combinatorial approach which repeatedly finds analytic solutions to the problem for small subsets of the fixed points, usually pairs on the line or triples in the plane. Dasarathy and White (1980) is one example of this type of approach. The practicality of this approach depends on the distance functions having relatively simple properties which is the case for the Euclidean, rectilinear, and general l_p distances from a point.

The graphical approach can be used to solve problems with costs which are nonlinear functions of distance (Hansen et al. 1981), with mixed distance norms, and with non-convex feasible regions (Brady and Rosenthal 1980). It is the ability to attack problems with non-convex, possibly disjoint, feasible regions that is a major advantage of this graphical procedure. The visual representation of the problem helps to build client confidence since both the validity of a location model and the appropriateness of its solution are more readily demonstrated (Brady and Rosenthal 1980). Also, the possibility of user modification of constraints and objectives can lead to greater realism. Only trivial modifications are required to generate the contour lines of the objective function which are useful when the decision maker wants a good but not necessarily optimal location (Francis 1967; Brady and Rosenthal 1980).

An inherent advantage of the graphical method over the specialized numerical methods is that it deals with the demand points P_i one at a time. Because relationships between pairs or triples of demand points are not used in the solution procedure, it is relatively easy to apply this method to problems which have different distance metrics associated with different demand points or which possess complicated cost or distance functions.

1.5 Scope and Order of Presentation

Chapter 2 surveys the literature and discusses the state of the art with respect to the location problems considered in Chapters 3 and 4. Chapter 3 examines constrained single facility weighted minimax and maximin location problems with rectilinear distances and right-angled barriers to travel that are aligned with the travel directions. The demands are considered to be points and the feasible region for facility location may be non-convex or disjoint. A method of efficiently determining the level sets of the distance function in the presence of barriers to travel is developed and used to apply the graphical method to these location problems. A purely numerical algorithm for unweighted maximin problems is also described.

Chapter 4 examines a constrained weighted maximin single facility location problem with area demands which are allowed to be general, possibly non-convex, polygons instead of single points. This is presented in the context of optimally placing a single undesirable or noxious facility with respect to several vulnerable areas. Euclidean, rectilinear, and general l_p distances are treated, but barriers to travel are not considered. Extension of the method to asymmetric functions of distance and costs which are non-linear functions of distance is discussed.

Chapter 5 presents a summary of this research and recommendations for further investigations.

CHAPTER 2

STATE OF THE ART

This chapter reviews some of the literature related to the two types of problems considered, namely, maximin location problems with area demands and minimax or maximin location problems with rectilinear distances in the presence of barriers to travel.

The very extensive literature on location problems in general will not be surveyed here. Location problems in continuous space are covered in the forthcoming book by Love, Morris, and Wesolowsky as well as in the text by Francis and White (1974). Selective literature reviews can be found in Francis, McGinnis, and White (1983) and Hansen, Peeters and Thisse (1983). Domschke and Drexl (1985) have compiled an extensive international bibliography on location and layout-planning problems. One may refer to the textbook by Handler and Mirchandani (1979) for a comprehensive discussion and bibliography of location problems on networks. A discussion of discrete plant location problems can be found in the survey article of Krarup and Pruzan (1983).

2.1 Minimax and Maximin with Barriers to Travel

In this section, we review some early work on minimax location problems and then examine more recent work which is directly related to the minimax and maximin rectilinear distance location problems with barriers which are the subject of Chapter 3.

2.1.1 Minimax and Maximin

The history of minimax location problems in the plane dates back to 1857 when J. J. Sylvester posed the problem of finding the circle of smallest radius that enclosed a number of points. This minimum covering circle problem is a special case of the minimax location problem with Euclidean distances and with all weights equal. In 1860, Sylvester gave a geometrical solution method attributed to Peirce (Love, Morris, and Wesolowsky). Surprisingly, improved algorithms for this century-old problem are still being discovered (Oommen 1987).

Modern interest in minimax location problems appears to date from the mid-1960s. Smallwood (1965) used a hill climbing iterative method to find locally optimal solutions to a multifacility Euclidean distance minimax problem involving the placement of identical detection stations for optimum coverage of an arbitrary plane area. Interestingly, Smallwood appears to have used a computer graphics terminal to display and study the trajectories followed by the stations during the solution process, although computer graphics plays no role in the algorithm.

Elzinga and Hearn (1972a) presented efficient, finite solution methods based on geometrical arguments for unweighted single facility problems with rectilinear or Euclidean distances which they called "Delivery Boy" problems. Unlike Peirce's method, their algorithms could be conveniently implemented on an electronic computer. For the Euclidean distance problem, they constructed successively larger circles defined by two or three demand points until a minimum covering circle was obtained. For the rectilinear distance problem, a simple closed form solution was found. They also presented algorithms to solve related "Messenger Boy" problems in which positive constants k_i are added to the distances. Francis (1972) presented a similar result for rectilinear distance problems which could be applied to any bounded set in the plane.

Weighted minimax problems proved to be more difficult than unweighted (all w_i equal) problems and solution methods developed more slowly. Francis (1967) presented a geometric characterization of the solution of the weighted single facility minimax problem. For the Euclidean distance case, the solution is obtained by varying z to achieve the smallest non-empty intersection of discs with radius z/w_i centered at the demand points P_i. Francis pointed out that for Euclidean or rectilinear distances this graphical construction could be used to find the optimum location or to construct the level curves of the objective function. Francis presented other results for the unconstrained weighted single facility minimax problem which included a lower bound and a necessary condition for the optimum as well as an explicit solution when the lower bound was obtained.

The simple, almost linear, form of the rectilinear distance function leads to effective linear programming approaches to the minimax problem. Wesolowsky (1972) introduced a linear programming formulation of the weighted multifacility linearly constrained rectilinear distance problem which could be solved using parametric programming. Later, Elzinga and Hearn (1973) and Morris (1973) presented simpler linear programming formulations which could be solved without parametric programming.

Dearing and Francis (1974) presented a linear programming based method for the weighted multifacility problem with rectilinear distances and maximum distance constraints. The minimax rectilinear distance problem is not separable into x and y axis subproblems in the simple manner of the minisum rectilinear distance problem because the selection of the maximum distance demand point links the x and y coordinates together. However, Dearing and Francis were able to decompose it into two subproblems which could be solved independently by network flow techniques.

For weighted rectilinear distance multifacility problems which are unconstrained or which have linear constraints, formulation as a linear program has proved to be an effective solution method. For some constrained problems having convex feasible regions which cannot be represented by a set of linear inequalities, other mathematical programming methods are available such as the subgradient algorithm of Chatelon, Hearn, and Lowe (1982) for single facility minimax problems with l_p distances.

Drezner and Wesolowsky (1980a) presented a fast method for the unconstrained weighted l_p distance single facility minimax problem based on the fact that in a convex minimax problem in k dimensions there exists a subset of k+1 functions which defines the optimum solution (Drezner 1982). The method proceeds by finding the solution for a set of three demand points and then attempting to find a demand point which has a greater weighted distance from the current location. If such a point is found, then a new set of three demand points which yields a higher objective function value can be formed. The procedure is quite similar in spirit to the Elzinga and Hearn (1972a) method for unweighted Euclidean distance problems. A specialized version of the method which decomposes the problem into two independent one dimensional subproblems was used for the rectilinear distance case and proved to be very fast by solving a 5000 point problem in less than half a second. For unconstrained single facility problems with rectilinear distances, this method is very effective.

Brady and Rosenthal (1980) introduced interactive computer graphical optimization to location analysis and pointed out that the advantages of the graphical approach included the ability to deal with non-convex or disjoint feasible regions. They used a bisection search over objective function values with an interactive step in which the operator was required to decide if the intersection of a set of circles contained any feasible points. This interactive graphical method was used to solve single facility minimax problems with Euclidean or rectilinear distances and general constraints. Brady, Rosenthal, and Young (1983) extended the algorithm to location-allocation type multifacility Euclidean distance problems. This extension depended on the operator's ability to perceive whether or not a given pattern of circle intersections yielded m-facility coverage of the demand points.

Plastria (1987) presented a cutting plane method which could be be used to solve a large class of single facility location problems including minimax problems with mixed norms and nonlinear cost functions. Although the method is limited to unconstrained problems or problems having convex polyhedral feasible regions, it is much more flexible than most numerical methods.

Hansen, Peeters, Richard, and Thisse (1985) presented an algorithm called FOCUS AND EXPAND which could be used to solve single facility minimax location problems with a mixture of l_p norms and nonlinear cost functions. The feasible region was allowed to be the union of a set of convex polygons. In the first stage of their two stage algorithm, the unconstrained optimum was found using a procedure which is in essence a numerical version of Brady and Rosenthal's graphical method. The second stage of the algorithm checked if the unconstrained optimum was a feasible solution to the constrained problem. If it was not feasible, the boundaries of the convex polygonal constraints were searched using their visibility concept and dominance rules.

Methods for maximizing the minimum weighted rectilinear distance to a number of demand points have received considerably less attention than methods for the minimax problem. Drezner and Wesolowsky (1983b) studied the single facility weighted maximin location problem with rectilinear distances and linear constraints. They presented two different solution methods. In the first, they searched the boundary of the feasible region and certain line segments in the interior. In the second, they broke the problem down into a number of linear programming problems, one of which must yield the solution. Mehrez, Sinuany-Stern, and Stulman (1986) provided an improved version of Drezner and Wesolowsky's linear programming based method but the improved version only applied to the special case of unweighted distances. The graphical method of Hansen, Peeters, and Thisse (1981) for locating an obnoxious facility can be applied to single facility rectilinear distance problems, but we will postpone describing their method until the section on noxious facility location.

2.1.2 Barriers

Although facility location in the presence of barriers to travel has many practical applications, very little work has been done on the problem. When one considers that a general barrier may, quite literally, be a maze, it is not surprising that only the rectilinear distance minisum location problem with barriers but without location constraints has been adequately treated.

Determination of the shortest path is an integral part of the facility location problem with barriers to travel and has been considered by a number of researchers in relation to a variety of problems. For Euclidean distances, Wangdahl, Pollock, and Woodward (1974) gave a dynamic programming method for minimum length pipe routing in ship design. Lozano-Perez and Wesley (1979) gave a similar algorithm in the context of navigating a robot vehicle among convex polygonal obstacles. Shortest Euclidean distance paths in the presence of disjoint linear barriers were studied by Chein and Steinberg (1983). Lee and Preparata (1984) considered two special cases of the shortest Euclidean distance path problem with polygonal or linear barriers. Algorithms which have been developed for the related problem of routing wiring traces on circuit boards were discussed by Belter (1987).

Larson and Li (1981) developed an algorithm to find the shortest rectilinear distance path in the presence of polygonal barriers to travel. Using results from this work, Larson and Sadiq (1983) proved that a location-allocation type of multifacility minisum problem with rectilinear distances and general barriers to travel could be reduced by inspection to a finite network problem. The network problem could then be solved using any available p-median algorithm.

Viegas and Hansen (1985) studied the problem of finding the shortest l_p distance path in the presence of polygonal barriers and gave an algorithm for its solution.

Aside from Larson and Sadiq (1983), the only other work on location problems with barriers to travel in continuous space is that of Katz and Cooper (1979a, 1979b, and 1981), Muralimohan and Babu (1983a, 1983b), and Ravindranath, Vrat, and Singh (1985).

Katz and Cooper (1981) discussed a general minisum single facility problem with l_p distances and a single region where travel is forbidden which they called a "forbidden region" rather than a "barrier to travel." After formulating the problem with Euclidean distances and one circular barrier to travel, they used a standard nonlinear programming method to solve two numerical examples and found several local minima in each case. Katz and Cooper (1979b) extended this to Euclidean distance problems with several circular barriers.

Muralimohan and Babu (1983a) considered the Weber problem with one convex barrier to travel. Muralimohan and Babu (1983b) extended this to two convex non-intersecting barriers and applied it to the case of two ellipses. Ravindranath, Vrat, and Singh (1985) formulated a single facility location problem with rectilinear distances and one rectangular or circular barrier to travel. They used a standard mathematical programming search technique to minimize a weighted combination of minisum and minimax objective functions.

2.1.3 Summary of Minimax and Maximin with Barriers

Although both minimax and maximin single facility location problems with rectilinear distances and right-angled barriers to travel have many practical applications, no solution methods have yet been presented for either problem.

Excluding graphical approaches, the methods used to solve other minimax and maximin problems with rectilinear distances are not easily adapted to problems with barriers. Simple extensions of the linear programming formulations do not appear to be possible because the distance between a demand point and a potential facility location is no longer a simple function of the two sets of coordinates. The method of Drezner and Wesolowsky (1980a) and convex programming methods are ruled out because the distance function is no longer convex when there are barriers to travel. The minisum rectilinear distance problem with barriers could be solved by Larson and Sadiq because, like many other median-type problems, an optimum solution could be found in a finite set of easily identified candidate points. For example, in a median problem on a network an optimum solution can always be found at one of the nodes. Neither the minimax, a center-type problem, nor the maximin rectilinear distance problem have this convenient property.

2.2 Noxious Facilities with Area Demands

In this section, we review some of the previous work related to the topic of Chapter 4, locating a noxious facility with the objective of maximizing the minimum weighted distance to any of a number of polygonal areas. We will first examine work on noxious facilities with point demands and then work which has been done on facility location with area demands.

2.2.1 Noxious Facilities

Locating a facility as far as possible from a set of demands is a problem which has received much less attention than those problems in which the objective is to place the facility as close as possible to the demands. Problems in which the objective is to locate the facility as far as possible from the demands are usually called "undesirable," "obnoxious," or "noxious" facility location problems. Although locating a noxious facility is the more common interpretation, it is also possible to interpret these types of problems as locating a vulnerable facility as far as possible from a set of hazardous points. For example, this is the interpretation used by Mehrez, Sinuany-Stern, and Stulman (1983) when locating a facility along a line with the objective of maximizing the minimum distance to the closest of a set of points on the line.

Some examples of early work on noxious facility location are Paroush and Tapiero (1976) which examined the problem of locating a polluting plant on a line under uncertainty, as well as Church and Garfinkel (1978) which examined the problem of locating a single facility on a network with the objective of maximizing the minimum weighted distance from the facility to any node.

Maximin location problems in continuous space have inconvenient properties which tend to make them more difficult than minimax problems. Constraints which define a bounded feasible region for facility placement are required for the problem to be non-trivial. Otherwise, the optimum and unrealistic solution is to place the facility at infinity. Also, the minimum of a set of convex distance functions is not convex unlike the maximum of a set of convex functions which is convex. The resulting maximin objective function has many local minima and maxima, precluding many mathematical programming techniques. Most mathematical programming approaches to maximin problems attempt to identify and check certain interior points at which the optimum may occur as well as to search the boundary of the feasible region.

Shamos and Hoey (1975) solved an unweighted Euclidean distance maximin problem which they called "the largest empty circle problem" with a technique employing Voronoi diagrams. The location was restricted to lie inside the convex hull of the set of points. They showed that the center of the required circle must lie either at one of the Voronoi points or at the intersection of a Voronoi edge and the convex hull.

Dasarathy and White (1980) examined the unweighted single facility maximin problem with Euclidean distances and point demands. For the two dimensional case, they extended the algorithm of Shamos and Hoey to treat problems in which the optimum location is required to lie inside a convex polygon. They also presented a combinatorial algorithm for the three dimensional case which checked certain interior points as well as the faces and edges of the convex polyhedron in which the optimum location was required to lie. Melachrinoudis (1980) presented a combinatorial method for weighted Euclidean distances in the two dimensional case.

Drezner and Wesolowsky (1980b) used a different type of mathematical programming approach than the preceding methods to solve a maximin location problem with Euclidean distances and maximum distance constraints. Their method used a bisection search over objective function values like the method of Brady and Rosenthal (1980). The existence of feasible solutions at a particular objective function value was checked by a numerical method rather than by interactive computer graphics. Their numerical method searched for a feasible solution on the circumference of at least one of a set of circles defined by the current objective function value.

Drezner and Wesolowsky (1983) presented an algorithm for the minimax and the maximin single facility location problem on the sphere with great circle distances. Drezner (1983) provided a unified treatment of problems in which both the objective function and the constraints involved monotonic functions of Euclidean distance in the plane or great circle distance on the sphere. This formulation allowed the facility to be excluded from the interior of some circles and to be required to lie inside others.

Melachrinoudis and Cullinane (1985a) extended Melachrinoudis' combinatorial algorithm for weighted

Euclidean distance maximin problems to cases with non-convex polygonal feasible regions. They also included minimum distance constraints in their formulation producing circular infeasible regions centered on the demand points and called these "forbidden regions." (These are not barriers to travel and should not be confused with the "forbidden regions" of Katz and Cooper, which are.) Their solution method finds all local maxima to a relaxed problem without the "forbidden regions" by checking interior points and searching the boundary. If the best of these local maxima violates one or more of the "forbidden regions," the violated constraints are added to the problem and new local maxima are found by considering constraints three at a time. They applied their method to a problem of locating a waste dump in the state of Massachusetts. The circular "forbidden regions" about the demands can be regarded as an indirect and partial solution to the problem that the cities in their application are extended geographical areas rather than point demands.

Melachrinoudis and Cullinane (1985b) presented a heuristic version of Melachrinoudis' combinatorial algorithm for maximin location problems with convex polygonal feasible regions. The heuristic version reduced the amount of computation required for large problems by checking only close triples of demand points.

Weighted maximin problems with rectilinear distances have been treated by Drezner and Wesolowsky (1983b) who considered a problem with constraints in the form of linear inequalities and presented two solution methods. The first method searched the boundary of the feasible region and certain interior line segments. The second method broke the problem down into a number of linear programming problems. Mehrez, Sinuany-Stern, and Stulman (1986) showed that the linear programming approach used by Drezner and Wesolowsky for the weighted case could be significantly improved when applied to unweighted problems.

Drezner and Wesolowsky (1985) established duality relationships between a multifacility maximin location problem with maximum service radius constraints and a related minimax problem. Abhinorasaeth and Melachrinoudis (1985) as well as Benhamou and Melachrinoudis (1987) considered single facility location problems with a combination of minisum and maximin criteria.

Hansen, Peeters, and Thisse (1981) presented a graphical algorithm (BLACK AND WHITE) for locating a single obnoxious facility in relation to a set of point demands. Their method consisted of first finding an unshaded feasible point and then shading all those parts of the feasible region which are dominated by the feasible point. (In a maximin problem, a point is dominated if the objective function of the point is less than that of the last feasible point evaluated.) This is repeated until the remaining unshaded feasible region, which must contain the optimum, is sufficiently small. They pointed out that this is a very flexible method since it can easily handle problems with mixed distance norms, cost functions which are non-isotropic or which are non-linear functions of distance, and any feasible region which can be drawn, including non-convex and disjoint regions.

Melachrinoudis and Cullinane (1986) considered the problem of locating an undesirable facility so that the maximum weighted inverse of the squared Euclidean distance to the fixed points was minimized. The inverse square of the Euclidean distance was chosen because the levels of emissions from point sources often follow a $1/r^2$ law where r is the distance from the source. A slightly modified version of Melachrinoudis' combinatorial algorithm for maximin problems was presented for problems which have a convex feasible region. For problems with non-convex feasible regions, a simple interactive computer graphic method was presented. The method drew the level curves of the $1/r^2$ functions about each fixed point for a series of decreasing, closely spaced, objective function values. This produced a set of expanding circles centered at the fixed points. The last feasible point left outside the circles by this procedure was the optimum location. The authors noted that when the remaining feasible space has been reduced to a very small area, the graphics display can be "zoomed" in on the feasible area to gain additional accuracy.

In summary, all previous work on noxious facility location problems has involved locating the facility in relation to a finite set of fixed points. The solution algorithms presented for these problems fall into two categories: mathematical programming methods, usually combinatorial in nature, and graphical or interactive computer graphic methods.

2.2.2 Area Demands

Almost all location models treat demands as a finite set of points, although practical applications may involve locating a facility in relation to a population which is distributed over one or more regions. The limited amount of work on facility location in relation to area demands has been concentrated, almost exclusively, on minisum location problems.

Minisum location problems with area demands and Euclidean distances were treated in Love (1972), Bennett and Mirakhor (1974), Drezner and Wesolowsky (1978), and in Cavalier and Sherali (1986). Drezner (1986) examined a related Euclidean distance minisum problem in which both the facility and the demands are circular regions. Minisum location problems with area demands and rectilinear distances were treated in Wesolowsky and Love (1971), Marucheck and Aly (1981), and in the forthcoming book by

Love, Morris, and Wesolowsky. Minisum location problems with area demands and general l_p distances were treated in Drezner and Wesolowsky (1980c). Because the algorithms presented in these works use a variety of methods which are not applicable to the maximin location problems, we will not discuss them any further.

In their book, Love, Morris, and Wesolowsky point out that the extreme points of the area demands determine the minimax solution. Thus, a minimax problem with area demands in the form of polygons can be reduced to a minimax problem involving only the corners of the polygons, a finite number of points, and solved by algorithms designed for point demand problems. This, undoubtedly, accounts for the lack of interest shown by operations researchers in minimax problems with area demands. The sole exception is Francis (1972) which gives a geometrical solution procedure for the unweighted minimax location problem with rectilinear distances and area demands which were allowed to be any bounded set in the plane.

2.2.3 Summary of Noxious Facilities with Area Demands

Although the maximin or noxious facility location problem with area demands has practical applications, no solution methods have yet been presented. The maximin problem with area demands does not reduce to a problem involving a finite number of points because any point on the boundary of an area demand may play a role in determining the optimum solution to a maximin problem. The mathematical programming methods are also difficult to adapt to area demand problems because they depend on the distance function having convenient properties such as convexity. For the case of area demands, the level curves are not simple and the distance function is not necessarily convex. Thus, the available methods for maximin or noxious facility location cannot be directly applied to problems with area demands or, with the exception of the graphical method, modified to handle them without considerable difficulty.

CHAPTER 3

RECTILINEAR DISTANCE LOCATION PROBLEMS WITH BARRIERS

3.1 Introduction

In this chapter we present an interactive computer graphic method for single facility minimax and maximin location problems where the underlying distance metric is rectilinear and there exist barriers to travel. A purely numerical algorithm for unweighted maximin location problems is also discussed.

3.1.1 Description of Problem

We examine the problem of placing a single facility in the plane with the objective of either minimizing the maximum distance from the facility to a set of demand points or maximizing the minimum distance.

For this chapter, we assume that the underlying distances can be approximated by rectilinear distances as would be the case when travel is restricted to two orthogonal directions by a rectangular street grid or by aisles in a factory or warehouse. Thus, the travel distance between two points, P_1 and P_2, can be calculated using the rectilinear metric, $d(P_1,P_2) = |x_1-x_2|+|y_1-y_2|$, unless the distance is increased by the presence of barriers.

A barrier to travel is a region of the plane through which travel is not possible or not permitted. We assume that one or more barriers to travel are present and that the barriers to travel are right-angled, possibly non-convex, polygons with their edges aligned with the travel directions (Figure 3.1). We assume without loss of generality that the polygons do not overlap.

Since the boundaries of parks and zones where certain types of traffic are prohibited tend to follow the street grid, such right-angled polygons are a common type of barrier. Barriers to travel with other shapes, such as lakes, can be approximated by enclosing them in right-angled polygons.

In addition, we assume that the facility is constrained to lie inside a given feasible region. The feasible region may be non-convex or disjoint. In minimax problems, the feasible region may be the entire plane. For convenience, we will assume that the feasible region consists of one or more general polygons, although our method can accommodate feasible regions of any shape which can be drawn and filled on the screen. The complement of the feasible region is the infeasible region for facility placement. Travel through an infeasible region is permitted, although the facility cannot be located there.

Barriers to travel have been called "forbidden regions" by some authors. To avoid confusion with

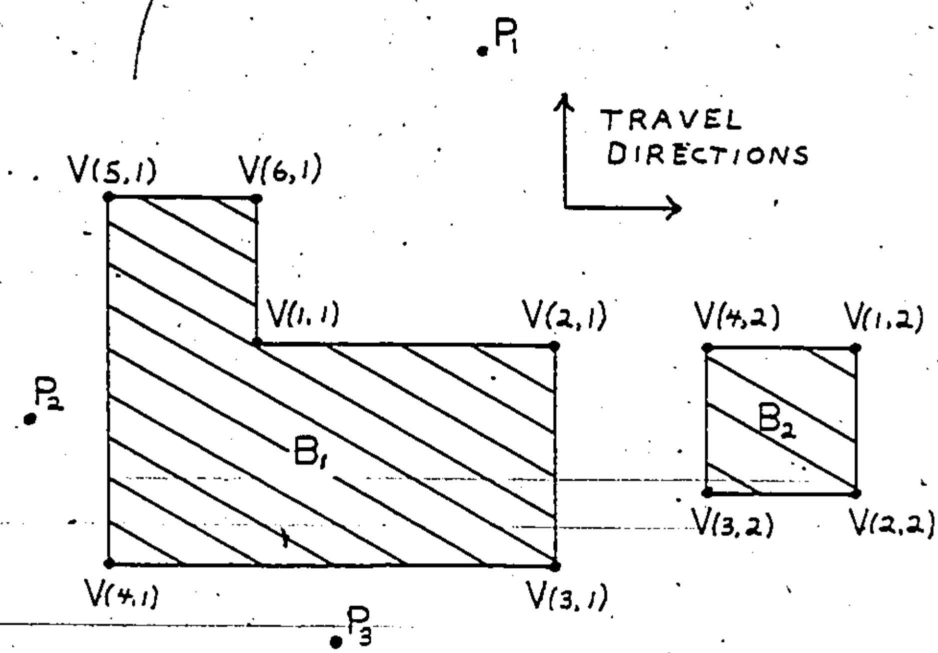

Figure 3.1 Demand points and right-angled barriers.

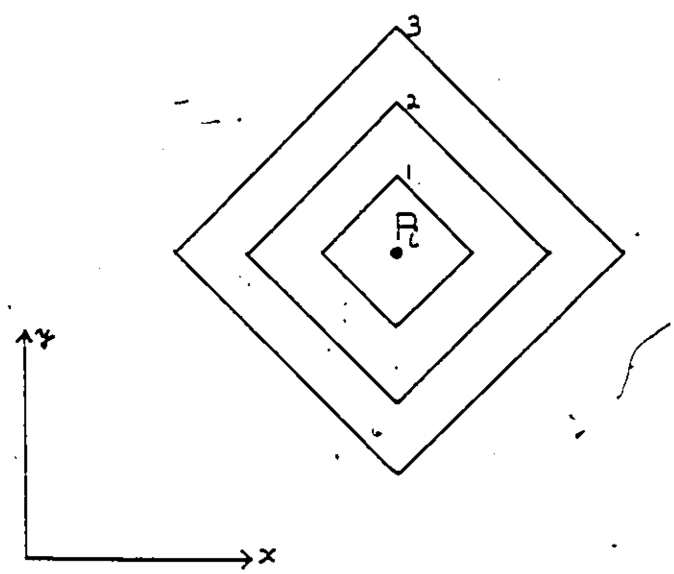

Figure 3.2 Rectilinear distance diamonds.

"infeasible regions" as defined earlier, we will use the more descriptive "barriers to travel" or simply "barriers" and avoid the term "forbidden regions."

In a practical sense, a barrier to travel is also an infeasible region for facility placement, since a facility lying in the interior of a barrier to travel cannot be reached. However, we will reserve the term "infeasible region" for those regions where travel is permitted but facility placement is not.

3.1.2 Formulation

Let B_m, $m = 1,\ldots,N_B$ be a set of right-angled barriers to travel with vertices $V(k,m) = (x(k,m),y(k,m))$, $k = 1,\ldots,v_m$ and $m = 1,\ldots,N_B$ (Figure 3.1). Let F be the feasible region for facility placement.

The minimax single facility location problem with rectilinear distances and barriers to travel can be formulated as

$$\underset{X \text{ in } F}{\text{minimize}} \ \{ \underset{i=1,\ldots,N_P}{\text{maximum}} \ f_i(X) \} \tag{3.1.1}$$

where

$f_i(X) = w_i d(X,P(i))$,

$X = (x,y)$ = location of facility,

$P(i) = (x(i),y(i))$ = location of i'th demand,

w_i = positive weight associated with i'th demand,

N_P = number of demands,

and

$d(X,P(i))$ = minimum rectilinear distance from X to P(i) in presence of barriers B_m, $m = 1,\ldots,N_B$.

The related maximin problem can be obtained by changing the objective function to

$$\underset{X \text{ in } F}{\text{maximize}} \; \{ \underset{i=1,\ldots,N_P}{\text{minimum}} \; f_i(X) \}. \qquad (3.1.2)$$

A bounded feasible region for facility placement is necessary in order for the maximin problem to be non-trivial.

3.1.3 Requirements for Graphical Approach

The graphical method alternates between filling the area between level curves of a distance function and calculating the distance from a demand point to a general feasible point. In the absence of barriers to travel, both of these operations are extremely simple. The travel distance between any two points is given by the rectilinear distance metric and the level curves are simply squares (usually called "diamonds") as shown in Figure 3.2.

Since the right-angled polygonal barriers to travel may literally be a maze, the level curves of the distance function $d(X,P(i))$ can have quite complex shapes and the distances can be correspondingly difficult to calculate. Figure 3.3 shows the level curves of a distance function $d(X,P(i))$ in the presence of a simple set of barriers.

We require a method which will allow us to determine those areas which lie between particular level curves of the

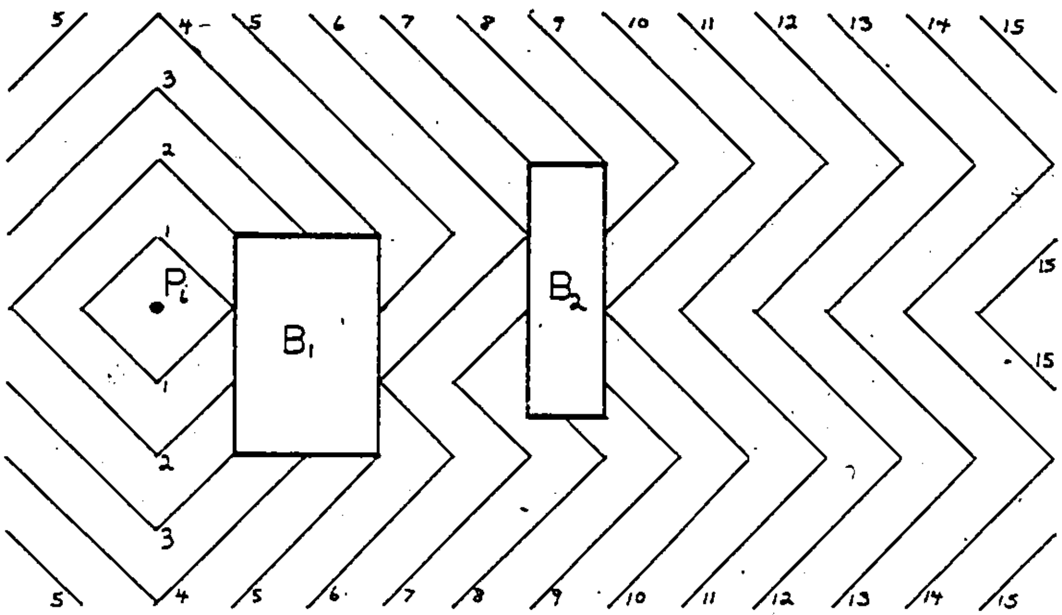

Figure 3.3 Level curves with simple barriers.

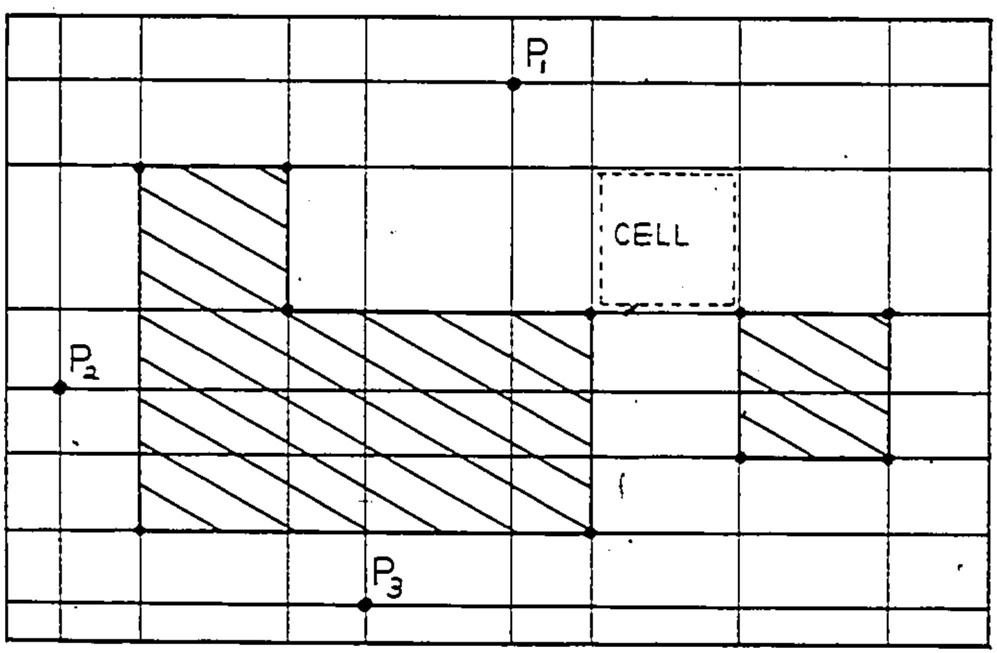

Figure 3.4 Division of problem universe into cells.

distance function when we are given any set of right-angled barriers to travel. We will accomplish this by dividing the plane into rectangular areas, "cells," in which the level curves of the distance function are reasonably simple to calculate if we know the values of the distance function at each of the corners of the cell. As a by-product of our method of determining the level curves, we will also obtain a method of calculating the distance from a demand point to a general feasible point.

Section 3.2 explains the division of the plane into cells and proves that only the distances to the corners of a cell are necessary to calculate the distance to any point inside the cell. Using this result, section 3.3 presents the distance function and the level curves of the distance function inside a cell. Section 3.4 explains how the theory and results of Larson and Li (1981) can be used to calculate the distances from the demand points to the corners of the cells. Section 3.5 presents the algorithm for maximin location with a description of a program which implements it and an example. Section 3.6 presents the algorithm for minimax location problems and discusses the differences from maximin location problems. Section 3.7 discusses a purely numerical version of the algorithm which can be applied to unweighted maximin problems.

3.2 Division of Plane into Cells and Properties

The maximin and minimax location problems with barriers to travel were formulated in the entire xy-plane. However, working with such an unbounded region would unnecessarily complicate our discussion. We can avoid this complication by introducing a large finite rectangular region which we will call the "problem universe" containing all the elements of the location problem. This will not alter the optimal solution of any problem and is similar to the common practice of choosing some large but finite number to represent infinity when an algorithm is implemented as a computer program.

We note that such a rectangular region can always be found for a given problem. For minimax problems with a bounded feasible region or any non-trivial maximin problem which necessarily has a bounded feasible region, we can use as the problem universe any rectangle which completely encloses the demand points, the barriers, and the feasible region. For minimax problems with a feasible region which consists of the entire plane, any rectangle completely enclosing the demand points and the barriers can be used as the problem universe.

Let wxmin and wxmax be the x-coordinates of the left and right edges of the rectangular problem universe. Let wymin and wymax be the y-coordinates of the bottom and top edges of the problem universe.

Definition 3.1 Let the problem universe be partitioned into rectangular cells (see Figure 3.4) by vertical lines located at each distinct x-value in the set {wxmin, wxmax; x(i), i=1,...,N_P; x(k,m), k=1,...,v_m and m=1,...,N_B} and horizontal lines at each distinct y-value in the set {wymin,wymax; y(i), i=1,...,N_P; y(k,m), k=1,...,v_m and m=1,...,N_B}.

In effect, the rectangular problem universe is cut into small rectangles, cells, by the vertical and horizontal lines.

3.2.1 Cell Properties

Lemma 3.1:

For any given cell, any demand point P(i) or barrier vertex V(k,m) must lie either at a corner of the cell or outside the cell in one of the "corner regions" (Figure 3.5).

Proof:

Let xmin and xmax be the x-coordinate of the left and right sides of the cell, respectively. Let ymin and ymax be the y-coordinates of bottom and top of the cell, respectively.

Assume that the P(i) does not lie at a corner of the cell or outside the cell in one of the "corner regions." Therefore, either the x-coordinate of P(i) lies strictly between xmin and xmax, or the y-coordinate lies strictly between ymin and ymax, or both. However, all distinct values were used to construct the cells, giving rise to a

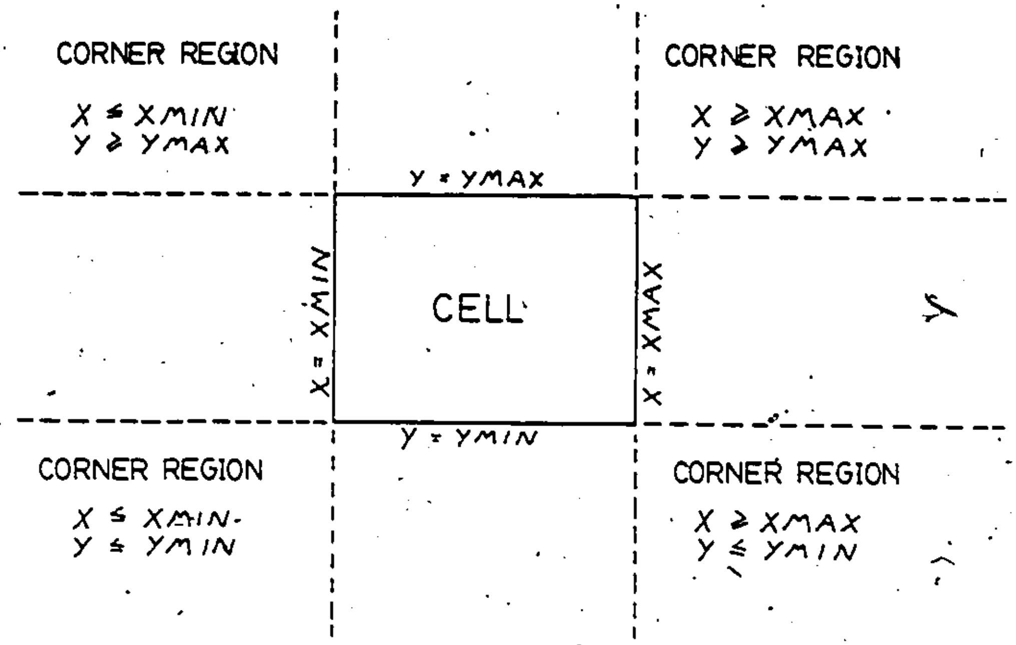

Figure 3.5 Cell and corner regions.

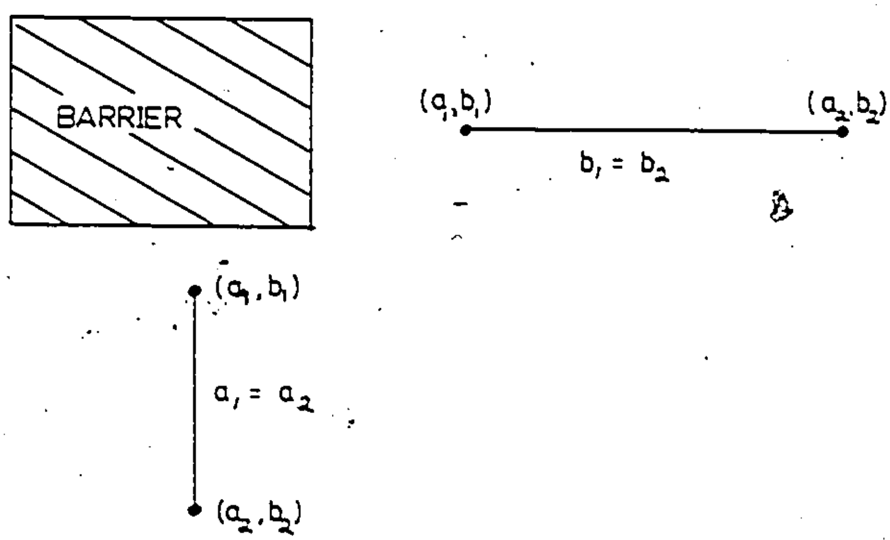

Figure 3.6 Simply communicating vertices - Condition 1.

contradiction. The proof for a barrier vertex V(k,m) is identical. □

Lemma 3.2:

No boundary of a barrier lies in the interior of a cell. Thus, the interior of a cell is either entirely inside a barrier or entirely outside all barriers.

Proof:

Since the barriers are right-angled polygons aligned with the x and y axes any barrier side must be a horizontal or vertical line segment with a barrier vertex at each end. If this side lies in the interior of the cell then at least one of the coordinates of the barrier vertices at its end points is distinct from those used to determine the boundaries of the cells. This is a contradiction. □

3.2.2 Four Entry Point Theorem

Before proving an important property of cells, it is necessary to review some results obtained by Larson and Li (1981). They studied the problem of finding minimum rectilinear distance paths in the presence of several polygonal barriers to travel and presented an algorithm, POLYPATH, which determined such paths. The barriers that they considered were general polygons and were not restricted to right-angled polygons aligned with the travel directions as are the barriers in this work.

In Larson and Li's network terminology, both the origin and destination of a path as well as the vertices of the polygonal barriers to travel are termed "vertices."

Larson and Li defined vertices which were connected by particularly simple feasible paths as being "simply communicating vertices." For the special case of right-angled barriers aligned with the travel directions their definition can be simplified and restated in an equivalent form.

Definition 3.2: Two vertices with coordinates (a_1, b_1) and (a_2, b_2) are simply communicating vertices under either of the following conditions:

(1) If either $a_1 = a_2$ or $b_1 = b_2$ and the line segment joining the two vertices has no point in common with the interior of any barrier and no other vertices lie on it (Figure 3.6). This includes the case of adjacent vertices of a barrier.

(2) If a horizontal ray drawn from one vertex intersects a vertical ray drawn from the other vertex before either ray encounters the interior of a barrier or another vertex. Thus, the rays must intersect either at (a_1, b_2) or (a_2, b_1) and neither of the line segments from the vertices to the point of intersection can have any point in common with the interior of a barrier or have any other vertex lying on it (Figure 3.7).

It is obvious from the preceding definition that the travel distance between a pair of simply communicating

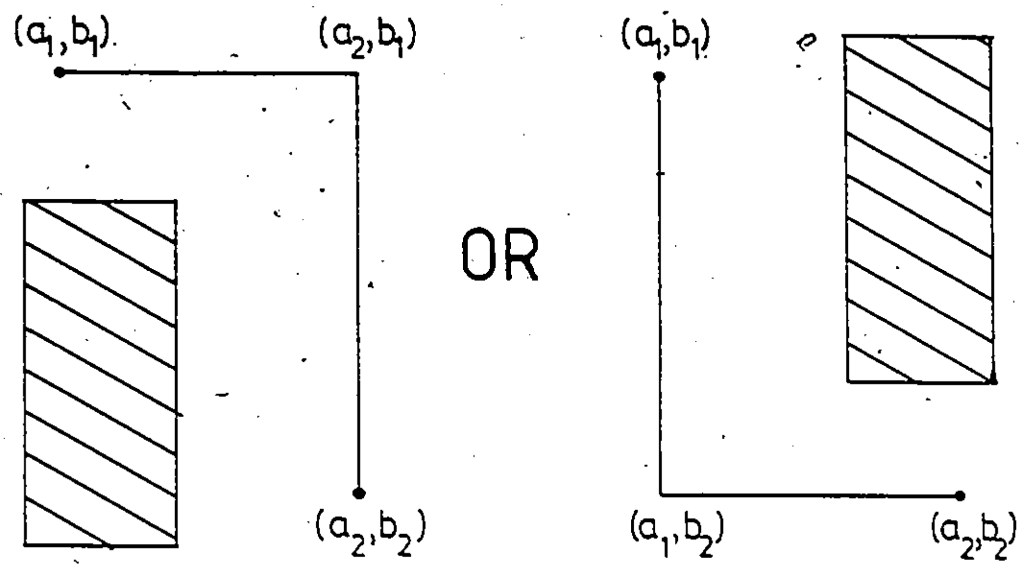

Figure 3.7 Simply communicating vertices - Condition 2.

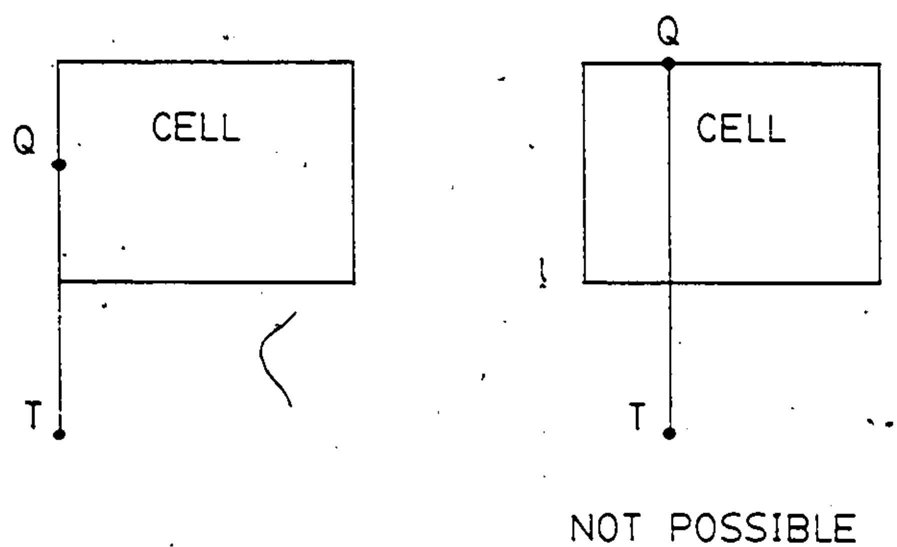

Figure 3.8 T and Q satisfy condition 1 and Q on a side.

vertices with coordinates (a_1,b_1) and (a_2,b_2) is equal to the rectilinear distance between them, $|a_1 - a_2| + |b_1 - b_2|$, since a feasible path exists which has not been increased in length by the presence of the barriers.

We also note that the minimum rectilinear distance path between any two points is usually not unique. In particular, there exist an infinite number of minimum rectilinear distance feasible paths between between any pair of simply communicating vertices which satisfy condition (2) of definition 3.2. Since the optimum facility location depends only on travel distances, we will not attempt to specify a unique path from among the possibly infinite number of minimum distance feasible paths between any pair of simply communicating vertices.

We can now state the very useful result of Larson and Li. In Larson and Li's formulation of the shortest path problem, let U be the set of vertices u(i) for i = 1,...,m where m is the total number of vertices. If there exists a feasible path from vertex u(s) to vertex u(d) both in U, then there exists a minimum distance feasible path from u(s) to u(d) which passes through a sequence of vertices {u(s), $u(k_1)$, $u(k_2)$, $u(k_3)$, ... , $u(k_n)$, u(d)} where the $u(k_j)$ for j = 1,...,n are in U and the pairs {u(s), $u(k_1)$}, {$u(k_1)$, $u(k_2)$}, {$u(k_2)$, $u(k_3)$}, ... , {$u(k_n)$, u(d)} are simply communicating vertices. This includes the possibility that n = 0, in which case the sequence of vertices contains only the pair of vertices (u(s), u(d)) and they are simply communicating vertices.

Larson and Li's result reduces the shortest rectilinear distance path problem in the continuous plane to a finite network problem since only paths passing through a sequence of simply communicating vertices need to considered in the search for a shortest path. Although the minimum distance path taken between any pair of simply communicating vertices in the sequence may not be unique, this need not concern us because we are only interested in the length of the minimum distance path between the pair of vertices and this length is unique and well defined. The total length of any minimum rectilinear distance feasible path which follows a sequence of simply communicating vertices is easily calculated since it is the sum of the distances between the pairs of simply communicating vertices through which the path passes.

<u>Theorem 3.1</u> (Four Entry Points):

Let Q be a point in the interior or on the boundary of a given cell. If a minimum distance feasible path exists between a demand point P(i) and Q, then a path of equal length from P(i) to Q can be found which passes through one of the corners of the cell.

<u>Proof</u>:

The demand point P(i) must lie outside the cell containing Q or at a corner of the cell by Lemma 3.1.

If P(i) or Q lies at a corner of the cell, then all paths from P(i) to Q pass through a corner of the cell and the theorem holds.

If P(i) lies outside the cell containing Q then consider the minimum distance feasible path from P(i) to Q. Using Larson and Li's results, we know that such a minimum distance feasible path can always be replaced by a path of equal length containing a sequence of simply communicating vertices.

Let T be the last vertex before Q on such a vertex following path from P(i) to Q. Since T is either a demand point or barrier vertex, it must lie outside the cell containing Q or at one of the cell's corners, by Lemma 3.1.

If T lies on a corner of the cell, then the vertex following path satisfies the theorem since it is a minimum distance path from P(i) to Q.

We will prove that if T lies outside the cell then a minimum distance feasible path from T to Q can be found which passes through a corner of the cell or can be transformed into a path which passes through a corner without increasing its length.

Consider the first condition for simply communicating vertices. The vertices T and Q are joined by a single horizontal or vertical line segment which has no point in common with the interior of any barrier and contains no other vertex.

If Q lies in the interior of a cell, then, T which is either a demand point or a barrier vertex, must have at least one coordinate which is distinct from the set of coordinates used to define the cell boundaries. This is a contradiction. Therefore, Q must lie on the boundary of the cell under this condition.

We have already demonstrated that if Q lies on a corner the theorem is satisfied, so that we only need to consider the case when Q lies on a side of the cell between the corners. In this case, T which is either a demand point or a barrier vertex must lie on an extension of that side of the cell on which Q lies by the definition of cells (Figure 3.8). Thus, one corner of the cell lies on a feasible path from T to Q and the vertex following path will satisfy the theorem.

Consider the second condition for simply communicating vertices. A feasible path exists from T to Q which consists of a single horizontal line segment joined to a single vertical line segment.

By Lemma 3.1, the demand point or barrier vertex T must lie in one of the corner regions relative to the cell containing Q.

We have already demonstrated that if T or Q lies at a corner of the cell, then the theorem is satisfied.

If Q lies on a side of the cell, or T lies on an extension of one side of the cell, or both, then one corner of the cell lies on a feasible path from T to Q and the theorem is satisfied (Figure 3.9).

If Q lies in the interior of the cell and T lies in the interior of a corner region, the only remaining case, then we will show that the path from T to Q can always be modified without increasing its length so that the modified path passes through a corner of the cell.

Consider the example shown in Figure 3.10. Let (a_1,b_1) and (a_2,b_2) be the coordinates of T and Q respectively. Let (c,d) be the coordinates of the cell corner which lies on the side intersected by the path and is nearest to T.

Modify the path by "pushing" the horizontal line segment to the corner. That is, replace the two line segments, T to (a_1,b_2) and (a_1,b_2) to Q, with the three line segments, T to (a_1,e), (a_1,e) to (a_2,e), and (a_2,e) to Q, where e is allowed to decrease from b_2 to d.

This path has the same length as the original path from T to Q and passes through a corner of the cell when e = d. It only remains to show that this is a feasible path.

The path from T to (a_1,e) remains feasible. By Lemma 3.2, the line segments Q to (a_2,e) and (a_2,e) to (c,e) are feasible, since Q to (c,b_2) is feasible.

The line segment (a_1,e) to (c,e) can cease to be a feasible path as it is "pushed" toward the corner only if it

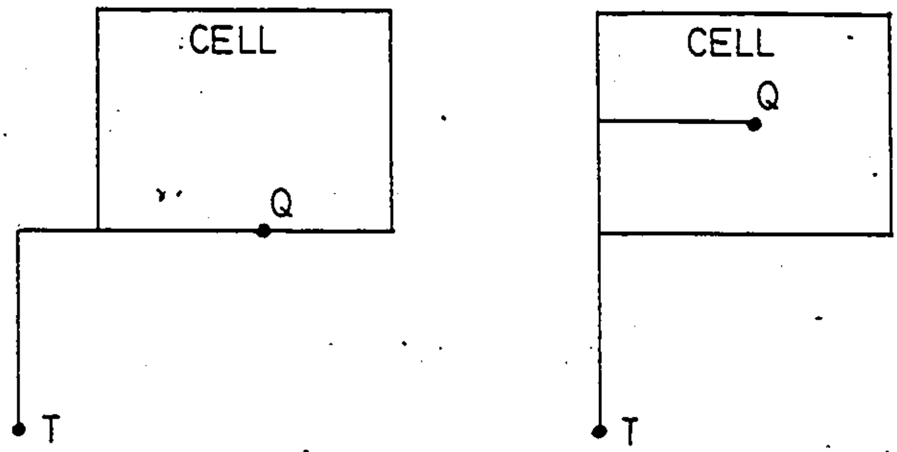

Figure 3.9 T and Q satisfy condition 2 and T or Q on side or extension of side.

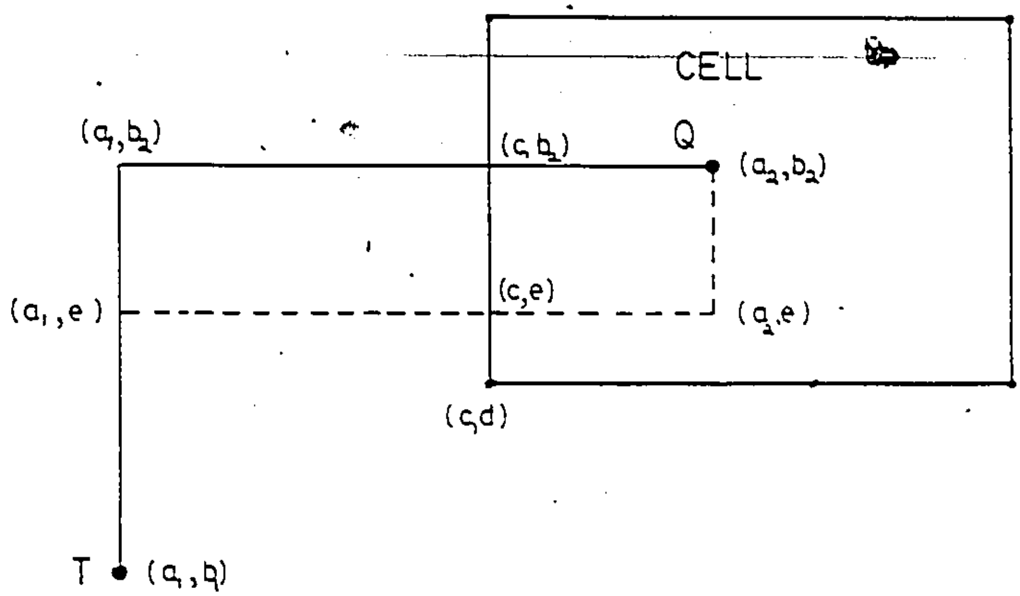

Figure 3.10 T and Q satisfy condition 2 and neither on side or extension of side.

encounters the interior of a barrier, and hence, a barrier vertex before it reaches the corner. By Lemma 3.1, no such barrier vertex can exist since any such barrier vertex would not lie at either a cell corner or outside the cell in one of the corner regions.

The proofs for T in other corner regions and for other path orientations are similar. This completes the proof of Theorem 3.1. □

3.3 Distance Function inside Cells

Consider any cell, the interior of which does not lie inside a barrier (by Lemma 3.2 the interior is entirely inside or outside barrier). Label its corners in counterclockwise order starting with 1 at the northeast corner (Figure 3.14). By Theorem 3.1, the distance from a demand point P(i) to a point Q = (x,y) in the cell is

$$d(P(i),Q) = \underset{j=1,\ldots,4}{\text{minimum}} \{d(P(i),C(j))+d(C(j),Q)\}, \quad (3.3.1)$$

where C(j), j=1,...,4 are the corners of the cell. For notational convenience and for this section only let (x(j),y(j)) represent the coordinates of the j'th corner of the cell. Since no part of a barrier lies inside the cell, $d(C(j),Q) = |x(j) - x| + |y(j) - y|$ = the rectangular distance from C(j) to Q without barriers. The distances from the demand point to the cell corners, d(P(i),C(j)) for j = 1,...,4, are constants independent of the location of the point Q inside the cell and can obtained using Larson and Li's algorithm.

In order to understand the nature of the function d(P(i),Q) inside the cell, we will examine the behaviour of the four functions [d(P(i),C(j)) + d(C(j),Q)] = [d(P(i),C(j)) + |x(j)-x| + |y(j)-y|]. In particular, we will consider the function associated with the third cell corner C(3). Let

$$z = d(P(i),C(3)) + |x(3) - x| + |y(3) - y|.$$

For any point Q = (x,y) which lies in the cell, $x \geq x(3)$ and $y \geq y(3)$, so that the equation can be simplified to

$$z = d(P(i),C(3)) + x - x(3) + y - y(3).$$

Rearranging this equation, we obtain

$$(x - x(3)) + (y - y(3)) - (z - d(P(i),C(3)) = 0$$

which is the equation of a plane in three dimensions passing through the point (x(3), y(3), d(P(i),C(3)) and which is orthogonal to the vector (1,1,-1). Let z(j) = d(P(i),C(j)) for j = 1,...,4. The equation of the plane associated with cell corner C(3) can now be slightly rewritten as

$$(x - x(3)) + (y - y(3)) - (z - z(3)) = 0.$$

Similarly, the function associated with each of the other cell corners C(j) can also be shown to be represented in three dimensions by a plane which passes through the point (x(j), y(j), z(j)). The normal vectors to the planes associated with the corners, C(1), C(2), and C(4), can be shown to be (-1,-1,-1), (1,-1,-1), and (-1,1,-1) respectively.

Inside a cell the surface representing each of the four functions, [d(P(i),C(j))+d(C(j),Q)], is a section of a plane. The edges of the section of the plane are defined by the limits of the cell (Figure 3.11). The lowest point on the section of the plane associated with the corner C(j) is found above the corner C(j) at a height of z = d(P(i),C(j)). The gradient of the plane is directed towards the interior of the cell at a 45° angle from both the x and y axes. The surface representing the distance function d(P(i),Q) in the cell is the minimum of four such sections of planes, each with its minimum at a different corner of the cell.

To help visualize the type of surfaces which can be formed by the minimum of four such planes, consider the entire planes from which the sections of planes were derived. Taking the minimum of two planes associated with diagonally opposite corners of the cell forms a surface with a ridge running at 45° to the x and y axes. The planes associated with the other pair of diagonally opposite corners also form a surface with a ridge running at 45° to the axes but this ridge is orthogonal to the preceding ridge (Figure 3.12).

The minimum of these two surfaces and also the minimum of the original four planes has three possible shapes (Figure 3.13). The orientation and the length of the ridge line depends on the values of the d(P(i),C(j)). The pyramid case can be considered as a special case of either

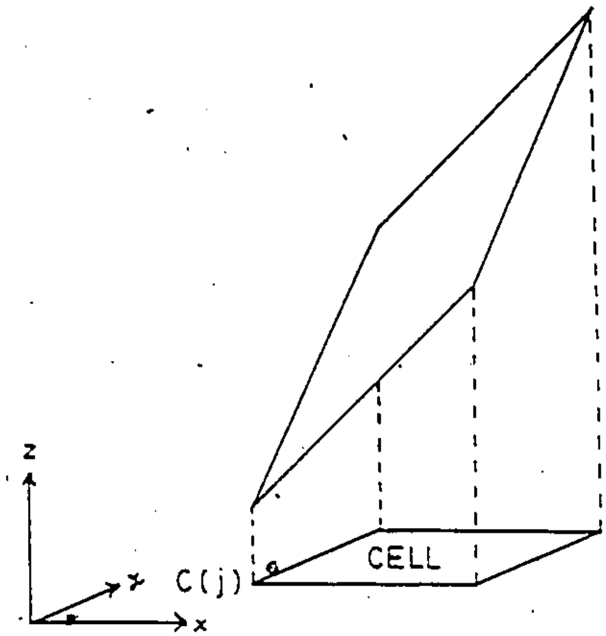

Figure 3.11 Function [d(P(i),C(j)) + d(C(j),Q)] in a cell.

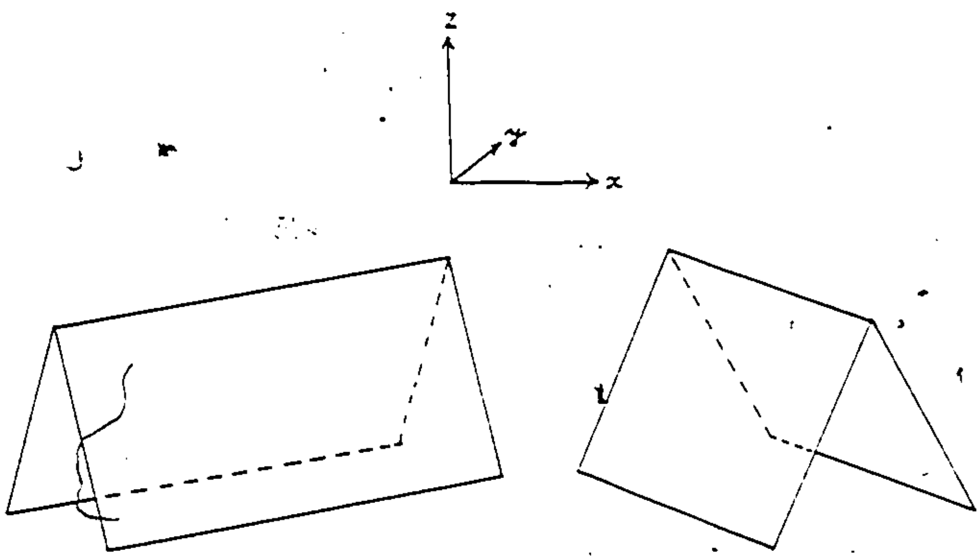

Figure 3.12 Minimum of 2 planes through opposite corners.

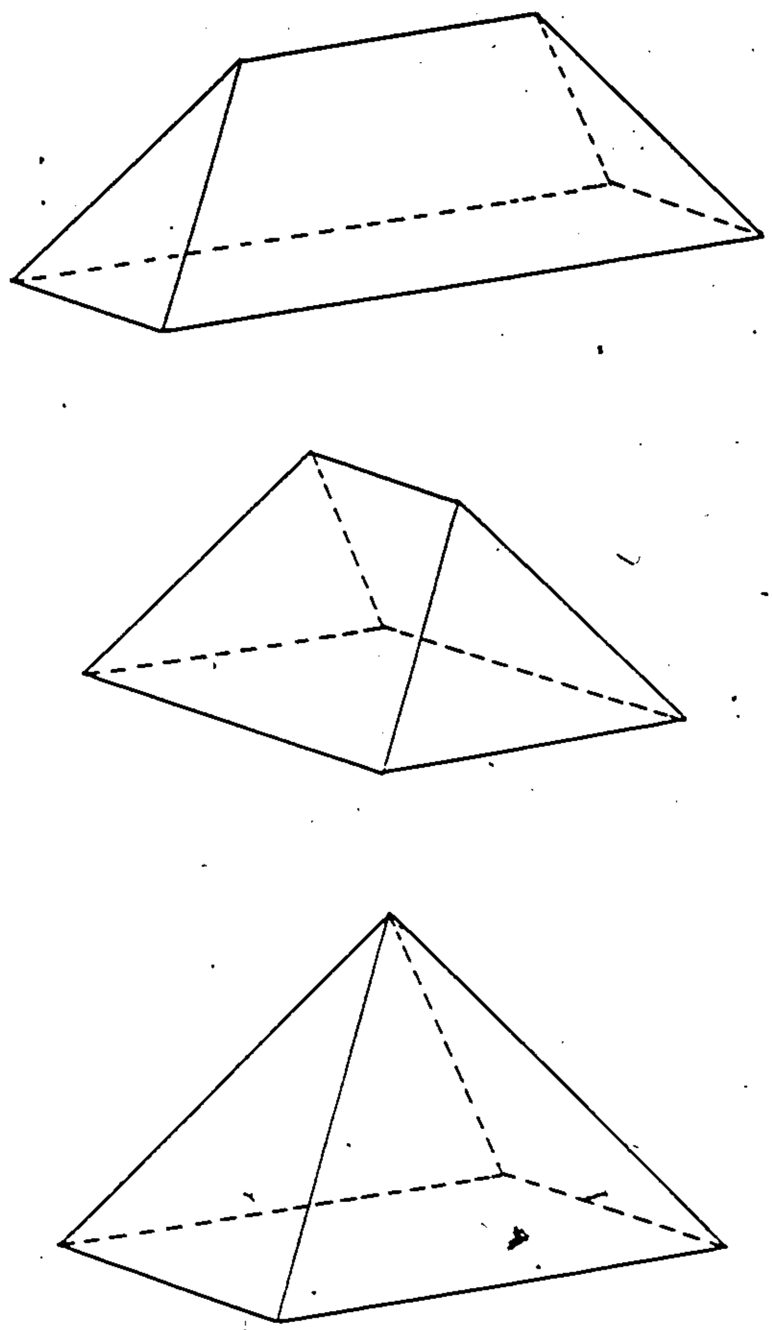

Figure 3.13 Minimum of 4 planes through cell corners.

of the other two possibilities with a ridge line of zero length. The shape of that portion of the surface which lies inside the cell depends on the location of the ridge line relative to the cell and hence on the values of the d(P(i),C(j)).

3.3.1 Level Curves inside a Cell

It is convenient to determine the region between two level curves by working with the minimum of the four planes and then clipping the resulting polygonal area so that only that part of the polygonal area which lies inside the cell is actually drawn. The region between two level curves inside the cell could also be calculated by first determining the type of pattern that the level curves have in the cell by examining the values of d(P(i),C(j)) for j = 1,...,4 and then calculating the two level curves using specialized formulae for that type of level curve pattern. It can be shown that there are 16 distinct patterns of level curves inside a cell which can be produced by variations in the values of the distances d(P(i),C(j)). By working with the minimum of the four planes and then clipping the resulting area, we can avoid dealing explicitly with the 16 possible patterns of level curves.

The equations of the planes associated with each corner are:

$$+(x-x(1)) + (y-y(1)) + (z-z(1)) = 0, \text{ for } C(1); \quad (3.3.1)$$

$$-(x-x(2)) + (y-y(2)) + (z-z(2)) = 0, \text{ for } C(2); \quad (3.3.2)$$

$$-(x-x(3)) - (y-y(3)) + (z-z(3)) = 0, \text{ for } C(3); \quad (3.3.3)$$

$$+(x-x(4)) - (y-y(4)) + (z-z(4)) = 0, \text{ for } C(4). \quad (3.3.4)$$

Let cz be the value of the distance associated with the level curve under consideration and let cz be less than the maximum function value in the cell. Finding one by one the intersection of these four planes with the plane z = cz and projecting the resulting four lines into the xy-plane yields the equations of the four lines (see Figure 3.14) which form the level curve for cz:

$$x + y = x(1) + y(1) - (cz - z(1)), \quad \text{for line } L(1);$$

$$-x + y = -x(2) + y(2) - (cz - z(2)), \quad \text{for line } L(2);$$

$$-x - y = -x(3) - y(3) - (cz - z(3)), \quad \text{for line } L(3);$$

$$x - y = x(4) - y(4) - (cz - z(4)), \quad \text{for line } L(4);$$

where line L(k) is associated with corner k. The level curve is a rectangle with sides oriented at 45° to the x and y axes. This rectangle may be partially or entirely outside the cell. The intersections of the four lines taken cyclically in pairs yields the end points of the line segments which form the level curve.

Let xmin = x(2) = x(3), xmax = x(1) = x(4), ymin = y(3) = y(4), and ymax = y(1) = y(2). From L(1) and L(2), we obtain the top corner of the rectangle with coordinates:

$$x = (xmax+xmin)/2 + (z(1)-z(2))/2, \quad (3.3.5)$$

$$y = (xmax-xmin)/2 + ymax - (2cz-z(1)-z(2))/2. \quad (3.3.6)$$

Similarly, the left corner, obtained from L(2) and L(3), has coordinates:

$$x = xmin - (ymax-ymin)/2 + (2cz-z(2)-z(3))/2, \quad (3.3.7)$$

$$y = (ymax+ymin)/2 + (z(2)-z(3))/2. \quad (3.3.8)$$

The bottom corner, obtained from L(3) and L(4), has coordinates:

$$x = (xmax+xmin)/2 - (z(3) -z(4))/2, \quad (3.3.9)$$

$$y = -(xmax-xmin)/2 + ymin + (2cz-z(3)-z(4))/2. \quad (3.3.10)$$

The right corner, obtained from L(4) and L(1), has coordinates:

$$x = xmax + (ymax-ymin)/2 - (2cz-z(1)-z(4))/2, \quad (3.3.11)$$

$$y = (ymax+ymin)/2 + (z(1) -z(4))/2. \quad (3.3.12)$$

3.3.2 Maximum Distance in a Cell

The formulae for the line segments which compose the level curves on the four-plane surface yield meaningless results when the contour value, cz, is greater than the maximum value of the distance function in the cell, cmax. Thus, it is necessary to check that cz is not greater than cmax before attempting to calculate the level curves, and we will require an expression for cmax.

From the results of subsection 3.3.1, it can be shown that the level curve for any value of cz < cmax is a rectangle which decreases in size as cz approaches cmax and that at cmax at least one pair of opposing sides of the rectangle must coincide (Figure 3.15). Thus, the value of

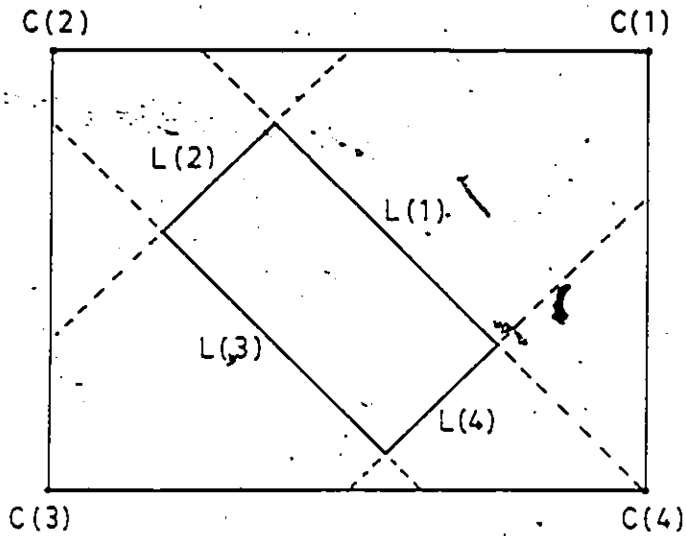

Figure 3.14 Level curve of distance function in a cell.

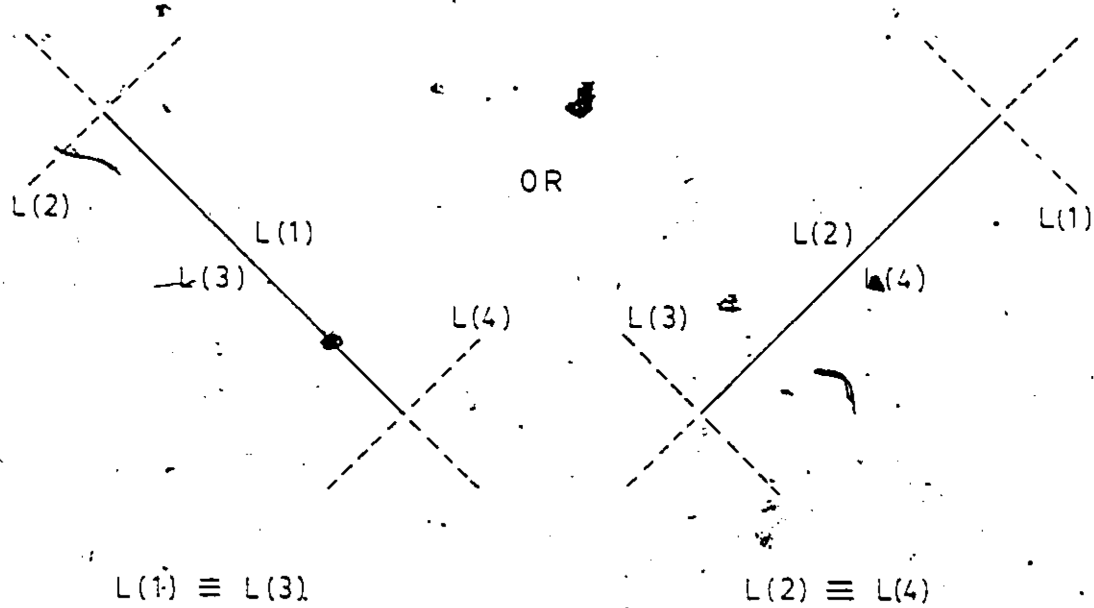

Figure 3.15 Condition for CMAX.

cmax can be found by determining the smallest value of cz for which either lines L(1) and L(3) coincide with each other or lines L(2) and L(4) coincide with each other.

Recall from subsection 3.3.1 that the equation of line L(1) is

$$x + y = x(1) + y(1) - (cz - z(1))$$

and the equation of line L(3) is

$$-x - y = -x(3) - y(3) - (cz - z(3)).$$

Let $cz_{1,3}$ be the value of cz for which these two lines coincide. The lines defined by ax + by = c and Ax + By = C coincide if and only if there exists a $k \neq 0$ such that a = kA and b = kB and c = kC. Thus, the lines L(1) and L(3) coincide if

$$x(1) + y(1) - (cz-z(1)) = x(3) + y(3) + (cz-z(3)).$$

Solving for $cz_{1,3}$ we obtain

$$cz_{1,3} = [z(1) + z(3) + x(1) - x(3) + y(1) - y(3)] / 2.$$

Similarly, it can be shown that the value of cz for which the lines L(2) and L(4) coincide is

$$cz_{2,4} = [z(2) + z(4) + x(4) - x(2) + y(2) - y(4)] /2.$$

The value of cmax is equal to the smaller of the two values, $cz_{1,3}$ and $cz_{2,4}$. Letting H = y(1)-y(3) = y(2)-y(4) be the height (in y-direction) of the cell and letting W = x(1)-x(3) = x(4)-x(2) be the width, we obtain the expression

$$cmax = \min\{ [z(1)+z(3)+H+W]/2, \quad [z(2)+z(4)+H+W]/2 \}. \qquad (3.3.13)$$

The preceding expression for cmax is valid for any real values assigned to the variables z(j), j = 1,...,4, and it gives the maximum value of z found on the surface generated by taking the minimum of the four planes (equations 3.3.1 to 3.3.4). This is sufficient for the purpose of checking the value of cz before attempting to calculate the level curves for the value cz.

Since we will also use equation 3.3.13 to calculate the maximum value of the distance function d(P(i),Q) inside the cell, we must show that there can always be found at least one point Q^* in the cell or on the boundary of the cell such that $d(P(i),Q^*)$ = cmax. This is not necessarily true when the z(j), j = 1,...,4 are allowed to assume any real values. However, the z(j)s represent distances in our problem and this makes it possible to prove the existence of such a point Q^* in our case.

This can be done by substituting equation 3.3.13 for cmax into the formulae for a corner point of the rectangular level curves (for instance, equations 3.3.5 and 3.3.6) in order to obtain the coordinates of a point for which d(P(i),Q) = cmax. One can then show that the corner point lies in the cell or on the cell boundary by using the triangle inequalities which the distances z(j), j = 1,...,4 must satisfy. As an example of the triangle inequalities, consider the distances d(P(i),C(1)) = z(1), d(P(i),C(3)) = z(3), and d(C(1),C(3)) = |x(1)-x(3)| + |y(1)-y(3)| = W + H:

These distances must satisfy the triangle inequality, $d(P(i),C(3)) \leq d(P(i),C(1)) + d(C(1),C(3))$ or equivalently $z(3) \leq z(1) + W + H$. In the same manner, we can prove that $z(1) \leq z(3) + W + H$. Similar but not necessarily identical inequality relations can be established between all other pairs of z(j)s.

3.3.3 Clipping Area between Level Curves

Any of the four line segments of a level curve calculated in subsection 3.3.1 may lie partly or completely outside the cell. Since Theorem 3.1 is valid only for points inside a cell, those portions of the line segments and the areas between line segments which lie outside the cell may not be valid. Therefore, the areas must be clipped in order that only those portions which lie inside the cell are actually shaded by the algorithm.

The problem of determining that portion of a polygon which lies inside a rectangular window is a well-studied problem in computer graphics for which efficient algorithms are available such as the Sutherland-Hodgman algorithm (Foley and Van Dam, p. 450-454). By splitting the area between the two level curves as shown in Figure 3.16, we obtain four trapezoids which can be clipped by the Sutherland-Hodgman algorithm since the trapezoids are simple polygons without holes.

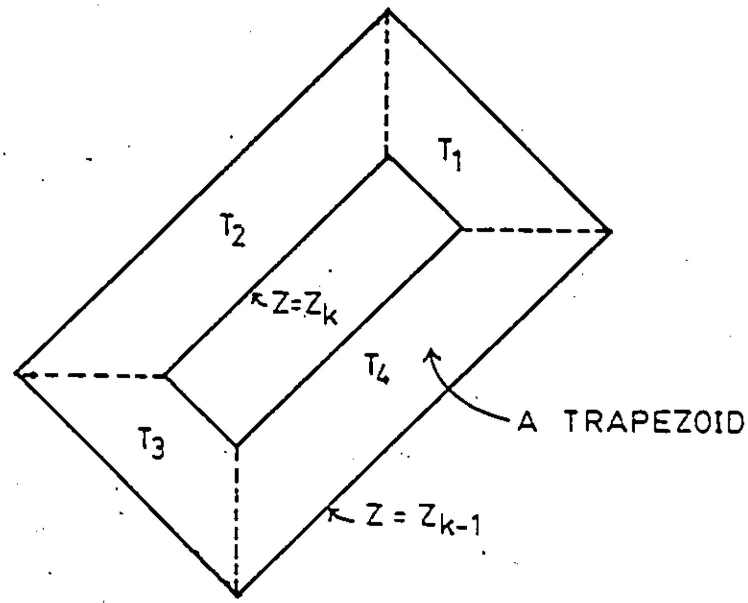

Figure 3.16 Splitting region between level curves into 4 trapezoids.

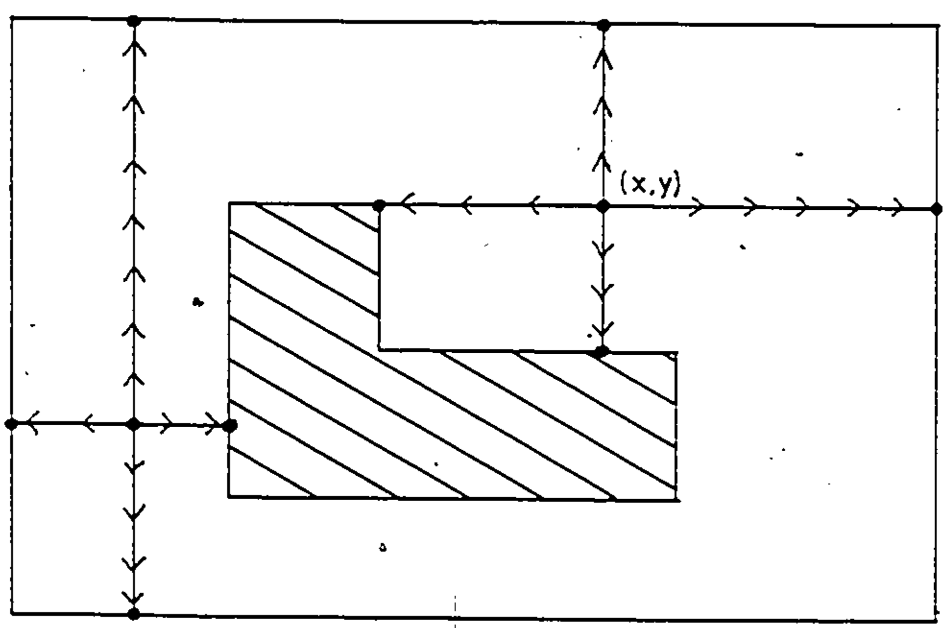

Figure 3.17 Vertex-seeking trees.

3.4 Finding Distances to Cell Corners

The results of sections 3.2 and 3.3 provide a method of calculating the distance to any point inside a cell and a method of filling the area between level curves inside a cell whenever we know the distances to the corners of the cell. In principle, the distances to all cell corners could be found by directly applying Larson and Li's POLYPATH algorithm. However, this would severely limit the size of the problems which could be solved. Instead, we will rely on Larson and Li's theory and methods while exploiting our problem's special features to reduce the problem's size and eliminate unnecessary computations.

3.4.1 Reducing Problem Size

We will calculate the number of cell corners as defined in section 3.2 and show that it is inconveniently large even for moderate numbers of demand points and barrier vertices.

The demand points contribute at most N_P distinct x-values to the set {wxmin, wxmax; x(i), i=1,..,N_P; x(k,m), k=1,...,v_m and m=1,...,N_B} and the edges of the problem universe contribute two distinct x-values to the set. Let NBV be the total number of barrier vertices in the problem. Since the barriers are right-angled polygons aligned with the coordinate axes, each barrier vertex shares its x-coordinate with at least one other barrier vertex. Thus, the barrier vertices contribute at most NBV/2 distinct x-values to the set. Therefore, the total number of possible distinct x-values in the set is $(N_p + NBV/2 + 2)$. Similarly, the total number of possible distinct y-values in the set {wymin, wymax; y(i), i=1,...,N_p; y(k,m), k=1,...,v_m and m=1,...,N_B} is also $(N_p + NBV/2 + 2)$. Of course, the number of distinct values in the sets may be less than this if some of the points in the problem happen to share the same coordinate.

The problem universe is divided into cells by vertical lines at each distinct x-value and horizontal lines at each distinct y-value in the preceding sets (Figure 3.4). Therefore, the maximum number of intersections and cell corners is $(N_p + NBV/2 + 2)^2$. A moderate sized problem with $N_p = 30$ and NBV = 30 could have as many as $(47)^2 = 2209$ cell corners and each cell corner would have associated with it the distances to each of the 30 demands. Storage considerations alone would limit such a direct and naive approach to small problems.

We note that the distance from any given demand point depends on the position of the demand point and the positions of the barriers but it is not affected by the positions of any of the other demand points. Thus, to find the distance from any given demand point to any point we can solve a reduced problem with all of the barriers present but only that one demand point included in the reduced problem.

The maximum number of distinct x-values or y-values in such a reduced problem is (NBV/2 + 3) and the maximum number of cell corners is $(NBV/2 + 3)^2$. For such a reduced problem with NBV = 30, the maximum number of cell corners is $(18)^2$ = 324 and each corner would have only one distance associated with it.

Although we must solve and store distances for N_p reduced problems, this is a substantial improvement over the naive approach. The only disadvantage to using N_p reduced problems is that the problem universe is divided into a slightly different set of cells in each reduced problem. Since the graphical method deals with the demand points one at a time in both its filling of contours and its calculation of minimum (or maximum) distance, this is not a significant disadvantage. It is only necessary to store the x-values and y-values which were used to divide the problem universe into cells for each of the N_p demand points. At worst, this requires that we store an additional 2(NBV/2 + 3) real values for each demand point.

3.4.2 Distances from One Demand Point

For each demand point, we must solve a reduced problem with that demand point and all of the barriers present in order to find the distances from the demand point to all accessible cell corners in the reduced problem. Some cell corners may be inaccessible because they happen to lie in the interior of a barrier (Figure 3.4). Since the distance to such inaccessible corners is undefined, they must be identified by our procedure.

To solve the reduced problem, we apply Larson and Li's result that if a feasible path exists between two vertices, then a minimum distance feasible path can always be found which contains a sequence of simply communicating vertices. The demand point, the barrier vertices, and the cell corners are all "vertices" in Larson and Li's network terminology. Thus, we can find the distance to any accessible cell corner by first finding those vertices which are simply communicating vertices and then applying Dijkstra's shortest path algorithm (Minieka, p. 44-45) to the resulting network of simply communicating vertices. The length of the arc joining two simply communicating vertices in the network is just the rectilinear travel distance between them. This distance is not increased by the barriers to travel because of the manner in which simply communicating vertices were defined.

Larson and Li's POLYPATH algorithm first finds which vertices are simply communicating vertices and then applies a shortest path procedure which exploits special features of the problem in order to achieve a computational complexity fractionally less than that of Dijkstra's shortest path algorithm. However, the computational complexity of both Dijkstra's and Larson and Li's shortest path procedures grow as the square of number of vertices in the network (Larson and Li). Since there could be as many as $(NBV/2 + 3)^2$ vertices, the complexity of solving one reduced problem would be proportional to $(NBV/2 + 3)^4$. Fortunately, it is not necessary to solve the reduced problem with all $(NBV/2 + 3)^2$ cell corners as vertices in the problem at one time. The distance from the demand point to any given cell corner depends only on the positions of the demand point, the barriers, and that cell corner. Thus, we can solve the reduced problem by solving one smaller problem for each cell corner in the reduced problem.

Consider one of the small problems which has only the demand point, the barriers, and one cell corner present. By Larson and Li's result, there exists a minimum distance path from the demand point to the corner point which consists of a sequence of simply communicating vertices. The last link on this path is either from one of the barrier vertices to the corner or from the demand point to the corner. The total length of this path is the length of the path from the demand point to the next to last vertex on the path plus the rectilinear distance from the next to last vertex to the corner point. If we know the minimum travel distance from the demand point to each of the vertices of the barriers, then we can find the minimum travel distance to any cell corner by finding that node, barrier vertex or demand point, which simply communicates with the cell corner and which minimizes the total path length. Thus, we only need to find the minimum travel distance from the demand point to the barrier vertices once for each reduced problem. Since this requires solving a shortest path problem with NBV + 1 nodes, the complexity is proportional to $(NBV + 1)^2$. In subsection 3.4.3, we will show that creating the network for the shortest path problem with NBV + 1 vertices is also of complexity $O(NBV^2)$.

To completely solve a reduced problem, we must find the minimum travel distance to, at most, $(NBV/2 + 3)^2$ cell corners and for each cell corner this requires checking each of NBV barrier vertices and the one demand point to determine if they simply communicate with the cell corner. Thus, the complexity of solving a reduced problem by splitting it into smaller problems is proportional to the cube of NBV. This is a substantial improvement over the $O(NBV^4)$ computational complexity which would have resulted if we applied Larson and Li's methods directly to the entire reduced problem containing all of the cell corners as vertices.

We briefly summarize our method of finding the distances to the cell corners in a reduced problem containing one demand point. The vertex-seeking trees mentioned are explained in subsection 3.4.3.

First, we determine the minimum rectilinear travel distance from the demand point to each of the barrier vertices. This is accomplished by creating a network containing as nodes the demand point and all of the barrier vertices. Next, we determine all pairs of nodes which simply communicate and join them by arcs with length equal to the rectilinear distance (unaffected by barriers) between the nodes. Then, we apply Dijkstra's shortest path algorithm to find the minimum distance from the node representing the demand point to the rest of the nodes. We save the distances to the barrier vertices and the vertex-seeking trees used to determine which pairs of vertices are simply communicating vertices.

For each cell corner in the reduced problem, we grow a vertex-seeking tree rooted at that cell corner. If a cell corner is inaccessible because it lies in the interior of a barrier, it will be found by the procedure which grows vertex-seeking trees and skipped after it has been flagged as inaccessible. Using the vertex-seeking tree rooted at the cell corner and those trees rooted at the network nodes which were saved from previous step, we determine those nodes which simply communicate with the cell corner. Among the nodes which communicate simply with the cell corner, we find that node which minimizes the sum of the distance to the node as determined by the shortest path procedure plus the rectilinear distance (unaffected by barriers) from the node to the cell corner. The minimum value found is the minimum distance to the cell corner.

We now have an efficient procedure for calculating the distances from the demand point to all cell corners. The next two subsections explain Larson and Li's method of finding simply communicating vertices and discuss an integer representation of part of the problem which reduces the storage requirements and simplifies some of the calculations.

3.4.3 Finding Simply Communicating Vertices

Larson and Li define simply communicating vertices in terms of vertex-seeking trees rooted at the vertices. Our definition of simply communicating vertices in subsection 3.2.2 is somewhat easier to understand and is consistent with Larson and Li's definition in the case where the barriers are required to be right-angled polygons aligned with the travel directions. However, their definition in terms of vertex-seeking trees is more convenient when actually determining which pairs of vertices are simply communicating vertices.

For the special case of right-angled barriers aligned with the travel directions, we can simplify Larson and Li's definition of a vertex-seeking tree.

Definition 3.3: A vertex-seeking tree rooted at the point (x,y) is the union of four rays, called probes, originating at the point (x,y) and extending in the positive x direction, the negative x direction, the positive y direction, and the negative y direction until they encounter either the interior of a barrier or a barrier vertex or a demand point or the boundary of the problem universe (Figure 3.17). We define the rays to include both of their end points.

For the case of right-angled barriers and using our simplified definition of vertex-seeking trees, we can define two vertices as being simply communicating vertices if their vertex-seeking trees have at least one point in common. This is a very convenient definition to use. Once we have constructed the vertex-seeking trees rooted at each vertex, we only need to check two pairs of line segments for intersections in order to determine if a pair of vertices communicate simply. The intersection tests which must be done for each pair of vertices are very simple operations requiring at most eight comparisons of coordinates. Since the computational effort required to test a pair of vertices does not depend on NBV, the complexity of finding all pairs of simply communicating vertices in a set of NBV + 1 vertices has a complexity of $O(NBV^2)$.

Growing the vertex-seeking trees is a more difficult operation but it only has to be done once for each vertex and the resulting vertex-seeking tree stored. For our simplified vertex-seeking trees, we only need to store the lengths of the four probes to fully describe a vertex-seeking tree rooted at (x,y).

The mechanics of constructing vertex-seeking trees is fairly straight-forward. Before constructing the vertex-seeking trees, we break the boundaries of the barriers and the boundary of the problem universe into a set of horizontal line segment barriers and a set of vertical line segment barriers (Figure 3.18). The vertical line segments are sorted according to their x-coordinates and the horizontal line segments are sorted according to their y-coordinates. For each line segment, we also keep a code indicating on which side of the line segment barrier travel is possible. These two sets of line segment barriers contain all of the information about the original barriers and the boundary of the problem universe. Creating the sorted lists of line segment barriers only needs to be done once for each problem since they do not depend on which demand point is included in a particular reduced problem. The complexity of creating the sorted lists of (NBV/2+2) line segment barriers is O(NBV log NBV), since n items can be sorted in O(n log n) time.

To construct a vertex-seeking tree at some point (x,y), we first check if the point lies on any of the line segments. If it doesn't lie on any line segment, then from (x,y) we grow horizontal probes in the positive and negative x directions until each probe intersects a vertical line segment (Figure 3.19). Similarly, we grow vertical probes until they intersect horizontal line segments. If a probe

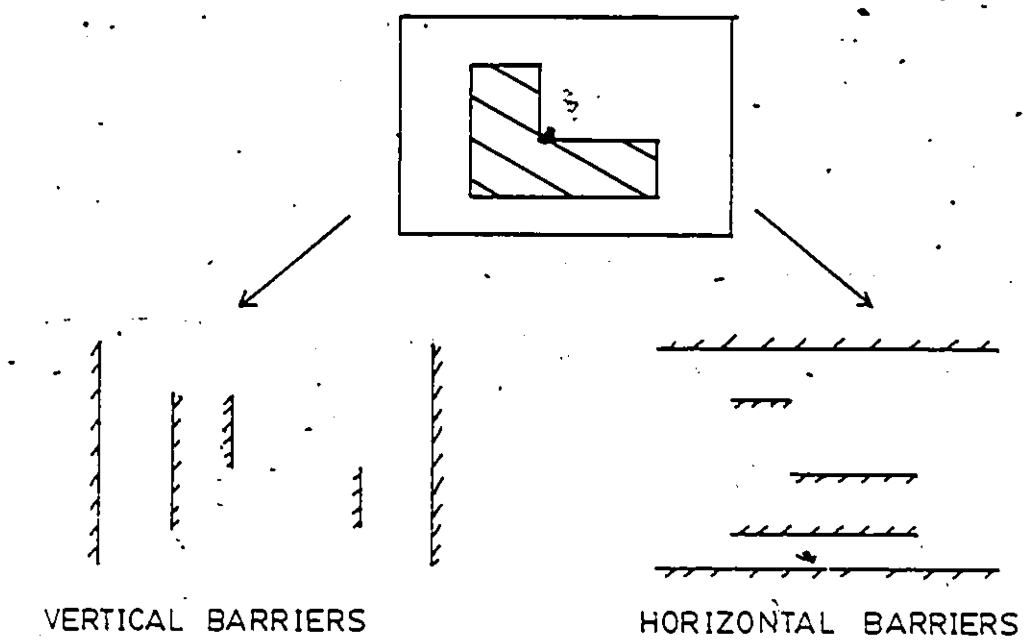

Figure 3.18 Breaking boundaries into line segment barriers.

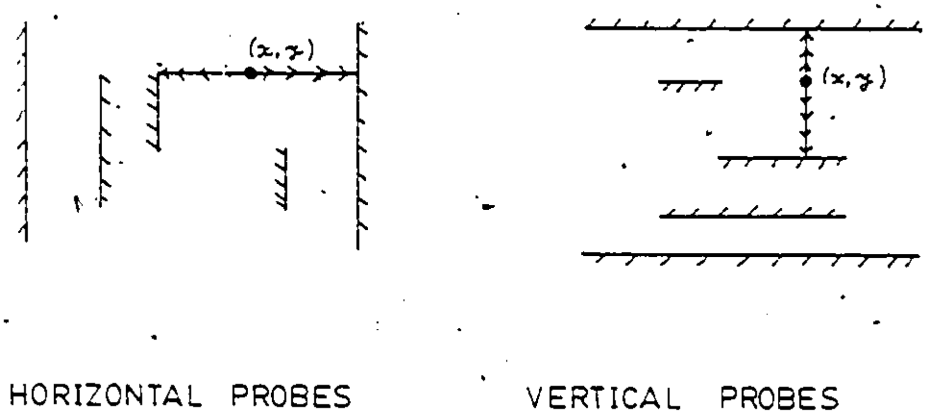

Figure 3.19 Growing probes of vertex-seeking tree.

strikes a line segment from that side of the line on which travel is not possible, then the point (x,y) must lie in the interior of a barrier and be inaccessible.

If the point (x,y) lies on a horizontal line segment between the end points of the line segment, then the end points of the line segment become the end points of the horizontal probes (Figure 3.20). The vertical probe toward the side which is open to travel is grown in the usual manner and the vertical probe in the opposite direction has zero length. If the point lies on a vertical line segment between the end points the result is similar.

If (x,y) lies at the end point of one horizontal line segment and at the end point of one vertical line segment, then it lies either at a corner of a barrier or at a corner of the boundary of the problem universe (Figure 3.21). The end point of one horizontal probe will coincide with the opposite end of the horizontal line segment on which it lies and the end point one of the vertical probes is similarly determined. If the corner on which (x,y) lies is an exterior corner, then the remaining horizontal and vertical probes are grown in the usual manner. If the corner is an interior corner, then the remaining two probes have zero length. Whether a corner is an exterior or interior corner can be determined by examining the orientation of the line segments in relation to the sides of the line segments which are open to travel.

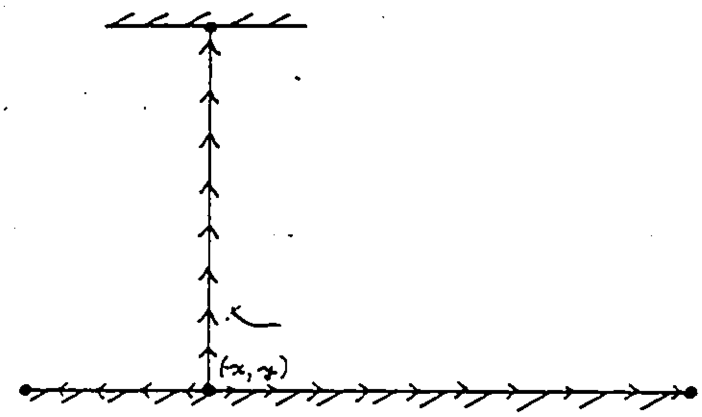

Figure 3.20 Root on line segment between end points.

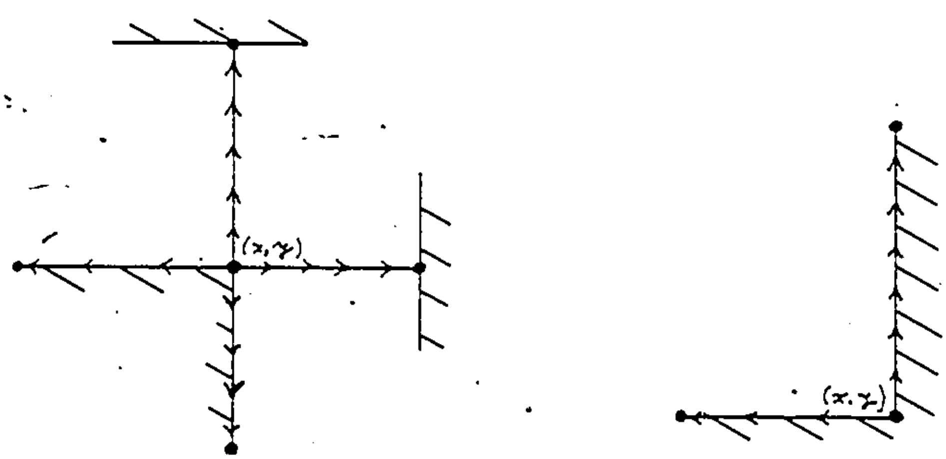

Figure 3.21 Root on a corner.

Finding the position of the root (x,y) in relation to the line segment barriers and then growing the four probes of the vertex-seeking tree until each collides with a line segment barrier requires computational effort which is proportional to the number of line segment barriers in the lists and hence is at worst proportional to NBV/2 + 2. Thus, the complexity of constructing and storing the NBV + 1 vertex-seeking trees rooted at the demand point and the barrier vertices is at worst $O(NBV^2)$.

In subsection 3.4.2, we stated that the network of NBV + 1 nodes, which we must solve to obtain the distances from the demand point to all barrier vertices, could be constructed with computational effort of only $O(NBV^2)$. Creating the network requires creating and storing lists of line segment barriers, growing and storing NBV + 1 vertex-seeking trees, and then finding all pairs of simply communicating vertices using the vertex-seeking trees. Since these steps have complexity O(NBV log NBV), $O(NBV^2)$, and $O(NBV^2)$, respectively, we have shown that creating the network has a complexity of $O(NBV^2)$.

3.4.4 Integer Representation

An integer representation of some parts of the problem can be used to reduce storage requirements on those computers which require less memory for integer variables than for real variables.

Consider the process of constructing vertex-seeking trees and of determining if two vertices simply communicate by checking if their vertex-seeking trees have a point in common. Both processes depend upon comparisons of coordinates to determine if one coordinate is greater than, equal to, or less than another coordinate. The actual amount by which one coordinate exceeds another does not affect the result of a comparison. Neither introducing cell corners nor growing vertex-seeking trees adds any new values to the sets of distinct x and y coordinate values which we introduced in the definition of cells. Thus, the x-coordinate values being compared come from a small set of real x-coordinate values and similarly for the y-coordinate values. The results of the tests to determine which vertices simply communicate would be unaffected if we replaced each real x-coordinate by an integer value equal to its position in the set of distinct x-values when that set has been ordered by increasing size. For example, if 10.7 is the third value in the set of distinct x-values after it has been sorted into order of increasing value, then we replace each x-coordinate in the problem which has the real value 10.7 with the integer value 3 as the new x-coordinate. The y-coordinates are replaced by integers in the same manner. This also allows us to represent each vertex-seeking tree by four integers rather than four real values.

After using the integer representation of the problem to determine which vertices communicate simply, we must use the real coordinates of the vertices to calculate the rectilinear distance between them before we apply a shortest path algorithm to the network of simply communicating vertices. The integer representation also allows us to code the routines which grow vertex-seeking trees and test for simply communicating vertices using comparisons of integer values rather than comparisons of real values. This simplifies the comparisons since we do not have to take into account the possibility of round-off errors which we would have to guard against when testing real values for equality. Although this integer representation of part of the problem is not necessary, it is both convenient and easily implemented.

3.4.5 Finding Closed Cells

Although the determination of open and closed cells is not a part of the process of finding the distances to cell corners, it is included in this section because it is closely related.

From Lemma 3.2 of subsection 3.2.1, we know that the interior of a cell is either entirely inside a barrier and hence closed to travel or entirely outside all barriers. It cannot be determined whether a given cell is open or closed to travel merely by examining the distances from a demand to the cell corners since all corners of a cell may be accessible but the interior of the cell may still lie inside a barrier. Since the contour filling routines must skip cells which are closed to travel, the closed cells must be identified and the results saved for use by the contour filling routines. A simple method of checking whether a cell is open or closed is to attempt to grow a vertex-seeking tree rooted at the center of the cell. If the cell is closed, then the probes will strike the line segment barriers from the sides which are not open to travel indicating that the point at the center of the cell is inaccessible and that, by Lemma 3.2, the entire cell interior is inaccessible.

3.5 Maximin Location with Barriers

3.5.1 Maximin Location Algorithm

We have divided the maximin algorithm into two phases. The first phase calculates the distances to all cell corners and determines which cells are closed to travel. This data is then stored for use by the second phase. In the second phase, we apply the interactive graphical approach to find an optimum solution.

In the following description of the maximin algorithm, we do not include the integer representation discussed in subsection 3.4.4 because it is not an essential part of the solution procedure. Also, its inclusion would greatly complicate our description of the algorithm.

Phase I: Data Preparation

Step 1. Initialization

Choose the limits of the problem universe, wxmax, wxmin, wymax, and wymin such that it encloses all demand points, all barrier vertices, and all feasible points in the bounded feasible region F.

Set I = 1.

Step 2. Divide Reduced Problem into Cells

Find all distinct x-values in the set {wxmin, wxmax, x(I); x(k,m), k=1,...,v_m and m=1,...,N_B}. Sort the x-values found into ascending order. Let VLINES be a vector of the distinct x-values sorted into ascending order and let NVLINES be the number of distinct x-values. Find all distinct y-values in the set {wymin, wymax, y(I); y(k,m), k=1,...,v_m and m=1,...,N_B}. Sort the y-values into ascending order. Let HLINES be a vector of the distinct y-values sorted into ascending order and let NHLINES be the number of distinct y-values. VLINES specifies the set of vertical lines which divide the problem universe into cells and HLINES specifies the horizontal dividing lines.

Step 3. Create Sets of Line Segment Barriers

Create a set of vertical line segment barriers consisting of the two vertical edges of the problem universe and all of the vertical barrier sides. For each line segment include a code indicating on which side of the line travel is possible. Similarly, create a set of horizontal line segment barriers. Sort the vertical line segments so that the vertical line segments can be stored in order of increasing x-coordinate. Sort and store the horizontal line segments in order of increasing y-coordinate.

Step 4. Grow Vertex-Seeking Trees at All Network Nodes

Grow vertex-seeking trees rooted at the demand point (x(I),y(I)) and at each of the barrier vertices (x(k,m),y(k,m)) for $k=1,...,v_m$ and $m=1,...,N_B$.

Step 5. Find Distances to All Barrier Vertices

Create a network of simply communicating vertices. The nodes of the network consist of the demand point and the barrier vertices. If the vertex-seeking trees rooted at a pair of nodes have a point in common, then that pair of nodes communicate simply. Join all pairs of simply communicating nodes in the network by arcs of length equal to the rectilinear distance between the two points. For example, if the vertex-seeking trees rooted at the demand point (x(I),y(I)) and at barrier vertex (x(k,m),y(k,m)) have a point in common, then the nodes should be joined by an arc with length equal to the rectilinear distance between the points, $|x(I)-x(k,m)| + |y(I)-y(k,m)|$.

Find the distance from the demand point node to all other nodes in the network using Dijkstra's shortest path algorithm. Save the vertex-seeking trees created in Step 4 and save the distances from the demand point to all other nodes.

Step 6. <u>Find Distances to Cell Corners</u>

For j = 1,...,NHLINES and k = 1,...,NVLINES, attempt to grow a vertex-seeking tree rooted at the cell corner with coordinates (VLINES(k),HLINES(j)).

If the attempt to grow the tree fails because the point at which it is rooted is inaccessible, then set the distance to the cell corner D(j,k) equal to -999 in order to flag that corner as inaccessible.

Otherwise, find all nodes of the network with which the cell corner simply communicates. For each node with which the corner simply communicates, calculate the sum of the distance from the demand point to the node, as found in Step 5, plus the rectilinear distance from the node to the cell corner. For example, if the barrier vertex (x(k,m),y(k,m)) was found to be a distance of d units from the demand point in Step 5 and if it simply communicates with the cell corner, then calculate the sum of d plus (|x(k,m)-VLINES(k)| + |y(k,m)-HLINES(j)|). Set the distance D(j,k) equal to the smallest sum found over all of those nodes which simply communicate with the cell corner.

Step 7. <u>Find Closed Cells</u>

For j = 1,...,NHLINES-1 and k = 1,...,NVLINES-1, attempt to grow a vertex-seeking tree rooted at the coordinates ((VLINES(k) + VLINES(k+1))/2, (HLINES(j) + HLINES(j+1))/2). In other words, attempt to grow a tree at the center of the k'th cell in the j'th row of cells.

If the attempt to grow a tree fails because the center point is inaccessible, then set CCLEAR(j,k) = -99 to indicated that cell is closed to travel. Otherwise, set CCLEAR(j,k) = +1.

Step 8. Save Corner Data

Save the values NVLINES and NHLINES, the vectors VLINES and HLINES, and the matrices D and CCLEAR which are associated with the demand point I for use by the interactive phase of the algorithm.

If $I = N_p$ then stop. Otherwise, set I = I + 1 and go to Step 2.

Phase II: Interactive Optimization

Step 1. Initialization

Choose the problem universe as that region of the plane which is to be represented on the screen. Shade the infeasible region for facility placement. Shade the interiors of all barriers to travel.

If no unshaded point remains on the screen, then stop since no feasible solution exists.

Otherwise, set k = 1 and $z_0 = 0$.

Step 2. Improve Value of Objective Function

Choose any unshaded point $X_k = (x_k, y_k)$, preferably one which lies near the center of an unshaded area.

For each demand point I, retrieve the corner data from storage. That is, retrieve the values NVLINES and

NHLINES, the vectors VLINES and HLINES, and the matrices D and CCLEAR associated with the demand point I. Determine in which open cell X_k lies by finding m and n such that VLINES(n) $\leq x_k \leq$ VLINES(n+1), HLINES(m) $\leq y_k \leq$ HLINES(m+1), and CCLEAR(m,n) = +1. Calculate $f_I(X_k)$

$$= w_I \min\{ D(m,n) + |x_k-VLINES(n)| + |y_k-HLINES(m)|,$$
$$D(m+1,n) + |x_k-VLINES(n)| + |y_k-HLINES(m+1)|,$$
$$D(m,n+1) + |x_k-VLINES(n+1)| + |y_k-HLINES(m)|,$$
$$D(m+1,n+1) + |x_k-VLINES(n+1)| + |y_k-HLINES(m+1)|\}.$$

Calculate $z_k = \min \{ f_I(X_k);\ I=1,\ldots,N_P\}$.

Step 3. Eliminate Dominated Regions

Shade the area between the level curves of $f_I(X)$ corresponding to the values z_{k-1} and z_k for each $I=1,\ldots,N_P$. That is, for each demand point I we retrieve the corner data associated with it. For m = 1,...,NHLINES-1 and n = 1,...,NVLINES-1, if CCLEAR(m,n) = -99, then skip to the next cell since cell (m,n) is closed. Otherwise, calculate CMAX for cell using the corner distances D(m,n), D(m+1,n), D(m,n+1), and D(m+1,n+1). If both z_{k-1}/w_I and z_k/w_I are greater than CMAX or if both z_{k-1}/w_I and z_k/w_I are less than min {D(m,n), D(m+1,n), D(m,n+1), D(m+1,n+1)}, then skip to the next cell because no part of cell (m,n) needs to be shaded. Otherwise, calculate the region which lies between the level curves as described in section 3.3 and shade it after clipping the region against the cell boundaries.

Set k = k + 1.

If only a few points (pixels) remain unshaded, then go to Step 4. Otherwise, go to Step 2.

Step 4. Accept or Expand Scale

If the size of the remaining unshaded area is sufficiently small so that the uncertainty in the location of the optimum is acceptable, then choose a point in the remaining unshaded area as a near-optimal location X^*. Calculate $z^* = \min \{ f_I(X^*); I=1,\ldots,N_P\}$. Stop.

Otherwise, choose a small rectangle enclosing the unshaded region and expand the scale so that the rectangle fills the screen. Shade the infeasible region and the interiors of all barriers. For all $I = 1,\ldots,N_P$, shade the area between the level curves of $f_I(X)$ corresponding to the values Z_0 and z_{k-1}. Go to Step 2.

3.5.2 Demonstration Program

The algorithm for single facility maximin location with rectilinear distances and right-angled barriers to travel was implemented on a Digital Equipment Corporation (DEC) Professional 350 personal computer. Phase I and Phase II of the algorithm were implemented as separate FORTRAN 77 programs. The interactive optimization program for Phase II utilized DEC's Core Graphics Library of graphic subroutines which follow the proposed Core Graphics System standard of the ACM SIGGRAPH Graphics Standards Planning Committee (1979). Excluding comment lines, the Phase I program consists of approximately 1300 FORTRAN lines and the Phase II program consists of approximately 1800 lines of FORTRAN.

A maximum of 30 demand points can be accommodated by the programs and each demand point may have a different positive weight associated with it. The programs allow a maximum of 30 barrier vertices which can be divided among as many as 7 different right-angled barriers to travel. The feasible region is assumed to consist of one or more polygons up to a maximum of 20 with the total number of vertices not exceeding 200.

The Phase I program reads the barrier, the demand point, and the feasible region information from text files. It writes the corner data that it calculates for each demand point onto a hard disk in the form of an unformatted FORTRAN file which is read by the Phase II program.

During the interactive optimization phase, positions on the screen are indicated by moving the cursor about using the keyboard. Since there may be more than one small region which the user would like to expand in Step 4 of Phase II, the program allows more than one region to be picked for expansion. The most recently picked region is expanded and the information on the other regions is placed on a stack for later retrieval.

3.5.3 Example

A large example with 30 demand points and a total of

30 barrier ~~vertices was~~ created (Figure 3.22). Each of the demand points was assigned a weight of 1. The bounded feasible region for the example is a relatively simple polygon with 7 vertices. The coordinates of the demand points, the barrier vertices, and the vertices of the feasible region are given in Appendix A.

The best location found was at (16.251, 19.152) on the boundary of the feasible region. The minimum distance to the nearest demand point, number 14 situated at (19.0, 14.0) was found to be 7.901 units. The distance to the next nearest demand point, number 13 at (12.0, 15.5), was found to be 7.903 units. The third nearest demand point was number 12 situated at (10.0, 17.0) and the distance to it was 8.403 units.

3.6 Minimax Location with Barriers

Since only minor changes are required to convert the maximin algorithm into an algorithm for minimax location problems, we will not repeat the rather lengthy description of the algorithm here. Instead, we will discuss the differences between maximin and minimax location problems and specify those changes which must be made to the maximin algorithm given in subsection 3.5.1 to convert it to a minimax algorithm.

3.6.1 Differences from Maximin Location

A bounded feasible region for facility placement is

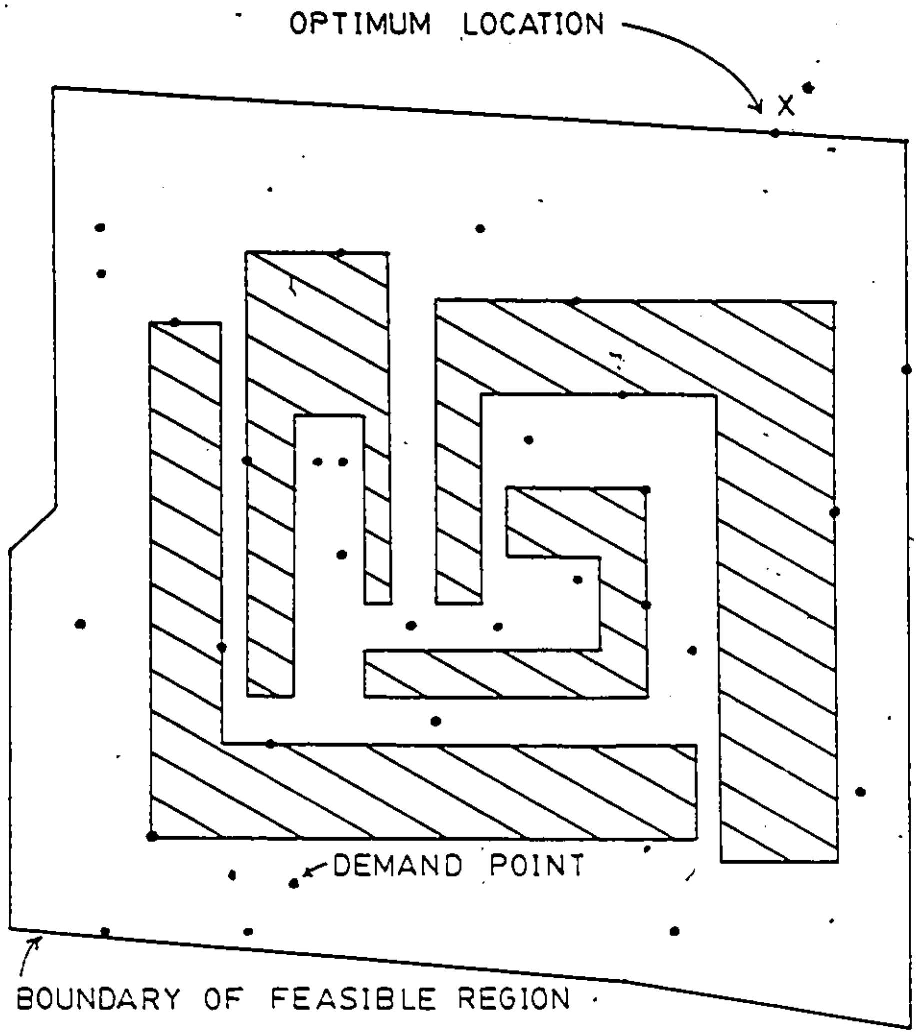

Figure 3.22 Large maximin example.

not necessary for a minimax location problem to be non-trivial because the process of minimizing the distance to the farthest demand point in effect "pulls" the facility toward the demand points rather than "pushes" it away as in maximin location problem. Accordingly, the feasible region for minimax location problems may consist of the entire plane. If this is the case, the problem universe should be picked so that it encloses the demand points and the barrier vertices.

In the maximin algorithm, we begin with the smallest possible objective function value, $z_0 = 0$. By picking a series of feasible points, we generate a series of increasing objective function values. In the minimax algorithm, we must begin with a sufficiently large objective function value and generate a series of decreasing objective function values. Since there is no upper bound on the objective function value, the choice of the initial objective function value is more difficult than in the maximin case. For our initial z_0, we use the largest weighted distance from any demand point to any accessible point in the problem universe since this value is relatively simple to calculate from the corner data files.

3.6.2 <u>Minimax Location Algorithm</u>

To obtain an algorithm for minimax problems, make the following changes to the maximin algorithm given in subsection 3.5.1.

For Phase I of the algorithm, we only need to include in the description of Step 1 the sentence "If the feasible region consists of the entire plane, then choose the limits of the problem universe so that it encloses all demand points and barrier vertices."

For Phase II of the algorithm, several changes are required. Replace $z_k = \min \{ f_I(X_k); I=1,\ldots,N_P \}$ by $z_k = \max \{ f_I(X_k); I=1,\ldots,N_P \}$ in Step 2 of Phase II. Replace $z^* = \min \{ f_I(X^*); I=1,\ldots,N_P \}$ by $z^* = \max \{ f_I(X^*); I=1,\ldots,N_P \}$ in Step 4 of Phase II.

We also need to calculate an initial value for z_0 in Step 1 of Phase II. Replace the instruction setting $z_0 = 0$ by the following instructions: For I = $1,\ldots,N_P$, calculate the maximum distance from demand point P(I) to any point in the problem universe by calculating CMAX for each open cell and taking the maximum value of CMAX obtained. Call this maximum distance UMAX(I) for demand I. Set $z_0 = \max \{ w_I * \text{UMAX}(I); I = 1,\ldots,N_P \}$.

3.7 Non-Graphical Unweighted Maximin Location

Although this work is primarily concerned with graphical approaches to location problems, we note that simple extensions to the theory and methods developed in this chapter for the graphical algorithms can yield a purely numerical algorithm for unweighted maximin location problems when the feasible region is defined by one or more polygons. Since this type of location problem with rectilinear distances and barriers is of considerable practical interest and is unsolved, we will briefly sketch a numerical algorithm for the special case of maximin location with unweighted distances.

3.7.1 Unweighted Maximin Objective Function Properties

The key to a practical numerical algorithm for the unweighted maximin problem is an observation about the properties of unweighted distances in a cell. Consider a cell as defined in section 3.2 with all demand points present when the division into cells was made. The distance from a given demand point P(I) to any point X in a cell is given by min{ d(P(I),C(j)) + d(C(j),X); j=1,...,4} where d(P(I),C(j)) is the distance from the demand point P(I) to the j'th corner of the cell C(j) and d(C(j),X) is the rectilinear distance from corner C(j) to X. In maximin problems, the objective function to be maximized is the minimum distance over the demand points P(I), I=1,...,N_p. Since this is an unweighted problem and the d(C(j),X) do not depend upon I, the minimization over the corners and the minimization over the demand points can be interchanged.

Let d(*,C(j)) = min{ d(P(I),C(j)); I=1,...,N_p } for each corner C(j). After interchanging the minimization over demand points and minimization over corners, the objective function at any point X in the cell is given by min{ d(*,C(j)) + d(C(j),X); j=1,...,4 }. Since this form of the objective function is identical to the form of the distance functions inside a cell, those results which we obtained for the level curves and maximum value of the distance in a cell can also be applied to the objective function inside a cell. In particular, equation 3.3.13 can be used to calculate the maximum objective function in a cell which we will call OCMAX. If H and W represent the height and width of the cell, then

OCMAX = min{ [d(*,C(1)) + d(*,C(3)) + H + W]/2,
[d(*,C(2)) + d(*,C(4)) + H + W]/2 }.

The line segment on which the objective function has the value OCMAX can be obtained by substituting OCMAX into the equations of the level curves given in subsection 3.3.1 after replacing z(j) by d(*,C(j)) for j=1,...,4. This line segment will always lie entirely in the cell or on the cell boundary. This can be proved by substituting the definitions of the d(*,C(j))s and the preceding expression for OCMAX into the equations (3.3.5 to 3.3.12) for the corners of the level curves and then using the triangle inequalities for distances which were discussed in subsection 3.3.2.

3.7.2 Numerical Unweighted Maximin Location Algorithm

A practical numerical algorithm for unweighted maximin problems can be obtained by replacing Phase II, the interactive graphical phase, of the algorithm presented in subsection 3.5.1 by the following numerical procedure. The procedure first determines the maximum feasible value of the objective function in the cells through which the boundary of the feasible region passes. It then calculates the maximum objective function value OCMAX in each of those cells which lie inside the feasible region and which do not contain any part of the boundary.

First divide the problem universe into cells with all of the demand points present in the problem when the division is made. Let C(j,k) represent a general cell corner resulting from this division. For each of the cell corners in the problem, calculate $d(*,C(j,k)) = \min \{ d(P(I),C(j,k)); I=1,...,N_P \}$ using the corner data prepared by Phase I of the algorithm. The distances to cell corners which do not appear in a particular reduced problem can be calculated in the same way as the distance to a general point so that no changes to the Phase I program are necessary. For each cell, determine if it is open or closed to travel by testing the point at the center of the cell in order to determine if it is an accessible point. Using the Phase I corner data associated with any one of the demand points, this can be done by finding the reduced problem cell in which the center point lies. If that cell in the reduced problem is open then the center point is accessible.

For each open cell through which the boundary of the feasible region passes, determine the maximum feasible objective function value in the cell. The maximum feasible objective function value in a cell either occurs on the boundary of the feasible region or is equal to OCMAX for the cell. Calculate OCMAX and the line segment on which the objective function value of OCMAX occurs using equations of the level curves. Determine if any point on the line segment is feasible. If the entire line segment is infeasible, then search that portion of the boundary which lies inside the cell in order to find the maximum feasible value in the cell.

For each of the remaining open cells, determine if they lie entirely inside or entirely outside the feasible region by checking one interior point to determine if that point and hence, the entire cell is feasible. For each entirely feasible cell found, calculate OCMAX and the line segment on which OCMAX occurs.

Choose the maximum feasible objective function value found in the search over all cells as the optimum value and choose as the optimum location that point or line segment on which the optimum value was found.

CHAPTER 4

NOXIOUS FACILITY LOCATION WITH AREA DEMANDS

4.1 Introduction

This chapter addresses the problem of locating a single noxious facility so that the minimum of the distances to several polygonal areas is maximized. Euclidean, rectilinear, and general l_p distances are treated, but there are no barriers to travel. We discuss extensions of the method to problems involving the minimization of the maximum cost, where the costs are functions of distance and direction, or distance alone.

4.1.1 Problem Description

Although a facility may provide important or essential services, the facility itself may have undesirable characteristics which make it necessary to locate the facility as far as possible from certain areas. The facility may be unaesthetic and unpleasant for those living nearby; waste-treatment plants and garbage dumps certainly fall into this category. The facility's normal operation may pose a health hazard to the surrounding population. It is suspected but has not been proved that this is the case for coal-fired generating stations (Cohen 1983). The facility may even threaten nearby residents with catastrophic accidents such as have occurred at oil-fired generating stations and natural gas facilities. Machines, equipment, or electrical components which produce excessive emissions of noise, heat, or electrical interference can pose similar problems in plant layout and in the design of electronic equipment.

We will use the term "noxious facility" to describe such a facility. In the literature, this type of facility is frequently called an "obnoxious facility." Less often, it is termed an "undesirable" or "controversial" facility. We note that in general usage "obnoxious" is almost exclusively applied to persons and that it produces distracting connotations when it is applied to inanimate objects.

The noxious facility is to be located as far as possible from several areas which we will call area demands or vulnerable areas. We assume that these area demands are general polygons or can be approximated by general polygons. These polygons may be non-convex and may overlap. Without loss of generality, we assume that the polygonal areas have no interior holes. A polygonal area with holes can always be represented by two adjacent polygons without holes.

These area demands may represent sensitive components in a layout problem or populated areas which are to be avoided when locating a noxious facility. If the entire feasible region for facility placement is populated, then the polygonal areas can still be used to represent high density areas, such as cities, which should be avoided. Although our discussion is in terms of locating a noxious facility in relation to several vulnerable regions, we could also interpret the model in terms of locating a vulnerable facility as far as possible from several noxious regions.

We assume that the noxious facility is constrained to lie in a bounded feasible region. Engineering, geographical, political, or legal considerations may constrain the facility to lie inside a particular region in practical problems. Maximum service distances may also place constraints on the location of the noxious facility because it provides services either to the vulnerable areas or to other service recipients with fixed positions relative to the vulnerable areas. In the absence of such a bounded feasible region for facility placement, the trivial and unrealistic solution to a maximin location problem is to place the facility "at infinity."

Weighted Euclidean, rectilinear, and general l_p distance norms are allowed in our formulation. Also allowed are mixed norm problems in which different area demands have different distance norms associated with them. The weights associated with each demand can be used to reflect the relative insensitivity to damage of the area demands.

The undesirable effects of a noxious facility often can be assumed to decrease with distance, even though the magnitude of its effects and the rate of decrease with distance are not known with reasonable accuracy. By using the maximin criterion, maximizing the minimum distance from the vulnerable areas to the noxious facility, we will at least be assured of minimizing the largest of these adverse effects.

4.1.2 Formulation

Let $A(i)$, $i=1,...N$ be a set of N polygonal areas (Figure 4.1). Let $V(m,i)$, $m=1,...,M_i$ be the vertices of polygonal area $A(i)$ in clockwise order. Each polygonal area is defined to include its boundary. The areas may be non-convex and may overlap one another. Let F be a closed, bounded feasible region for placement of the facility. The region F may be non-convex or disjoint.

The maximin single facility location problem with l_p distances and polygonal area demands can be formulated as

$$\underset{X \text{ in } F}{\text{maximize}} \; \{ \underset{i=1,...,N}{\text{minimum}} \; f_i(X) \}$$

where

$$f_i(X) = w_i \underset{Q \text{ in } A(i)}{\text{minimum}} \{ d(p(i);X,Q) \},$$

$X = (x,y)$ = location of new facility,

$Q = (a,b)$ = location of a general point,

w_i = positive weight associated with area $A(i)$,

$p(i)$ = p-value for points in $A(i)$,

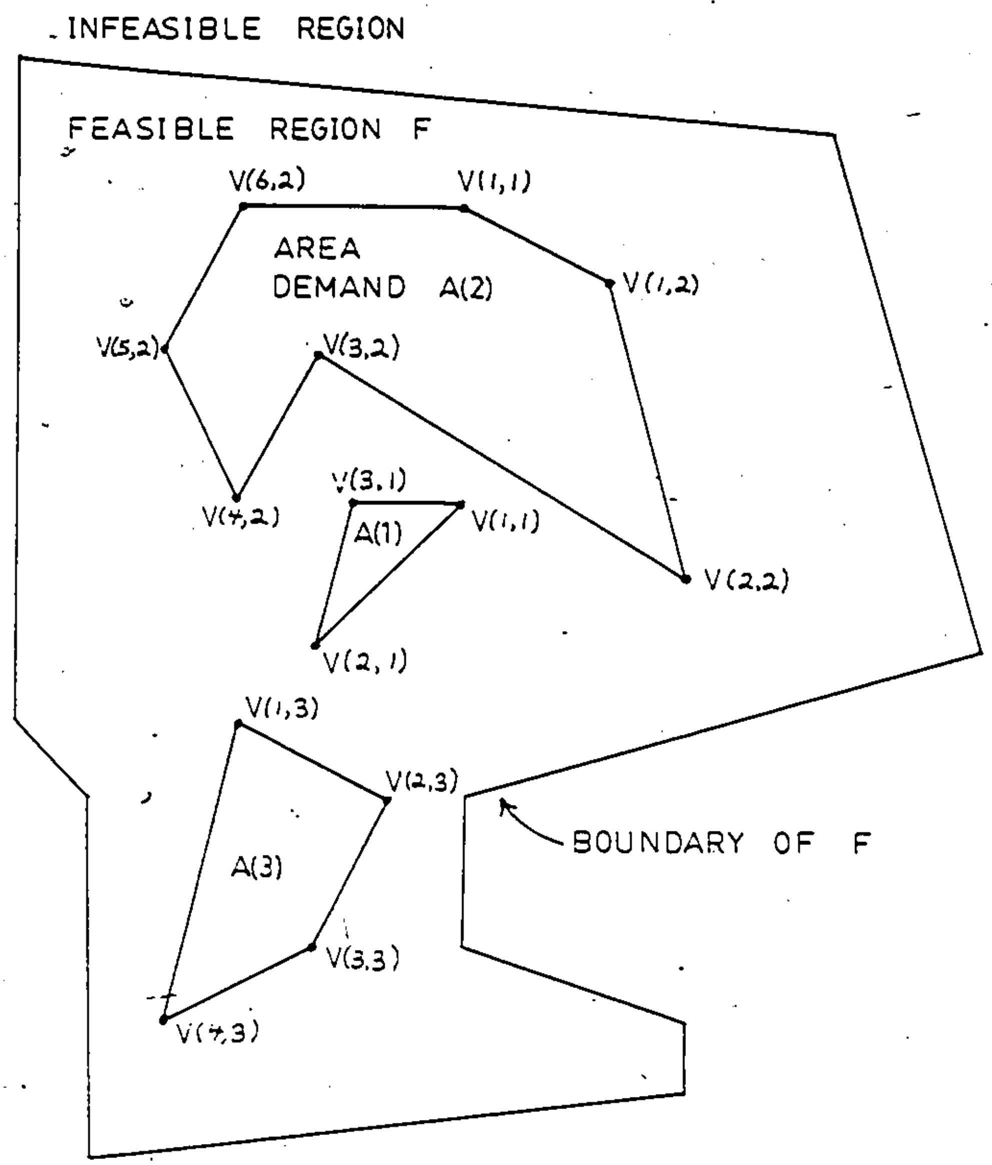

Figure 4.1. Feasible region and area demands.

and $d(p(i);X,Q) = [|x-a|^{p(i)} + |y-b|^{p(i)}]^{1/p(i)}$

$= l_{p(i)}$ distance from X to Q.

If the set of area demands covers the entire feasible region F, then the problem has a trivial solution with an objective function value of zero and all feasible points are optimum locations. If there exist one or more feasible points outside the set of area demands, then the formulation can be simplified since only points on the boundaries of the area demands need to be considered.

Lemma 4.1:

Let X be a point which lies outside the area demand A(i). If Q is a point in A(i) which minimizes d(p(i);X,Q), then Q lies on the boundary of A(i).

Proof:

The point Q must lie either in the interior or on the boundary of A(i). Assume Q lies in the interior of A(i). By the definition of an interior point, there exists an $e > 0$ such that if $d(p(i);Y,Q) \leq e$, then Y lies in A(i). Consider a point Z on the line segment from X to Q such that $0 < d(p(i);Z,Q) \leq e$. See Figure 4.2. It is well-known in the familiar Euclidean case, and easily proved for the general l_p distance case, that d(p(i);X,Q) = d(p(i);X,Z) + d(p(i);Z,Q) whenever Z lies on the line segment XQ between X and Q. Thus, Z lies in A(i) and d(p(i);X,Z) = d(p(i);X,Q) - d(p(i);Z,Q) < d(p(i);X,Q). This is a contradiction with our assumption that Q minimizes d(p(i);X,Q). Therefore, Q does

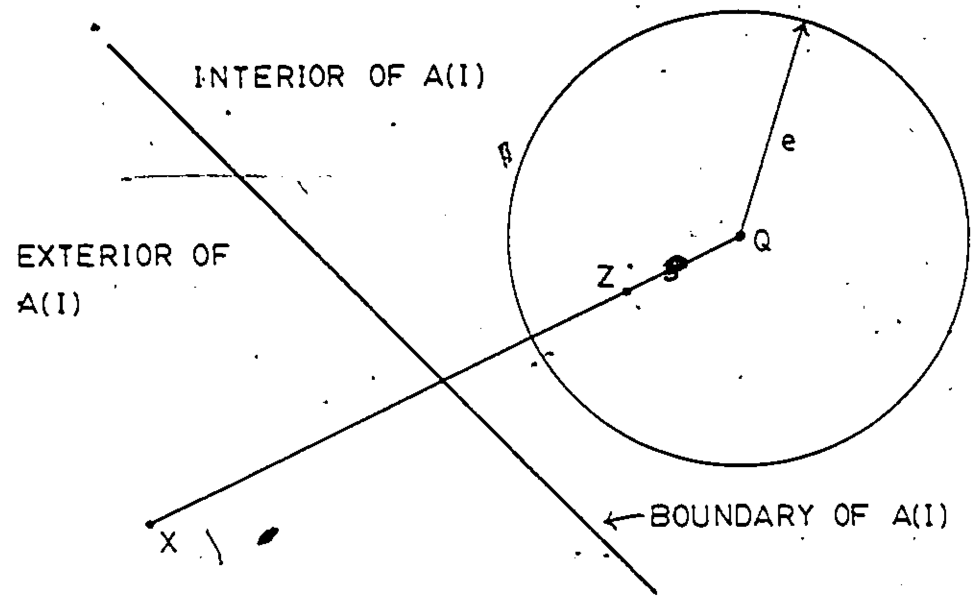

Figure 4.2. Diagram for lemma.

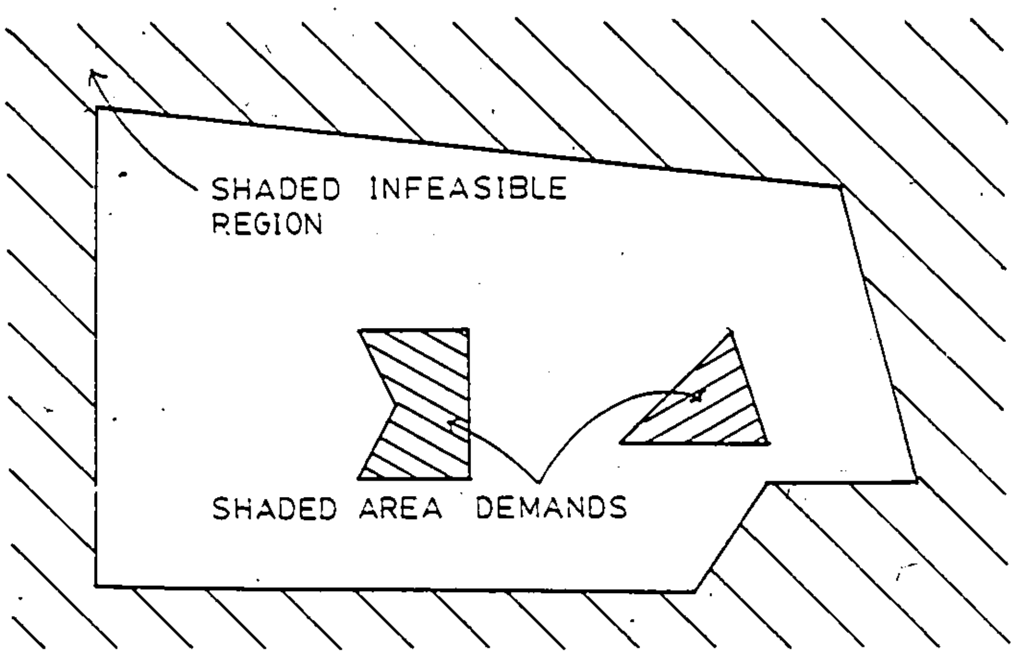

Figure 4.3. Maximin algorithm - Initialization stage.

not lie in the interior of A(i) and must lie on the boundary. □

Let L(m,i) be the m'th side of polygon A(i) with end points V(m,i) and V(m+1,i) for m = 1,...,M_i-1. Let L(M_i,i) be defined as the M_i'th side with end points V(M_i,i) and V(1,i).

By the Lemma 4.1, f_i(X) can be simplified for X not in A(i) since

$$f_i(X) = w_i \underset{Q \text{ in } A(i)}{\text{minimum}} \{ d(p(i);X,Q) \}$$

$$= w_i \underset{m=1,\ldots,M_i}{\text{minimum}} \{ d(p(i);X,L(m,i)) \}$$

where

$$d(p(i);X,L(m,i)) = \underset{Q \text{ in } L(m,i)}{\text{minimum}} \{ d(p(i);X,Q) \}$$

$$= \text{minimum } l_p \text{ distance from X to line segment } L(m,i).$$

In our preceding definition of L(m,i), we were forced to include a special definition for the M_i'th side L(M_i,i) because the general definition of L(m,i) as the line segment with end points V(m,i) and V(m+1,i) did fit the line segment from V(M_i,i) to V(1,i). To reflect the cyclic ordering of the M_i vertices of the polygon A(i), we define any reference to the vertex or side M_i+1 as being a reference to vertex or side number 1. For example, any reference to V(M_i+1,i) is to be interpreted as V(1,i).

4.2 Graphical Approach to Maximin Problem

The following algorithm for the maximin noxious facility problem is an extension of the algorithm given by Hansen, Peeters, and Thisse (1981) for noxious facility location in relation to a finite number of point demands. We have extended it to the case of polygonal area demands and added a scale expansion step which allows increased accuracy. Hansen et al. (1981) presented their graphical method in the context of minimizing the maximum cost where the costs are monotonically decreasing functions of distance. To avoid unnecessarily complicating the discussion of graphical methods for area demand problems, we will first deal with maximin distance problems and postpone until section 4.3 the treatment of minimax cost problems, even though the modifications required are straightforward.

Subsection 4.2.1 presents, in general terms, the algorithm for maximin location problems with polygonal area demands. The level curve (contour line) of a function for a value z is the set of points for which the function has the value z. In extending the graphical approach to area demands, we will be particularly concerned with determining and shading those regions of the plane which lie between the level curves of the function $f_i(X)$ for given function values (Figure 4.6). Subsections 4.2.2, 4.2.4, and 4.2.6 discuss this in detail for Euclidean, rectilinear, and general l_p distances. Calculation of the minimum distance to an area demand for each of the three distance metrics is covered in subsections 4.2.3, 4.2.5, and 4.2.7.

4.2.1 Algorithm for Maximin Location

Although the algorithm is designed to be implemented on a bit-mapped computer graphic system, if only modest accuracy is required, small problems can be solved by hand using graph paper and drawing instruments.

Step 1. Initialization

Choose the region of the plane which will be represented on the screen so that it encloses the entire feasible region for facility placement. Shade the infeasible region. Shade the interiors and boundaries of all area demands A(i), i=1,...,N. See Figure 4.3.

If no unshaded point remains on the screen, then only a trivial solution is possible. Stop. Optimum objective function value is zero and pick any feasible point as an optimum location.

Otherwise, set $k = 1$ and $z_0 = 0$.

Step 2. Improve Value of Objective Function

Choose any unshaded point X_k, preferably one which lies near the center of an unshaded region (Figure 4.4). Calculate

$$z_k = \min \{ f_i(X_k);\ i=1,\ldots,N\}.$$

Step 3. Eliminate Dominated Regions

Shade the area between the level curves of $f_i(X)$

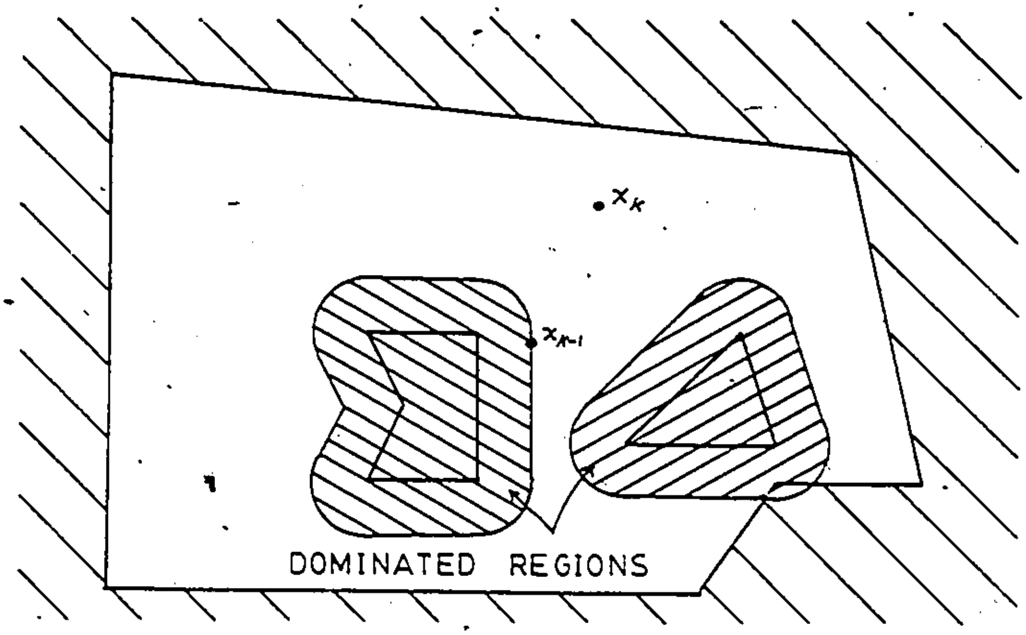

Figure 4.4. Maximin algorithm - Intermediate stage.

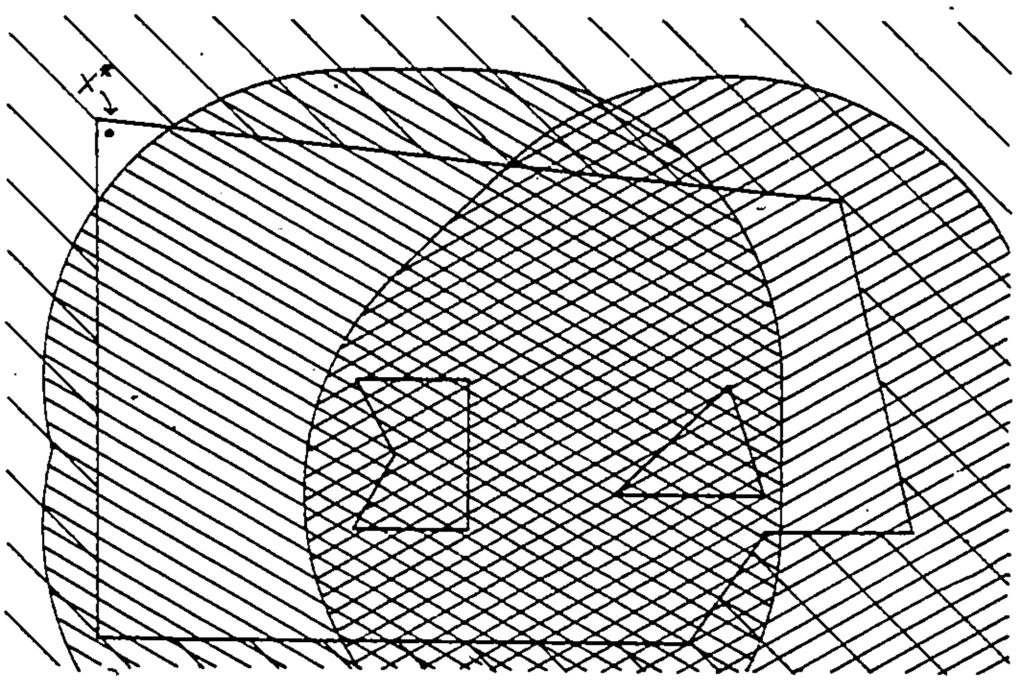

Figure 4.5. Maximin algorithm - Final stage.

corresponding to the values z_{k-1} and z_k for each i=1,...,N.

Set k = k + 1.

If only a few points (pixels) remain unshaded, then go to Step 4. Otherwise, go to Step 2.

Step 4. Accept or Expand Scale

If the size of the remaining unshaded area is sufficiently small so that the uncertainty in the location of the optimum is acceptable, then choose a point in the remaining unshaded area as a near-optimal location X^* (Figure 4.5). Calculate $z^* = \min (f_i(X^*); i=1,...,N)$. Stop.

Otherwise, choose a small rectangle enclosing the unshaded region and expand the scale so that the rectangle fills the screen. Shade the infeasible region and the area demands A(i), i=1,...,N. For all i=1,...,N, shade the area between the level curves of $f_i(X)$ corresponding to the values 0 and z_{k-1}. Go to Step 2.

4.2.2 Euclidean Level Sets of a Polygon

We require a procedure for determining and shading that region of the plane (Figure 4.6) which lies between two level curves of the function $f_i(X)$ when the distances are Euclidean. It would be equally effective to determine and shade the set of points for which $f_i(X) \leq z_k$, but it would be less efficient since the algorithm would repeatedly shade areas which have already been shaded during earlier

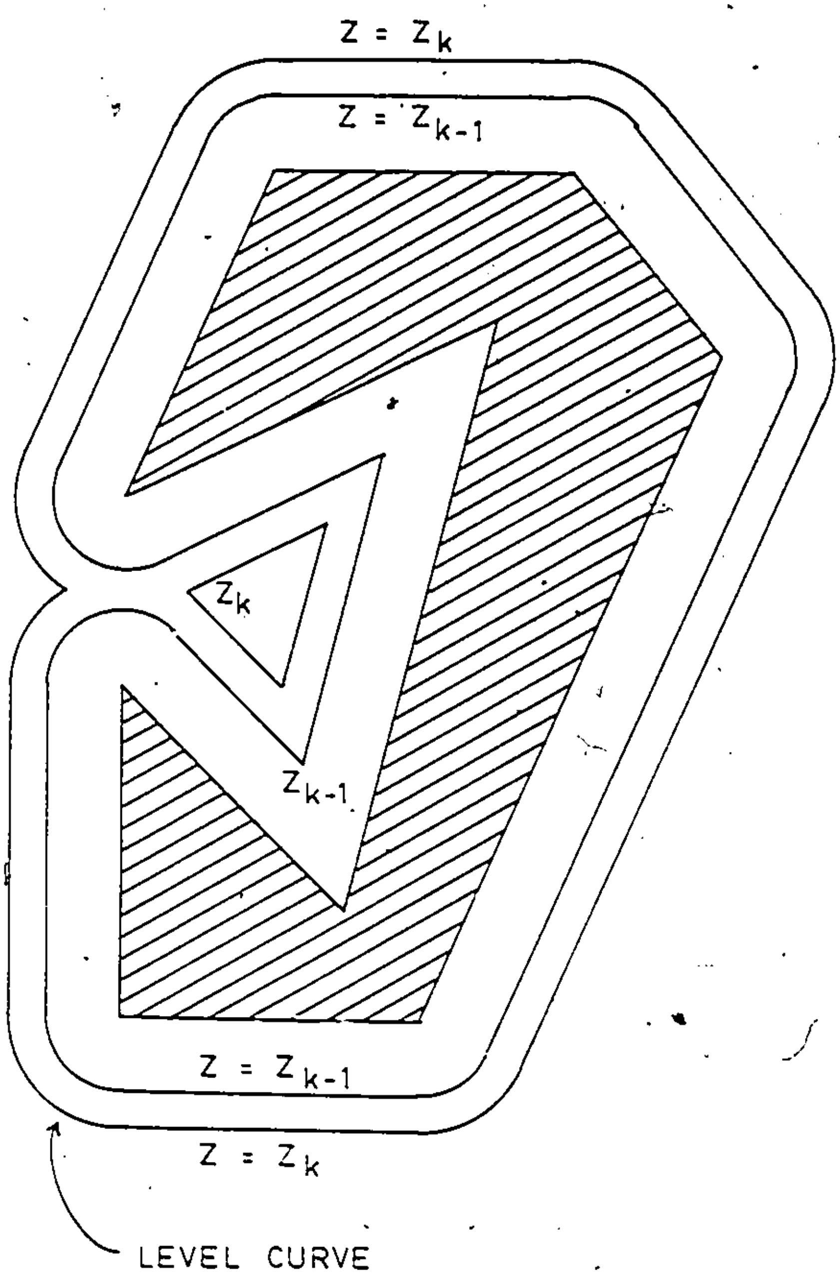

Figure 4.6. Euclidean level curves of area demand.

iterations. The set of points for which $f_i(X) \leq z_k$ is called the level set of the function $f_i(X)$ for value z_k (Bazaraa and Shetty, p. 81). The level curve (contour line) of $f_i(X)$ for a value z_k is the set of points X for which $f_i(X) = z_k$. Since $f_i(X)$ is a continuous function, the boundary of the level set of $f_i(X)$ for the value z_k is the level curve of $f_i(X)$ for the value z_k.

Consider a given area demand A(i) with p(i) = 2 indicating the Euclidean distance case and with weight w_i. The level set of the function

$$f_i(X) = w_i \underset{Q \text{ in } A(i)}{\text{minimum}} \{ d(p(i)=2;X,Q) \}$$

for a function value of z_k is the set of all points which are no more than z_k/w_i Euclidean distance away from at least one point in the area demand A(i). The level set of the function $w_i d(p(i)=2;X,Q)$ associated with each point Q in A(i) is a disc with center Q and radius z_k/w_i. The level set of $f_i(X)$ is the union of the level sets associated with each point in A(i). By Lemma 4.1, the points on the boundary of the area demand A(i) determine the boundary of the level set and the level curve of $f_i(X)$. As shown in Figure 4.7, the level curve of the function $f_i(X)$ for a value z_k forms an envelope of the family of level curves of the functions $w_i d(p(i);X,Q)$ associated with each point Q on the boundary of polygonal area A(i).

Since we are only concerned with a single area

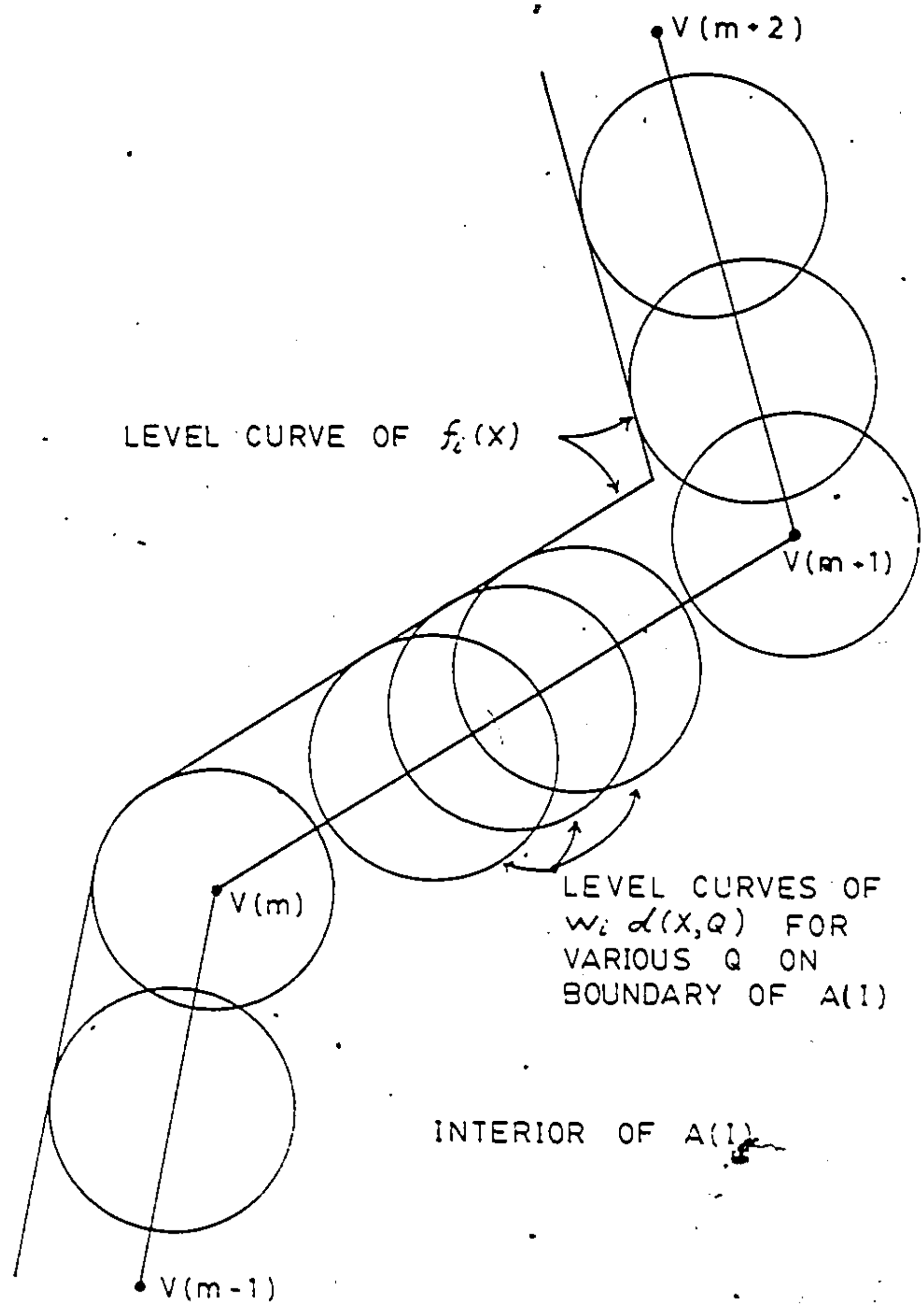

Figure 4.7. Euclidean level curve is envelope of circles.

demand, we can simplify our notation for the remainder of this chapter by suppressing the index which indicates the area demand A(i) when we refer to its vertices and sides. Accordingly, V(m) represents the m'th vertex of the area demand under consideration and (x(m),y(m)) represents the coordinates of V(m). Similarly, L(m) is the line segment from V(m) to V(m+1) which forms the m'th side of the polygonal area demand A(i).

Let the vector $S(m) = (s_x(m), s_y(m)) = V(m+1) - V(m) = (x(m+1)-x(m), y(m+1)-y(m))$. Recall that $S(M_i) = V(M_i+1) - V(M_i) = V(1) - V(M_i)$ by our definition of M_i+1 in subsection 4.1.2. The vector S(m) associated with the m'th side is parallel to the m'th side and is directed from V(m) to V(m+1).

Consider side L(m) of the polygon (Figure 4.8) between vertices V(m) and V(m+1). It can be seen that the area between the level curves for z_{k-1} and z_k can be completely shaded by shading one rectangular area for each side of the polygon and one sector of an annulus for each vertex with an exterior angle greater than 180°. This may shade points which are closer to the area demand than z_{k-1}/w_i when the polygon is non-convex. Since these points are in the dominated region which has already been shaded, the solution of the maximin problem is not affected.

The rectangular region to be shaded for the m'th side is $R_1R_2R_3R_4$. The corners are given by

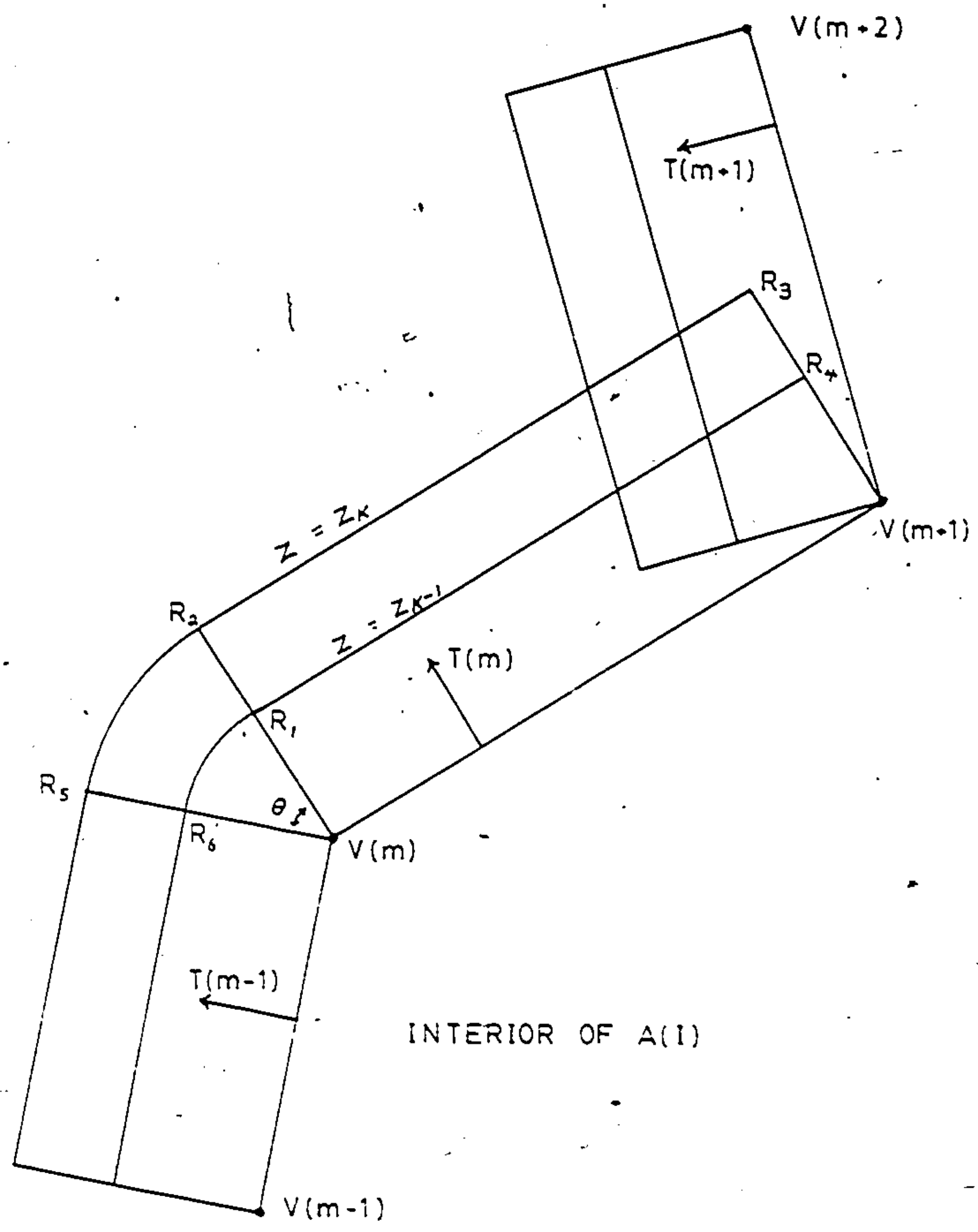

Figure 4.8. Rectangle and sector of annulus.

$$R_1 = V(m) + (z_{k-1}/w_i)\ T(m),$$
$$R_2 = V(m) + (z_k/w_i)\ T(m),$$
$$R_3 = V(m+1) + (z_k/w_i)\ T(m),$$
and $$R_4 = V(m+1) + (z_{k-1}/w_i)\ T(m)$$

where $T(m)$ is a unit vector perpendicular to the m'th side and directed toward the exterior of the polygon. If the vertices of the polygon have been labeled in clockwise order, then

$$T(m) = (\ -s_y(m)/|S(m)|,\ s_x(m)/|S(m)|\ )$$

where $|S(m)|$ is the length of the vector $S(m)$.

The exterior angle at vertex $V(m) = 180^\circ + \theta_{m,m-1}$ where $\theta_{m,m-1}$ is the angle between the vectors $S(m)$ and $S(m-1)$ measured from $S(m)$ to $S(m-1)$. The magnitude of the vector cross product of $S(m)$ and $S(m-1)$ is

$$|S(m)\ X\ S(m-1)| = |S(m)|\ |S(m-1)|\ \sin\theta_{m,m-1}.$$

By rearrangement and substitution, we obtain $\sin\theta_{m,m-1} = [s_x(m)s_y(m-1) - s_x(m-1)s_y(m)]\ /\ [|S(m)|\ |S(m-1)|]$. Since $\sin\theta_{m,m-1} > 0$ implies that $0^\circ < \theta_{m,m-1} < 180^\circ$, the exterior angle at $V(m)$ is greater that 180° if

$$[s_x(m)s_y(m-1) - s_x(m-1)s_y(m)] > 0.$$

In this case, a portion $R_6R_5R_2R_1$ of an annulus between radii z_{k-1}/w_i and z_k/w_i with center at $V(m)$ must be shaded. If the exterior angle is greater than 180°, then the angle θ (Figure 4.8) through which the annulus must be shaded is equal to the angle $\theta_{m,m-1}$ between the vectors $S(m)$ and $S(m-1)$. The dot product of the vectors $S(m)$ and $S(m-1)$ is

$S(m) \cdot S(m-1) = |S(m)|\ |S(m-1)| \cos \theta_{m,m-1}$.

After rearranging and taking inverse cosines, we obtain

$$\theta = \theta_{m,m-1}$$
$$= \arc\cos \{ [s_x(m)s_x(m-1) + s_y(m)s_y(m-1)] / [|S(m)|\ |S(m-1)|] \}.$$

For Euclidean distances, we have shown that the regions to be shaded are readily calculated and hence, the algorithm can be implemented if the computer graphics system provides the necessary primitive operations. To shade the polygonal area demands and the rectangles associated with each side of the polygons requires a routine which shades (fills) a polygon given its vertices. This type of polygon filling graphic primitive appears to be quite common. We expect very few graphics systems will provide a routine to fill the interior of a sector of an annulus. However, one can use a routine which fills a sector of a circle or fills an entire circle since the additional points shaded are in the dominated region and will not affect the solution of the maximin problem.

4.2.3 Euclidean Distance to a Polygon

In order to evaluate the objective function at a given X outside the area demands, we need to find the minimum distance from X to each of the A(i). By Lemma 4.1, we only need to consider points on the boundaries of the area demands. Since the A(i) are polygonal areas, this

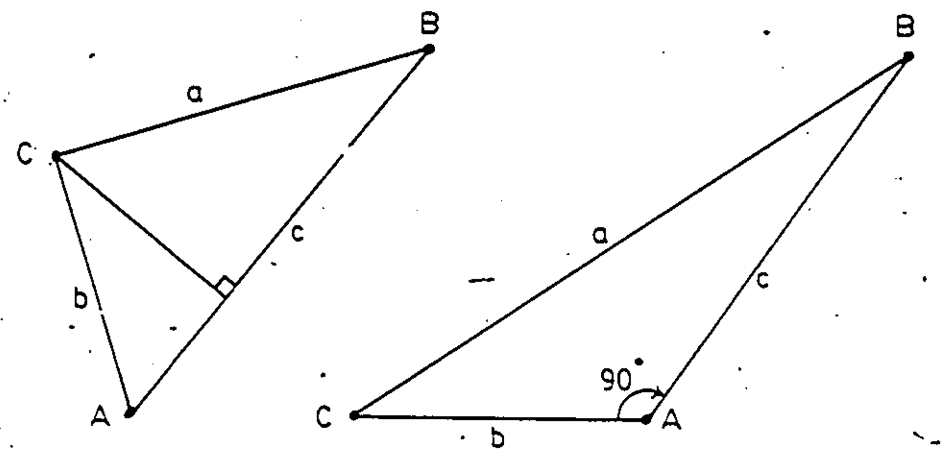

Figure 4.9. Euclidean distance to line segment.

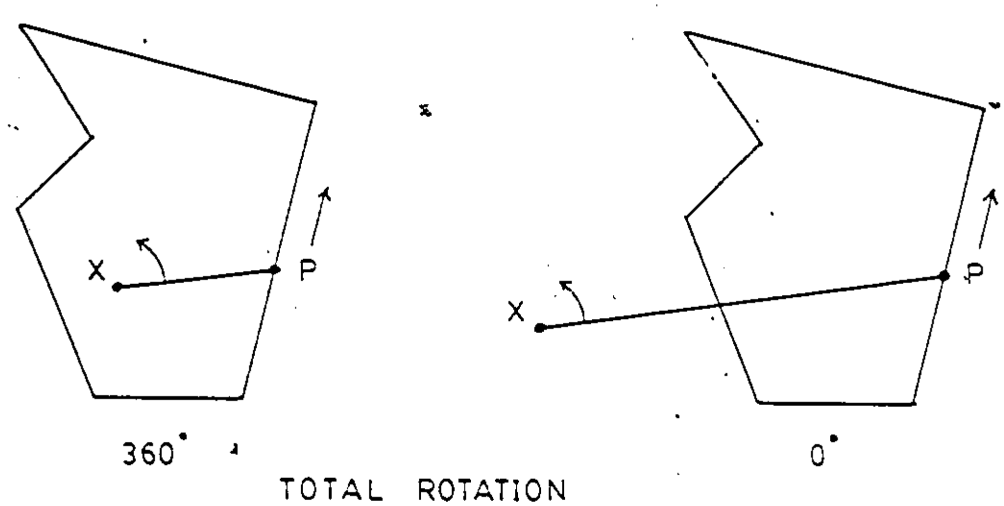

Figure 4.10. Test for exterior points.

reduces the problem to finding the minimum distance to a number of line segments.

Consider the problem of finding the minimum distance from point C to the line segment with end points A and B (Figure 4.9). If angle A or angle B of triangle ABC is greater than 90° ($\cos A$ or $\cos B < 0$), then the minimum distance is the distance to the nearest end point. If both angles A and B are equal or less than 90° ($\cos A$ and $\cos B \geq 0$), then the minimum distance is the perpendicular distance from C to the line passing through A and B.

From the "Law of Cosines" for triangles, we can obtain the equations $\cos A = (b^2 + c^2 - a^2)/ 2bc$ and $\cos B = (a^2 + c^2 - b^2)/ 2ac$. Degenerate triangles with sides a, b, or c equal to zero are simple to treat as special cases. The minimum Euclidean distance D from point $P_1 = (x_1, y_1)$ to a line with the equation $\alpha x + \beta y + \gamma = 0$ is given by the standard formula,

$$D = |\alpha x_1 + \beta y_1 + \gamma| \; / \; [(\alpha^2 + \beta^2)^{1/2}].$$

We have shown that the minimum Euclidean distance from a point to a line segment can be easily calculated. Thus, the minimum distance to a polygonal area A(i) from any point X outside it can also be easily determined.

We assumed that the point X lies outside all area demands. In order to guard against operator errors when picking the next feasible point in Step 2 of the algorithm, it is possible to verify that X lies outside a given polygonal area by a simple test (Figure 4.10). Let X be a point on the outside of a simple, closed curve (a polygon in this case). Consider a point P on the curve which traces out the curve and returns to its starting point. The total angle traced out by the vector from X to P as P traces the curve is zero. If the point X lies inside the closed curve the total angle will be 360°. Thus, a simple way to check whether the point X lies outside a non-convex polygon is by calculating and summing the angles subtended by the sides of the polygon at the point X.

4.2.4 Rectilinear Level Sets of a Polygon

In the rectilinear distance case, the level sets of the function $w_i d(p(i)=1;X,Q)$ are squares, usually called "diamonds," with their diagonals parallel to the x and y axes (Figure 4.11). Like the Euclidean case, the level set of the function

$$f_i(X) = w_i \min_{Q \text{ in } A(i)} \{ d(p(i)=1;X,Q) \}$$

for a function value of z_k is the union of the level sets of the functions $w_i d(p(i)=1;X,Q)$ associated with each point Q in A(i). The level curve of $f_i(X)$ for value z_k is the boundary of the level set of $f_i(X)$ and forms an envelope (Figure 4.12) about the level curves of the functions $w_i d(p(i)=1;X,Q)$ associated with each point Q on the boundary of A(i).

Consider the m'th side of a polygonal area demand

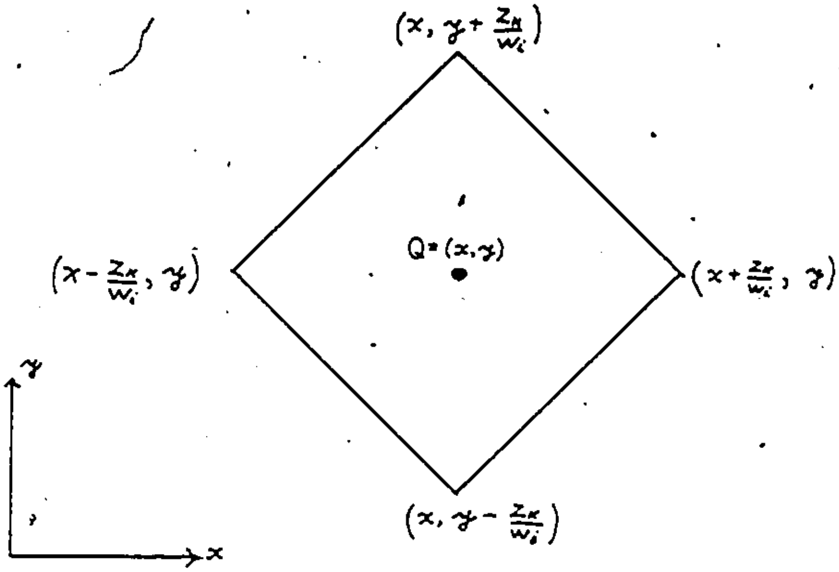

Figure 4.11. Rectilinear distance diamond.

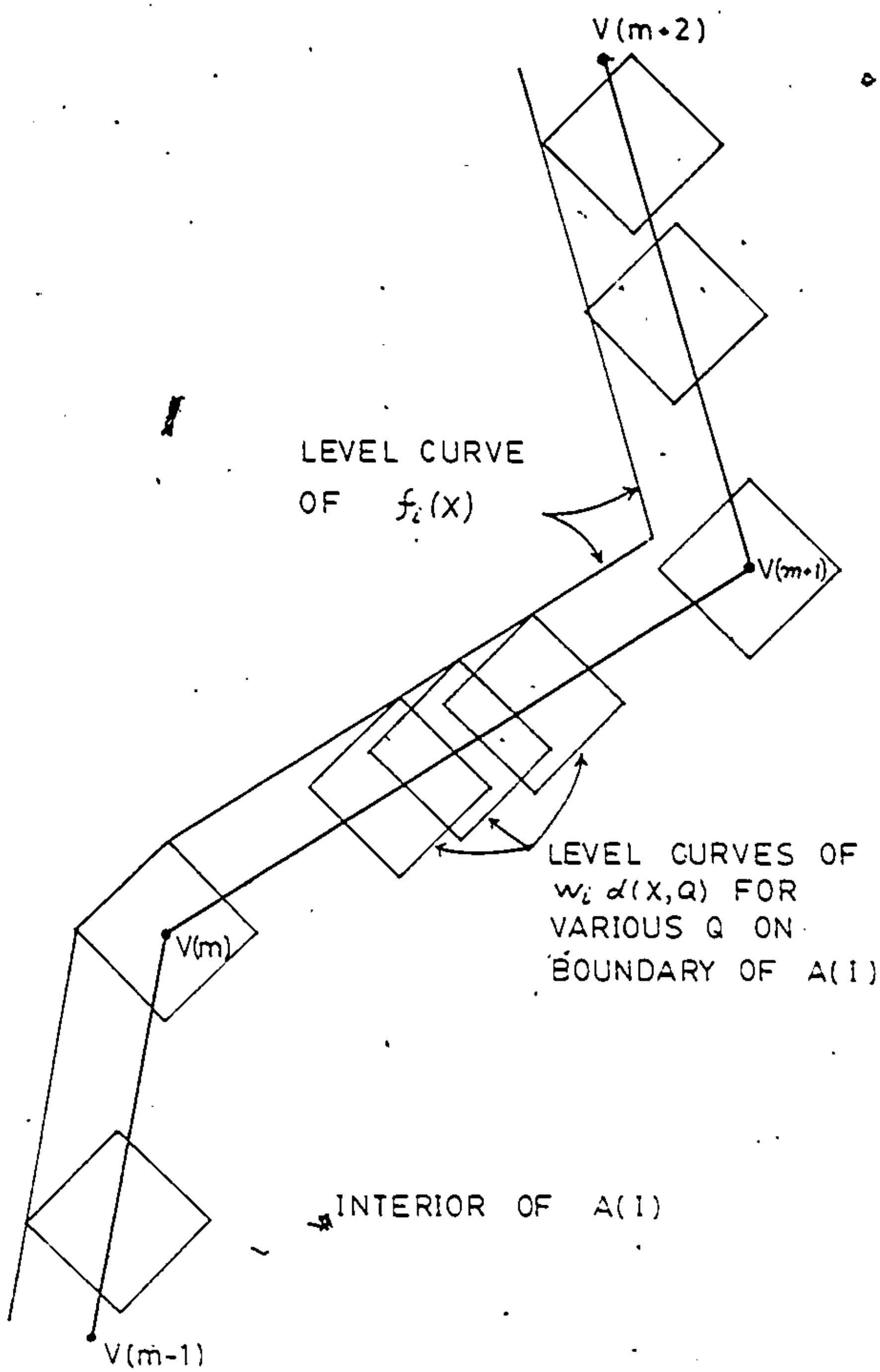

Figure 4.12. Rectilinear level curve is envelope of diamonds.

(Figure 4.13) between vertices V(m) and V(m+1). It can be seen that the area between the level curves for the values z_{k-1} and z_k can be completely shaded by shading one parallelogram for each side of the polygon and at most two trapezoids for each vertex with an exterior angle greater than 180°. Like the Euclidean case, this may shade points which are closer to the area demand than z_{k-1}/w_i when the polygon is non-convex. Since these points are in the dominated region which has already been shaded, the solution of the maximin problem is not affected.

The corners of the parallelogram $P_1P_2P_3P_4$ are given by

$$P_1 = V(m) + (z_{k-1}/w_i)\, U(m),$$
$$P_2 = V(m) + (z_k/w_i)\, U(m),$$
$$P_3 = V(m+1) + (z_k/w_i)\, U(m),$$
and
$$P_4 = V(m+1) + (z_{k-1}/w_i)\, U(m)$$

where U(m) is a unit vector pointing toward that corner of the diamond which touches or may touch the level curve. Let θ be the angle, measured counterclockwise, from the positive x direction to S(m) = V(m+1)−V(m). The unit vector U(m) is then defined as

$$\begin{aligned} U(m) &= (-1, 0) \text{ if } 45^\circ < \theta \le 135^\circ, \\ &= (0, -1) \text{ if } 135^\circ < \theta \le 225^\circ, \\ &= (1, 0) \text{ if } 225^\circ < \theta \le 315^\circ, \\ &= (0, 1) \text{ if } 315^\circ < \theta \le 360^\circ \text{ or} \\ &\qquad\quad\ \text{ if } 0^\circ \le \theta \le 45^\circ. \end{aligned}$$

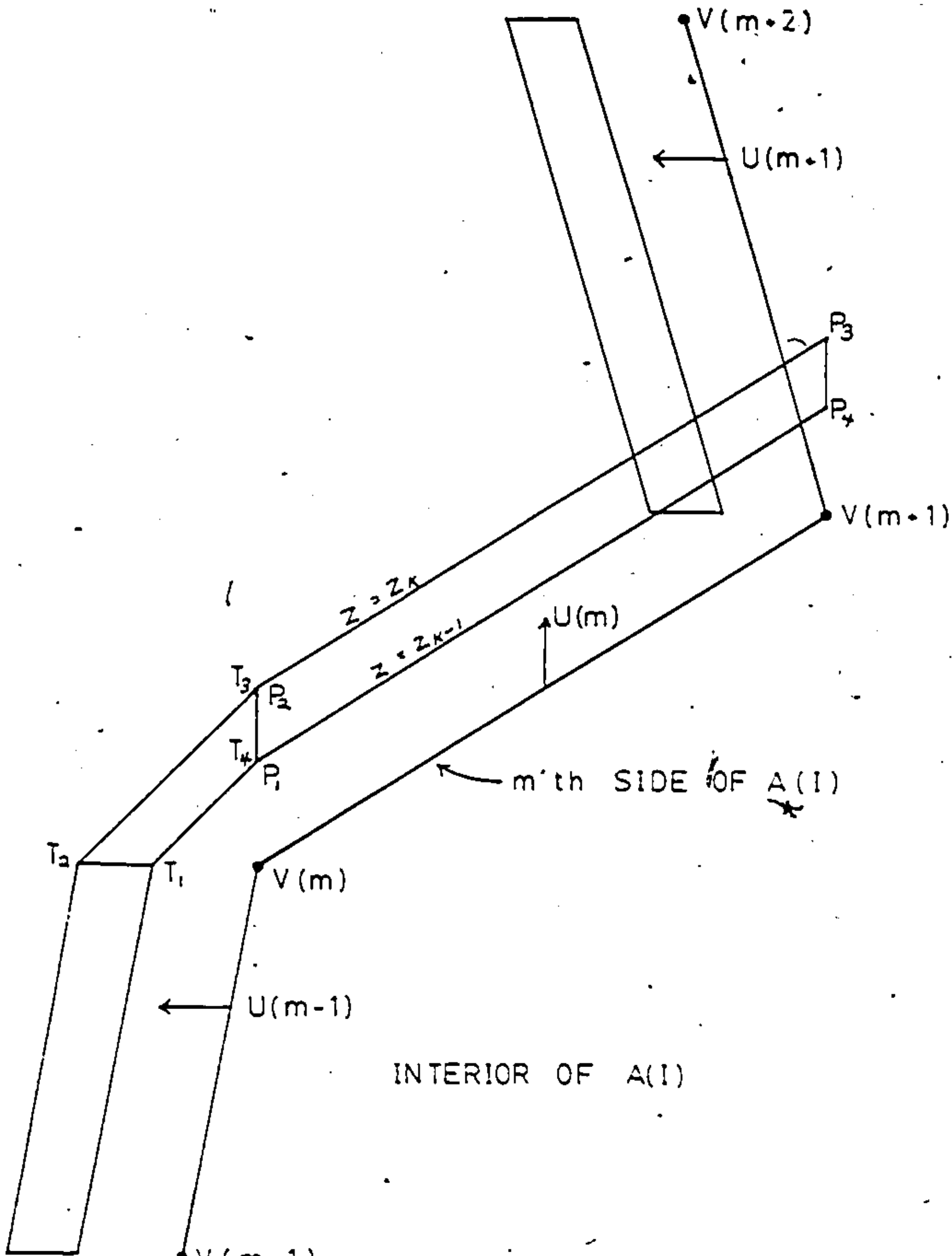

Figure 4.13. Parallelogram and trapezoid.

If the exterior angle at the vertex V(m) is greater than 180°, then it may be necessary to draw one or two trapezoids at the vertex (Figure 4.13). This depends on the orientation of the m'th and m-1'th sides of the polygon. If the unit vector U(m) associated with the m'th side is equal to the unit vector U(m-1) associated with the m-1'th side, then no trapezoids need to be drawn.

If the angle between U(m) and U(m-1) is 90°, then one trapezoid $T_1T_2T_3T_4$ must be drawn (Figure 4.13) and it has corners given by

$$T_1 = V(m) + (z_{k-1}/w_i)\ U(m-1),$$
$$T_2 = V(m) + (z_k/w_i)\ U(m-1),$$
$$T_3 = V(m) + (z_k/w_i)\ U(m),$$
and $$T_4 = V(m) + (z_{k-1}/w_i)\ U(m).$$

If the angle between U(m) and U(m-1) is 180°, then two trapezoids, $T_5T_6T_7T_8$ and $T_8T_7T_9T_{10}$, must be drawn at the vertex V(m) as shown in Figure 4.14. The formulae for T_5, T_6, T_9, and T_{10} are identical to those of T_1, T_2, T_3, and T_4, respectively. If $U(m) = (u_x(m), u_y(m))$, then the other two corners, T_7 and T_8, are given by

$$T_7 = V(m) + (z_k/w_i)\ U'$$
and $$T_8 = V(m) + (z_{k-1}/w_i)\ U'$$
where $U' = (-u_y(m), u_x(m))$ is a unit vector perpendicular to U(m) and directed toward the exterior of A(i).

For rectilinear distance cases, one only needs a polygon-filling routine since only parallelograms,

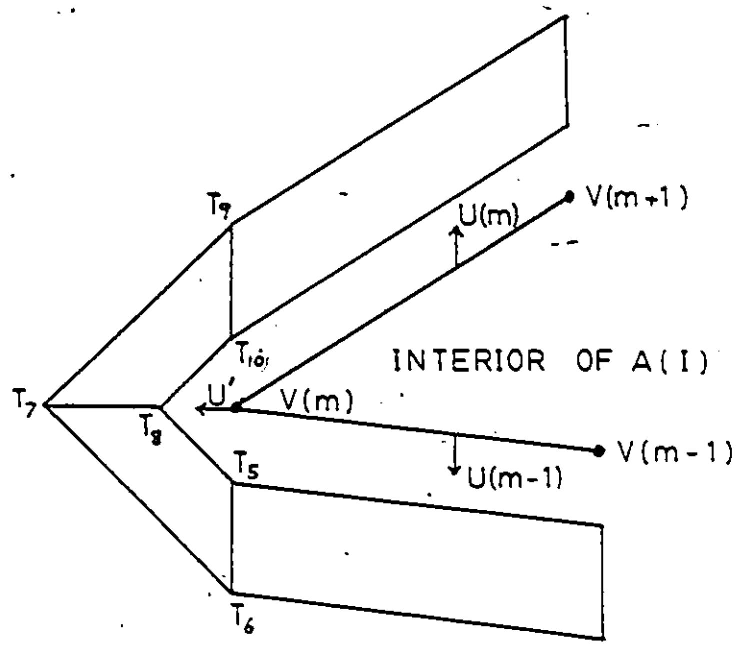

Figure 4.14. Twc trapezoid case.

trapezoids, and the polygonal area demands need to be shaded. If execution speed is not essential, one can dispense with the calculation of the corners of the trapezoids and simply draw a complete diamond at each corner which has an exterior angle greater than 180°. The additional shaded points lie in the dominated region and do not affect the solution of the maximin problem.

4.2.5 Rectilinear Distance to a Polygon

Like the Euclidean distance case, evaluating the minimum rectilinear distance from a point outside to a polygonal area reduces to finding the minimum distance to a number of line segments.

Consider a line segment AB and a point Q (Figure 4.15). If neither A nor B is the closest point on the line segment to Q, then at least one of the corners of the level curve for the minimum distance must touch the line segment between A and B as shown in Figure 4.15. Thus, to find the minimum rectilinear distance to the line AB, calculate the intersections, IX and IY, of the horizontal and vertical lines through Q parallel to the x and y axes with the line through the points A and B (Figure 4.16). Let S be the set consisting of the points A, B, and any of the intersection points (IX and IY) which lie between A and B. The minimum distance from the point Q to the line segment AB is the rectilinear distance to the closest point in the set S.

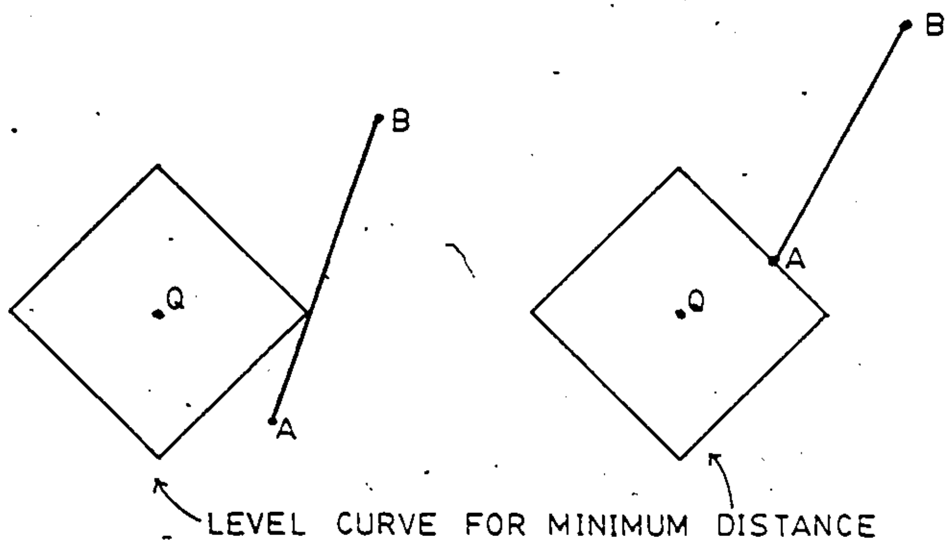

Figure 4.15. Rectilinear distance to line segment.

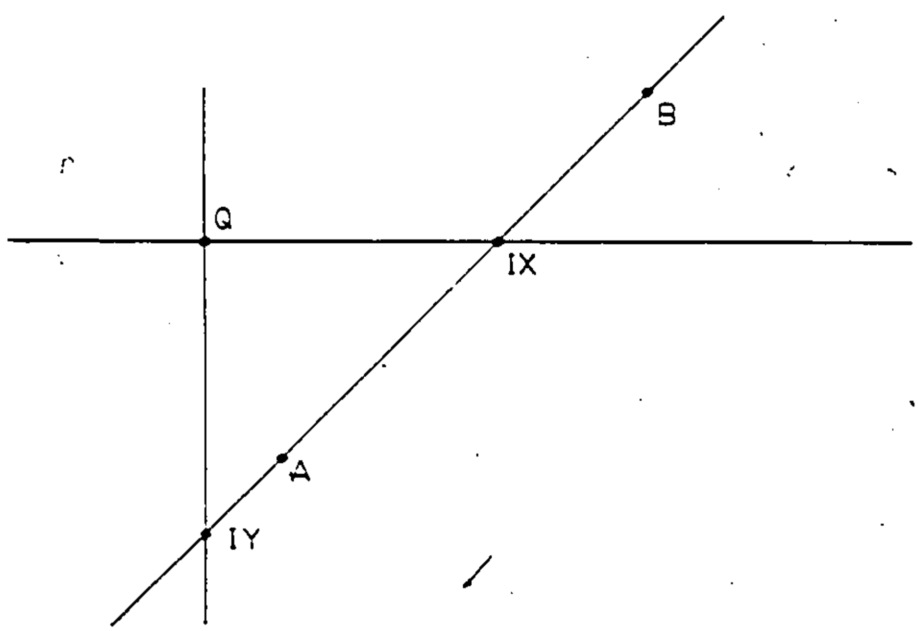

Figure 4.16. Intercepts IX and IY.

Those cases for which either IX or IY is undefined because the line segment AB is parallel to either the x or y axis must be treated separately.

4.2.6 L_p Level Sets of a Polygon

In the general l_p distance case, the level sets of the functions $w_i d(p(i);X,Q)$ are l_p circles as shown in Figure 4.17. For p-values greater than one, these l_p circles are strictly convex and the tangent exists at all points on the circumference. Also, l_p circles are symmetric about the lines x=y and x=-y as well as the x and y axes.

As in the Euclidean and rectilinear cases, the level set of the function

$$f_i(X) = w_i \min_{Q \text{ in } A(i)} \{ d(p(i);X,Q) \}$$

for a function value of z_k is the union of the level sets of the functions $w_i d(p(i);X,Q)$ associated with each point Q in A(i). The level curve of $f_i(X)$ for the value z_k is the boundary of the level set of $f_i(X)$ and forms an envelope (Figure 4.18) about the level curves of the functions $w_i d(p(i);X,Q)$ associated with each point Q on the boundary of A(i).

In the Euclidean distance case, we used a unit vector T(m), perpendicular to the m'th side and directed toward the exterior of the polygonal area demand, to point toward that point on the level curve of $w_i d(p(i)=2;X,Q)$ which coincided with the level curve of $f_i(x)$. In the

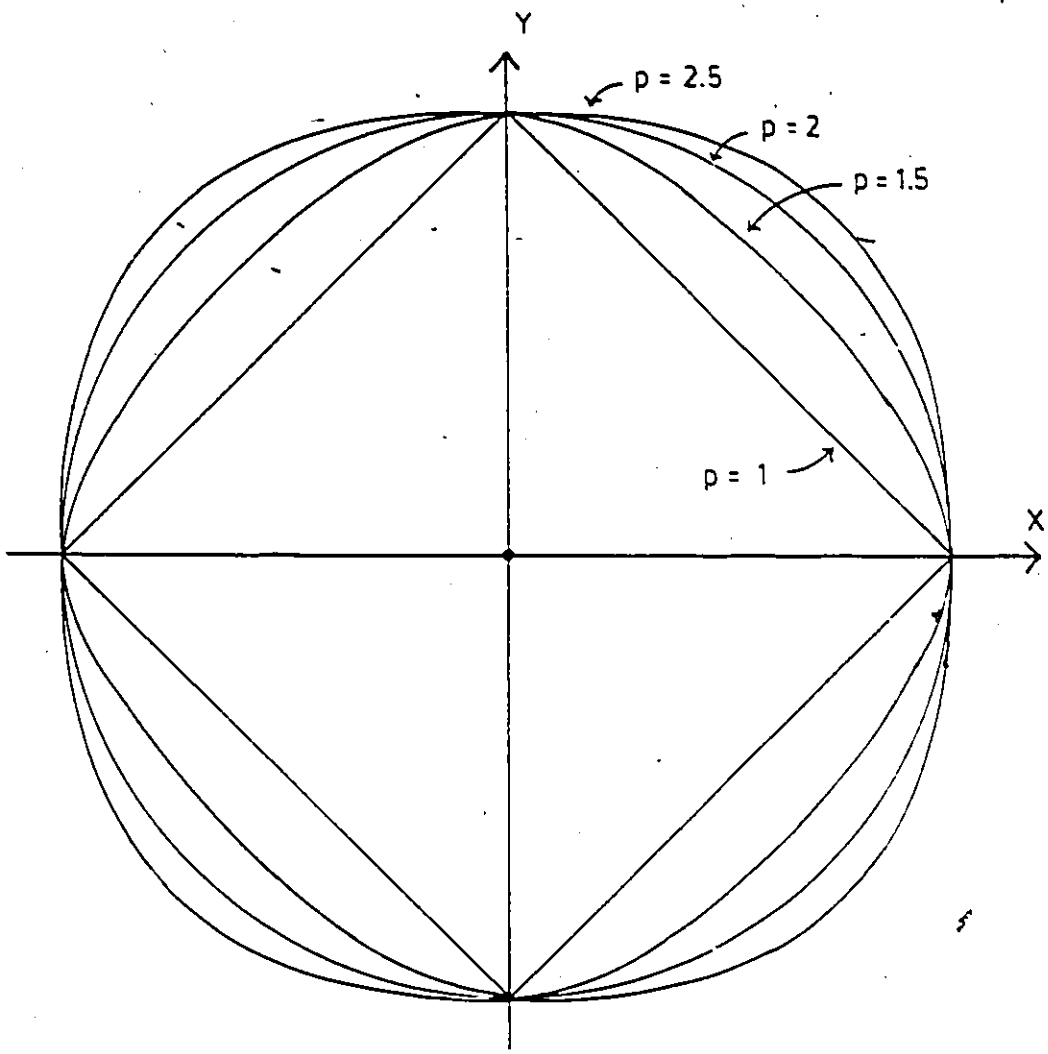

Figure 4.17. Lp circles for various values of p.

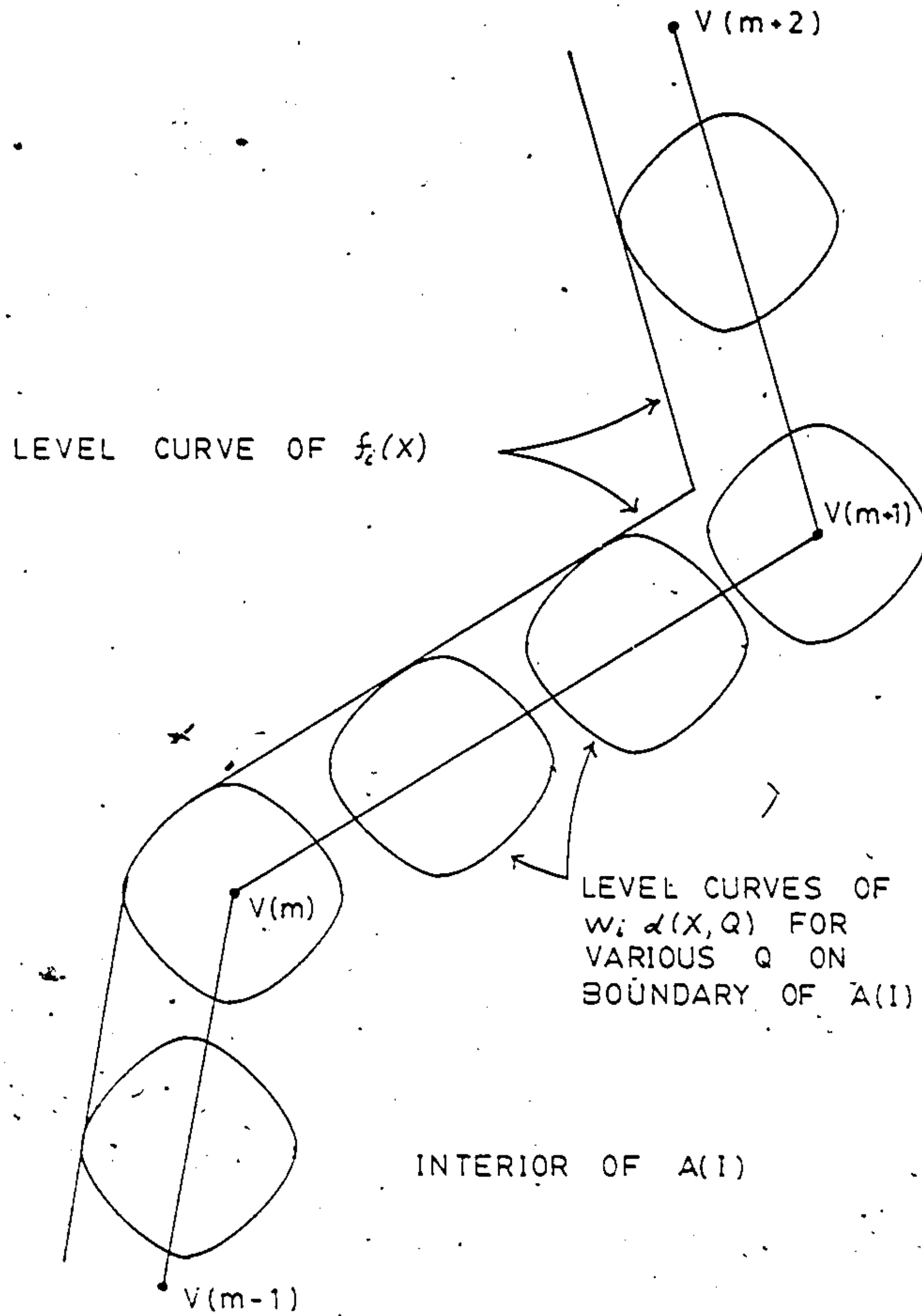

Figure 4.18. Lp level curve is envelope of Lp circles.

general l_p case, we construct a vector R(m) of unit l_p length directed toward that point on an l_p circle where the tangent is parallel to the m'th side of the area demand (Figure 4.19). Consider an l_p circle of radius Z and, for convenience, one with its center at the origin. This circle has the equation

$$|X|^p + |Y|^p = Z^p.$$

If we restrict our attention to points (X,Y) in the first quadrant, the absolute value brackets can be eliminated giving the simpler equation

$$X^p + Y^p = Z^p.$$

Implicit differentiation yields the following expression for the slope of the tangent at the point (X_o, Y_o):

$$dY/dX = -(X_o/Y_o)^{p-1} \quad \text{where } p > 1.$$

We note that the slope of the tangent to the l_p circle at (X_o, Y_o) depends only on the ratio of X_o to Y_o and not on the radius Z of the l_p circle. Of course, similar arguments apply to the other three quadrants. This implies that for a family of concentric l_p circles the locus of points on their circumferences which have a tangent of given slope is a straight line which passes through the center of the l_p circles (Figure 4.20). This useful property, well known in the Euclidean case, will be utilized when calculating the l_p distance to a line segment in subsection 4.2.7. This property also implies that the unit vector R(m), which we require, can be derived from the unit l_p circle.

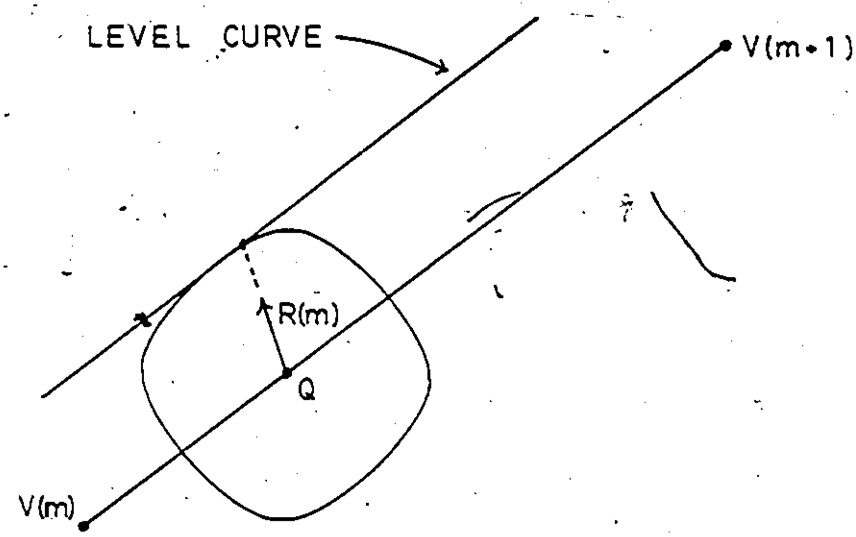

Figure 4.19. Vector R(m) associated with m'th side.

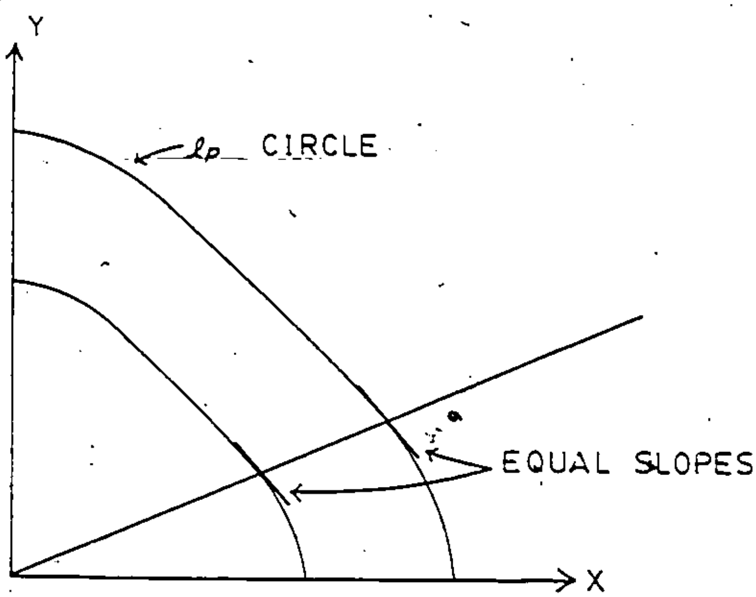

Figure 4.20. Tangent property of Lp circles.

Consider the equation of the unit l_p circle

$$|x|^p + |y|^p = 1.$$

For points (x,y) lying in the first quadrant, this can be simplified to

$$x^p + y^p = 1.$$

The slope of the tangents to this unit l_p circle at points lying in the first quadrant is in the range zero to negative infinity. If t is the slope of the tangent at such a point (x,y) in the first quadrant, then the coordinates of the point are given by

$$y_t = \{ 1/[1+(-t)^{(p/(p-1))}] \}^{(1/p)}$$

and $x_t = (-t)^{(1/(p-1))} \{ 1/[1+(-t)^{(p/(p-1))}] \}^{(1/p)}$

$$= (-t)^{(1/(p-1))} y_t$$

when $p > 1$. These expressions can be obtained by substituting the expression for the slope of the tangent into the equation of the unit l_p circle.

Let t be the slope of the vector S(m) = V(m+1)-V(m). If S(m) lies in the fourth quadrant, then (x_t, y_t) is a vector of unit l_p length and directed toward that point on the l_p circle where the slope of the tangent is t. Thus, $R(m) = (x_t, y_t)$ is the unit vector which we require whenever S(m) lies in the fourth quadrant. If S(m) does not lie in the fourth quadrant, then we can rotate the axes by some multiple of 90° to bring S(m) into the fourth quadrant, calculate (x_t, y_t), and then transform back to the original axes to get R(m) for S(m) lying in the first, second, or third quadrant.

For each side of the area demand, we must shade one parallelogram $P_1P_2P_3P_4$ and if the angle at the vertex is greater than 180°, one sector of an annulus of an l_p circle $P_5P_6P_2P_1$ in order to shade the area between the level curves of $f_i(X)$ for values z_{k-1} and z_k (Figure 4.21). Consider the m'th side of a polygonal area demand between vertices V(m) and V(m+1). The corners of the parallelogram are

$$P_1 = V(m) + (z_{k-1}/w_i)\ R(m),$$
$$P_2 = V(m) + (z_k/w_i)\ R(m),$$
$$P_3 = V(m+1) + (z_k/w_i)\ R(m),$$

and $$P_4 = V(m+1) + (z_{k-1}/w_i)\ R(m).$$

The other two corners of the sector of an l_p annulus are

$$P_5 = V(m) + (z_{k-1}/w_i)\ R(m-1)$$

and $$P_6 = V(m) + (z_k/w_i)\ R(m-1)$$

where R(m-1) is the unit vector associated with the m-1'th side. Although the parallelogram can easily be shaded by a polygon-filling routine, shading a sector of an annulus of an l_p circle is rather inconvenient. A graphic routine which draws a curved line through a given set of points and fills from a specified point to that curved line can be used to fill the sector of the l_p circle of radius z_k/w_i which lies between the directions of R(m) and R(m-1) or to fill the entire l_p circle. The additional shaded points lie in the dominated region and do not affect the solution of the maximin problem. The number of points on the circumference

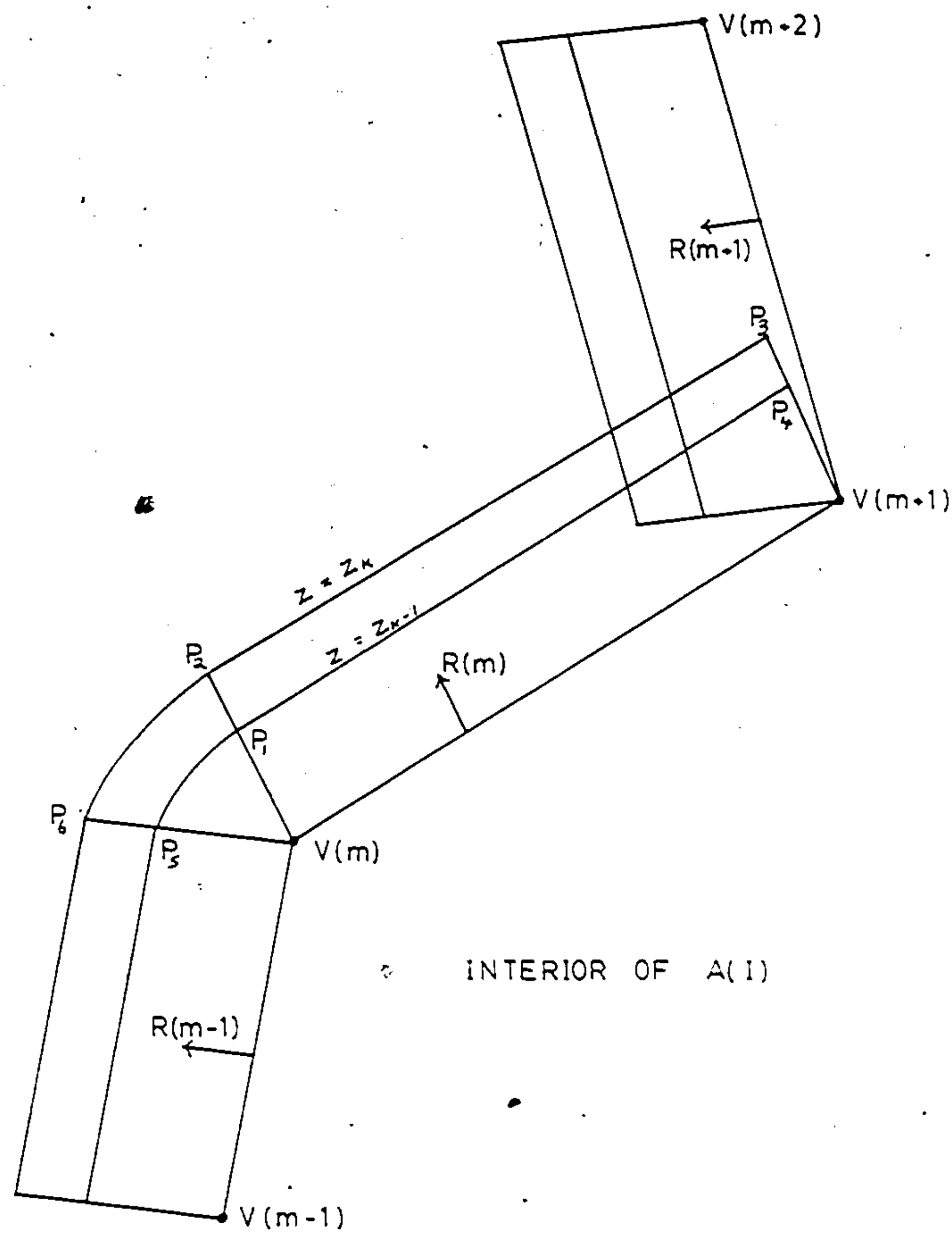

Figure 4.21. Parallelogram and sector of Lp annulus.

of the l_p circle whose coordinates need to be calculated for input to the curved line subroutine can be reduced by noting that only points on one-eighth of the circumference need be calculated and the rest can be obtained by symmetry.

The Euclidean distance case can be treated using the methods for general l_p distances given in subsections 4.2.6 and 4.2.7; but the rectilinear distance case must be treated separately because its level curves are not smooth with the slope defined at all points on the curves.

4.2.7 <u>L_p Distance to a Polygon</u>

Like the Euclidean and rectilinear distance cases, finding the minimum l_p distance from a point outside to a polygonal area demand reduces to finding the minimum l_p distance to a number of line segments.

Consider the line segment V(m) to V(m+1) and a point Q (Figure 4.22). If neither V(m) nor V(m+1) is the closest point on the line segment to Q, then the level curve for the minimum distance is tangent to the line segment V(m)V(m+1) at some point lying between V(m) and V(m+1). In order to find the minimum distance from Q to the line segment, calculate the intersection I of the line through V(m) and V(m+1) with the line through Q having the same slope as R(m). If the point I lies on the line segment V(m)V(m+1) between V(m) and V(m+1), then the minimum l_p distance to the line segment is the l_p distance from Q to I. Otherwise, the

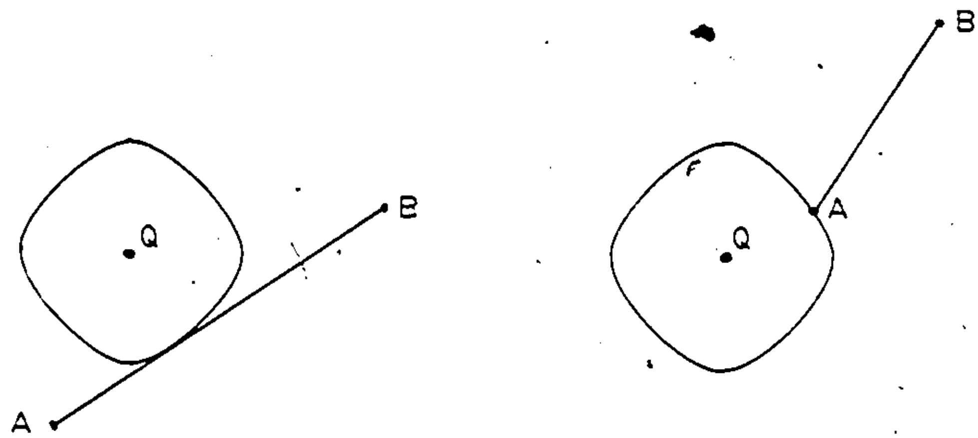

Figure 4.22. Lp distance to line segment.

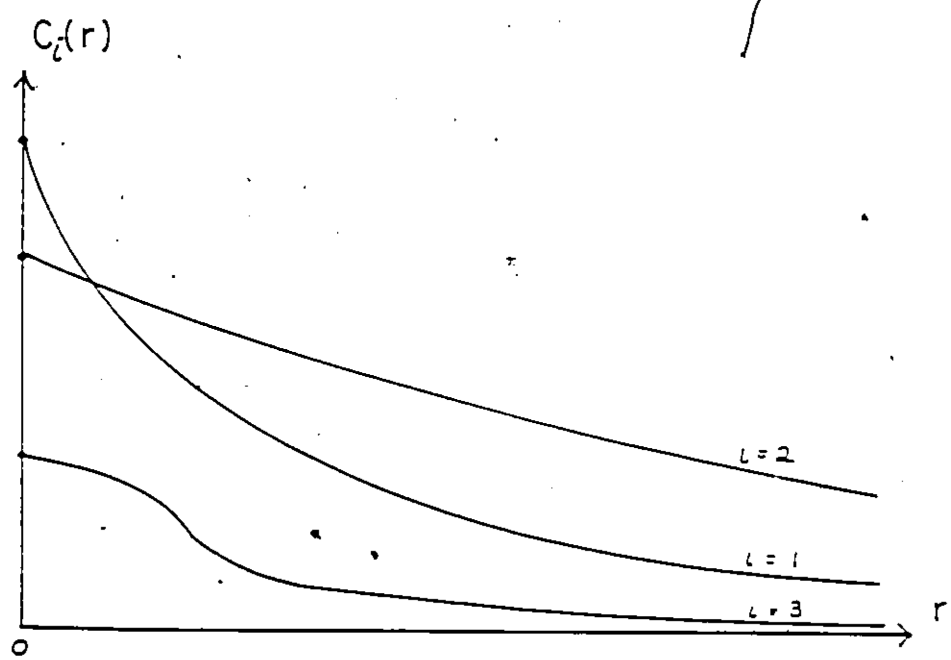

Figure 4.23 Nonlinear cost functions.

minimum distance is the l_p distance from Q to the closest end point, V(m) or V(m+1). If t is the slope of the line V(m)V(m+1) and p > 1, then from the results of subsection 4.2.6, the slope of the vector R(m) and the line QI is

$$-(1/t)^{(1/(p-1))} \quad \text{if } t > 0$$

and $$(-1/t)^{(1/(p-1))} \quad \text{if } t < 0.$$

The special cases which occur when V(m)V(m+1) is parallel to either axis must be treated separately.

4.3 Extension to Minimax Cost Problems

The obnoxious facility location model of Hansen, Peeters, and Thisse (1981) was formulated in terms of minimizing nuisance costs, which were assumed to be continuous and decreasing functions of distance, rather than maximizing distances to the facility. They noted that the perceived advantage in moving a noxious facility ten miles further away is larger when the facility is located close by then when it is initially further away. Thus, they concluded that linear cost functions with constant marginal costs are often too rough an approximation to real nuisance costs. Fortunately, the algorithm for maximin distance facility location with area demands, presented in section 4.2, is easily adapted to minimax cost problems when the costs are continuous, decreasing functions of distance alone.

The minimax criterion may be more appropriate for locating a noxious facility than the minisum criterion if the variation of the facility's effects with distance is not known with reasonable accuracy, because minimizing the sum of the effects implicitly involves trading off increased damage to some areas in return for reduced damaged to others. If the variation of the damage as a function of distance is not known with reasonable accuracy, one has insufficient information on which to make these trade-offs and might more appropriately attempt to minimize the damage to the most severely affected area.

In particular, the minimax criterion can be more appropriate than the minisum criterion for the location of a noxious facility which is believed to pose health hazards to the surrounding population. The variation with distance of the levels of noxious physical or chemical agents about the facility may be known with reasonable accuracy from theory or measurement. However, the relationship between the amount of exposure to the agent and the resulting health effects may be largely unknown at the low levels to which the population is actually exposed since the effects at those levels may be so small that they defy detection in any practicable experiment and cannot be isolated from other effects in survey data. In such cases, one has insufficient information to trade off damage between areas even though exposures as a function of distance are well known. In addition, if it is suspected that there exists a threshold level below which there are no adverse health effects, then the criterion of minimizing the maximum exposure seems to be the most appropriate criterion.

4.3.1 Nonlinear Cost Functions

Let $C_i(r)$ be a non-negative, continuous, monotonically decreasing function defined on the range $r \geq 0$ which is associated with the area demand A(i) for i=1,...,N. Let the limit of $C_i(r)$ as r tends to infinity be 0 and let $C_i(0)$ be finite for all i=1,...,N (Figure 4.23). Let $B_i(z)$ be the inverse of the function $C_i(r)$ defined on the range $0 < z \leq C_i(0)$ for i=1,...,N. For convenience, assume $C_1(0) \geq C_2(0) \geq \ldots \geq C_N(0)$. If this is not the case, then renumber the area demands accordingly.

The requirements that the nuisance costs $C_i(\cdot)$ be finite at zero distance and have a limit of zero cost as the distance from the facility tends to infinity would seem to be met in all practical problems. For example, the well known $1/r^2$ dependence of the intensity of light or sound from a point source is only an approximation which breaks down at small r where the source can not longer be regarded as a point and the intensity remains finite at r = 0. The requirement that the $C_i(\cdot)$ be monotonically decreasing functions appears to be more restrictive and may not hold in some practical problems.

We note that if identical monotonically decreasing cost functions are associated with each of the area demands A(i), then any location which maximizes the minimum distance to the area demands must also minimize the maximum cost. Thus, any minimax cost problem with $C_i(r)=C_j(r)$ for all i and j and for all $r \geq 0$ can be reduced to the maximin distance problem considered in sections 4.1 and 4.2. Accordingly, we assume that $C_i(r) \neq C_j(r)$ for some i and j.

The minimax cost single facility location problem with l_p distances and polygonal area demands can be formulated as

$$\underset{X \text{ in } F^*}{\text{minimize}} \left\{ \underset{i=1,\ldots,N}{\text{maximum}} \; h_i(X) \right\}$$

where

$$h_i(X) = \underset{Q \text{ in } A(i)}{\text{maximum}} \left\{ C_i(d(p(i);X,Q)) \right\}$$

$X = (x,y)$ = location of new facility,

$Q = (a,b)$ = location of a general point,

$p(i)$ = p-value for points in A(i),

and $d(p(i);X,Q) = [|x-a|^{p(i)} + |y-b|^{p(i)}]^{1/p(i)}$

$= l_{p(i)}$ distance from X to Q.

Since the same monotonically decreasing function $C_i(\cdot)$ is associated with all points in a given area demand A(i), the maximum over all points in A(i) will occur at that point Q which minimizes $d(p(i);X,Q)$. Thus, the function $h_i(X)$ can be restated as

$$h_i(X) = C_i(\, f_i(X) \,)$$

where

$$f_i(X) = \underset{Q \text{ in } A(i)}{\text{minimum}} \{ d(p(i);X,Q) \}.$$

This $f_i(X)$ is identical to the $f_i(X)$ which appeared in the maximin distance problem defined in section 4.1 except that all $w_i = 1$. The weights w_i are no longer necessary since the functions $C_i(\cdot)$ can include all of the differences in costs between the area demands A(i).

We note that the level curve of the function $f_i(X)$ for a value d_k is identical to the level curve of the cost function $h_i(X)$ for a value of $C_i(d_k)$ (Figure 4.24). Thus, we can shade the area between the level curves of $h_i(X)$ for values z_{k-1} and z_k by shading the area between the level curves of $f_i(X)$ for the values $B_i(z_{k-1})$ and $B_i(z_k)$. We also note that the procedures for shading the area between level curves of $f_i(X)$ in some cases shade points closer to the area demand than $B_i(z_{k-1})$ but this does not affect the solution of the minimax problem because these additional points have objective function values which are higher and less optimal than the current best objective function value. These additional points will lie in the dominated region for the minimax cost problem and will already have been shaded by the minimax algorithm. Thus, the geometrical procedures developed for the graphical solution of the maximin distance problem are directly applicable to the minimax cost problem and only minor modifications to the maximin distance

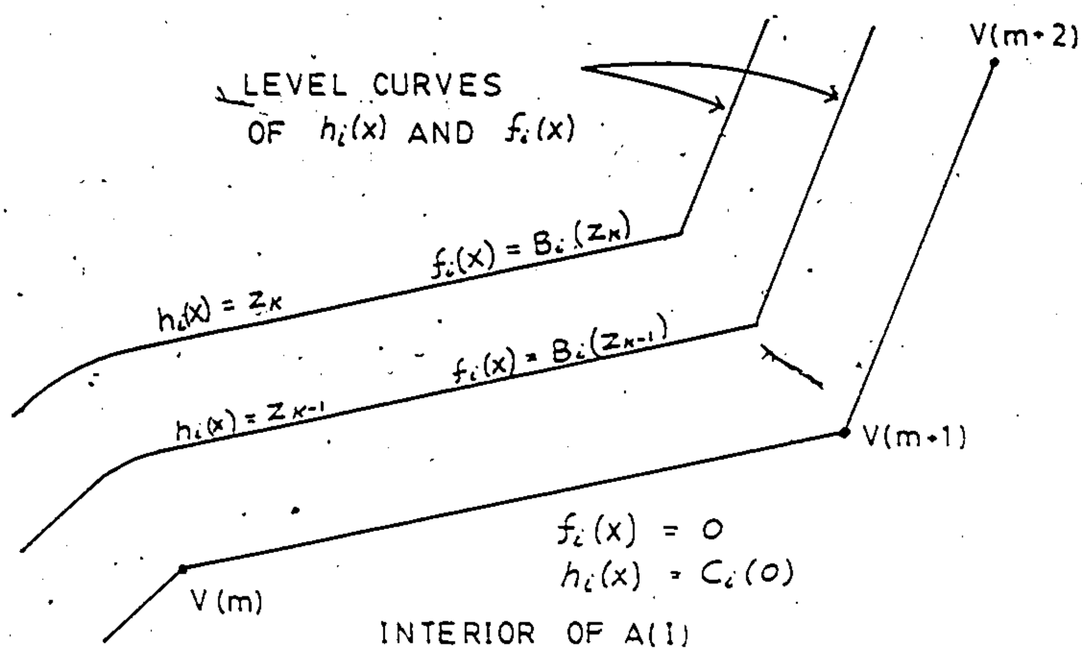

Figure 4.24. Level curves for cost and distance functions.

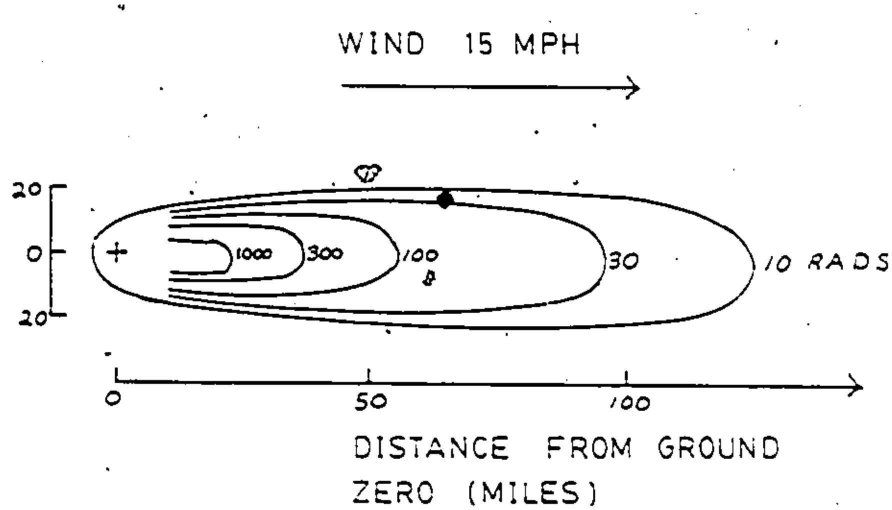

Figure 4.25. Contours of total dose due to fallout.

algorithm are necessary.

The graphical algorithm for the minimax cost problem starts with the highest possible objective function value $C_1(0)$ and works downward toward the optimal value. A minor complication is caused by the fact that points in the interiors of different area demands may have different costs associated with them; that is, $C_i(0)$ and $C_j(0)$ may not be equal. As a result, only the area demands with $C_i(0) = C_1(0)$ are shaded in the initialization step and the remaining area demands are shaded when their $C_i(0)$ is greater than current objective function value z_k.

The following is an algorithm for the minimax cost location problem as defined in this section. As can be seen, only minor modifications of the maximin distance algorithm were required.

Step 1. <u>Initialization</u>

Choose the region of the plane which will be represented on the screen so that it encloses the entire feasible region for facility placement. Shade the infeasible region. Shade the interiors and boundaries of all area demands A(i) for which $C_i(0) = C_1(0)$.

If no unshaded point remains on the screen, then stop. Optimum objective function value is $C_1(0)$; pick any feasible point as an optimum location.

Otherwise, set $k = 1$ and $z_0 = C_1(0)$.

Step 2. Improve Value of Objective Function

Choose any unshaded point X_k, preferably one near the middle of the unshaded region. Calculate

$$z_k = \max \{ h_i(X_k);\ i=1,\ldots,N\}.$$

Step 3. Eliminate Dominated Regions

Shade the area between the level curves of $h_i(X)$ corresponding to the values z_{k-1} and z_k for each $i=1,\ldots,N$. This is equivalent to shading the area between the level curves of $f_i(X)$ for the values $B_i(z_{k-1})$ and $B_i(z_k)$. If $z_{k-1} \geq C_i(0) > z_k$ for some demand i, then this should be interpreted as shading the polygonal area demand i and the area between the level curves of $h_i(X)$ for the values $C_i(0)$ and z_k. If both z_{k-1} and z_k are greater than $C_i(0)$, then shade nothing since there is no region of the plane which lies between these particular level curves for the demand i.

Set $k = k + 1$.

If only a few points (pixels) remain unshaded, then go to Step 4. Otherwise, go to Step 2.

Step 4. Accept or Expand Scale

If the size of the remaining unshaded area is sufficiently small so that the uncertainty in the location of the optimum is acceptable, then choose a point in the remaining unshaded area as a near-optimal location X^*. Calculate $z^* = \max \{ h_i(X^*);\ i=1,\ldots,N\}$. Stop.

Otherwise, choose a small rectangle enclosing the unshaded region and expand the scale so that the rectangle fills the screen. Shade the infeasible region and the area demands A(i) for which $C_i(0) = C_1(0)$. For all i=1,...,N, shade the area between the level curves of $h_i(X)$ corresponding to the values $C_1(0)$ and z_{k-1} following the interpretation of special cases as explained in Step 3. Go to Step 2.

4.3.2 Non-isotropic Cost Functions

Hansen, Peeters, and Thisse (1981) remarked that their algorithm for point demands, BLACK AND WHITE, could be extended to take into account non-isotropies in the nuisance costs such as might occur when dominant winds diffuse pollution further in some directions than in others. An example of the theoretical effect of wind direction on the deposition of particles is given in Figure 4.25 which shows the contours of the total radiation dose due to fallout 18 hours after a 2 megaton nuclear explosion with a wind of 15 mph (Glasstone and Dolan, p.426). Such non-isotropic distributions of noxious emissions are probably quite common and cannot be adequately modeled by cost functions which depend only on the distance from the facility.

We will show that the graphical algorithm for minimax cost location problems can be extended with little difficulty to a class of non-isotropic cost functions based on the asymmetric distance functions introduced by Drezner

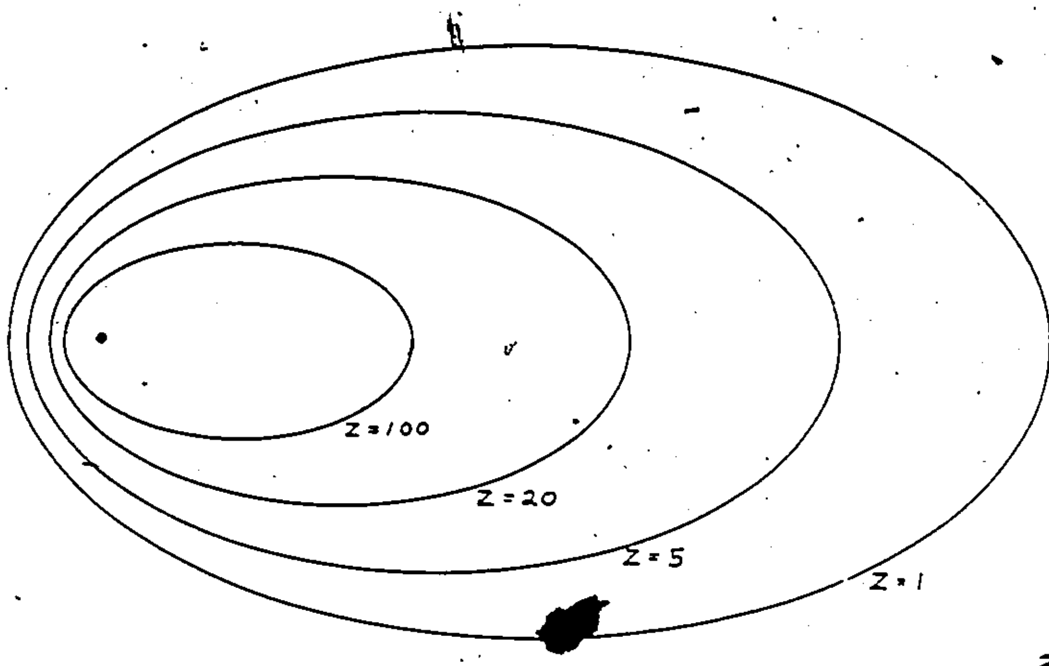

Figure 4.26. Family of level curves with same shape.

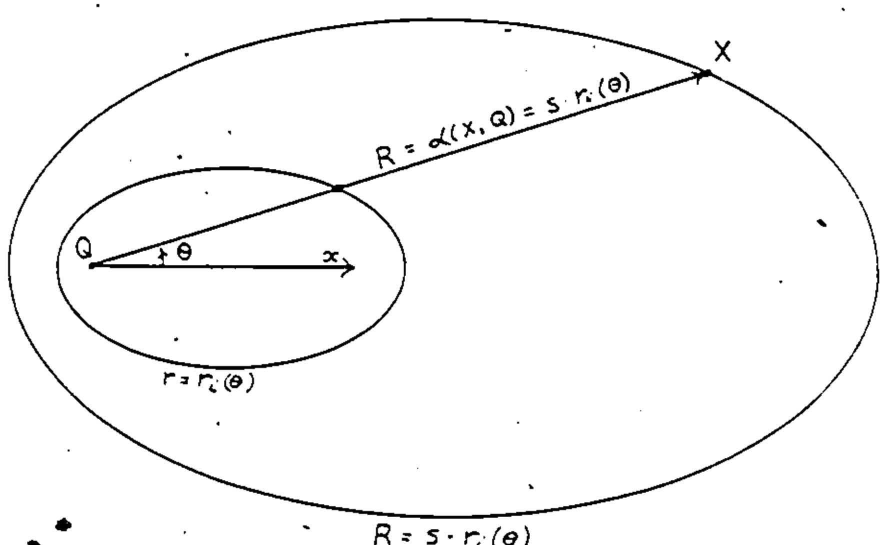

Figure 4.27. Relationship of level curves to r(θ).

and Wesolowsky (1986). All of the iso-cost contours of the type of function which we will consider have the same shape and orientation but differ in scale (Figure 4.26). That is, any two contours can be made to coincide by applying a translation transformation and a change of scale to one of the contours. This is in contrast to the dose contours shown in Figure 4.25 which differ in shape. Although somewhat restricted, this class of non-isotropic cost functions can provide much better approximations to actual non-isotropic nuisance costs than cost functions which depend on distance alone.

A minimax non-isotropic cost single facility location problem with polygonal area demands can be formulated as

$$\underset{X \text{ in } F}{\text{minimize}} \; \{ \underset{i=1,\ldots,N}{\text{maximum}} \; g_i(X) \}$$

where

$$g_i(X) = \underset{Q \text{ in } A(i)}{\text{maximum}} \{ C_i(d(X,Q)/r_i(\theta(X-Q))) \}$$

$X = (x,y)$ = location of new facility,

$Q = (a,b)$ = location of a general point,

$d(X,Q)$ = Euclidean distance from Q to X

$= [(x-a)^2 + (y-b)^2]^{1/2}$,

$\theta(X-Q)$ = angle measured counterclockwise from positive x-direction to vector X-Q,

$r_i(\theta(X-Q))$ = a function of $\theta(X-Q)$ for Q in A(i),

and the $C_i(\cdot)$ are monotonically decreasing cost functions as defined in subsection 4.3.1.

The functions $r_i(\theta)$ determine the directional dependence of the non-isotropic costs. We require that the functions $r_i(\cdot)$ be such that $r=r_i(\theta)$ is the equation in polar coordinates of a simple closed curve enclosing the origin and that the interior of the curve is convex. We will call the quantity $d(X,Q)/r_i(\theta(X-Q))$ the asymmetric distance from Q to X. Although we have placed the directionally dependent function in the denominator rather than in the numerator for our convenience, this asymmetric distance function is essentially that introduced by Drezner and Wesolowsky (1986).

The relationship between the level curves of the function $C_i(\ d(X,Q)/r_i(\theta(X-Q))\ )$ and the function $r_i(\theta)$ can be seen by noting that the level curve of $d(X,Q)/r_i(\theta(X-Q))$ for a value s coincides with the level curve of the cost function for a value $C_i(s)$. Thus, the equation $d(X,Q) = s\cdot r_i(\theta(X-Q))$ with s as a parameter describes the entire family of level curves of the cost function about the point Q (Figure 4.27). Each level curve of $C_i(d(X,Q)/r_i(\theta(X-Q)))$ for the value $C_i(s)$ has exactly the same shape as $r_i(\theta(X-Q))$ but is s times larger. The curve $r = r_i(\theta)$ can be regarded as the unit circle for the asymmetric distance function, since every point on it is one unit of asymmetric distance from Q.

Only the Euclidean distance function was needed in

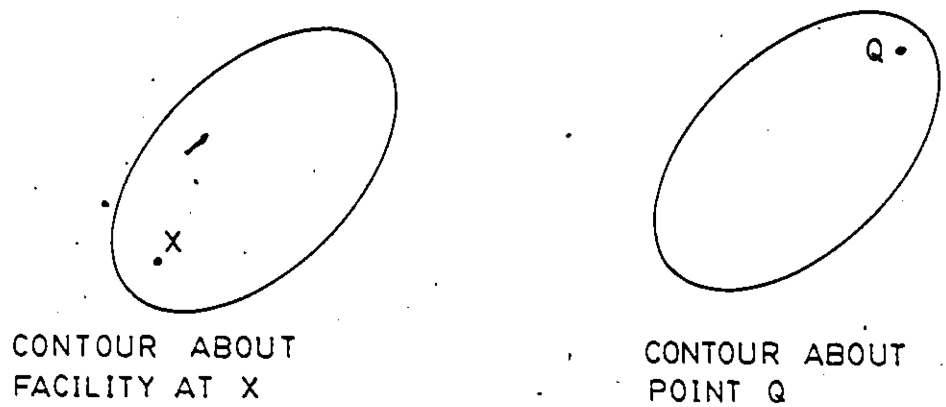

Figure 4.28. Fixed facility versus fixed vulnerable point contours.

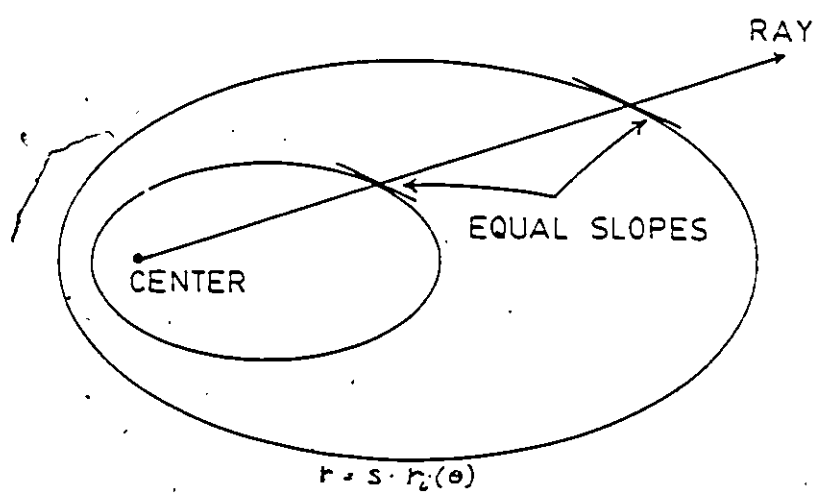

Figure 4.29. Tangent property of asymmetric distance circles.

the formulation of this problem because the functions $r_i(\theta)$ can include any variation with direction which would be produced by using rectilinear or general l_p distances. In the non-isotropic case, we must distinguish between contours about a facility which describe the costs imposed on points in the plane as in Figure 4.25 and contours about a point Q which describe the costs imposed on Q if a facility is placed at various points in the plane (Figure 4.28). The form of the contours is the same except that they are rotated 180° from each other. This occurs because the cost which a facility imposes on a point ten miles downwind is the same as the cost imposed on a point by locating a facility ten miles upwind. In our formulation, the function $r_i(\theta)$ describes the level curves about the point Q and $C_i(\cdot)$ gives the cost imposed on Q by locating the facility at various points in the plane.

Like the general l_p distance case, we require R(m), a unit vector in the asymmetric distance sense, which is directed toward the exterior of A(i) and which points toward that point on the asymmetric distance circles where the slope is equal to the slope of the m'th side. Let t be the slope of the tangent to the asymmetric distance circle $r = s\cdot r_i(\theta)$ at the point $(x_t, y_t) = (r\cdot\cos(\theta), r\cdot\sin(\theta)) = (s\cdot r_i(\theta)\cos(\theta), s\cdot r_i(\theta)\sin(\theta))$. Taking the ratio of the differentials $dx_t = x_t(dr_i/d\theta)d\theta - y_t d\theta$ and $dy_t = y_t(dr_i/d\theta)d\theta + x_t d\theta$, we obtain

$$t = \{ y_t(dr_i/d\theta) + x_t \} / \{ x_t(dr_i/d\theta) - y_t \}$$

where $dr_i/d\theta$ is the derivative of $r_i(\theta)$ with respect to θ and is evaluated at a value of θ corresponding to the direction of the point (x_t, y_t) from the center of the asymmetric distance circle. We note that all points on a ray from the center of a family of asymmetric distance circles will intersect the circles at points which have tangents of the same slope (Figure 4.29). This is similar to the property found for l_p distance circles and would not necessarily hold if the level curves of the distance function did not have the same shape.

Given t the slope of the m'th side, solving the preceding expression for the ratio of y_t to x_t in order to get the unit vector R(m), may be more difficult than in the general l_p case, depending on the particular function $r_i(\theta)$. If a simple closed form expression in terms of t cannot be found, the equation could be solved numerically. This is not a significant computational burden since it only needs to be done once for each side of each demand during the initialization stage of the algorithm and the results stored for later use.

Consider the non-isotropic cost function associated with the area demand A(i),

$$g_i(X) = \underset{Q \text{ in } A(i)}{\text{maximum}} \{ C_i(d(X,Q)/r_i(\theta(X-Q))) \}.$$

Since $C_i(\cdot)$ is a monotonically decreasing function the

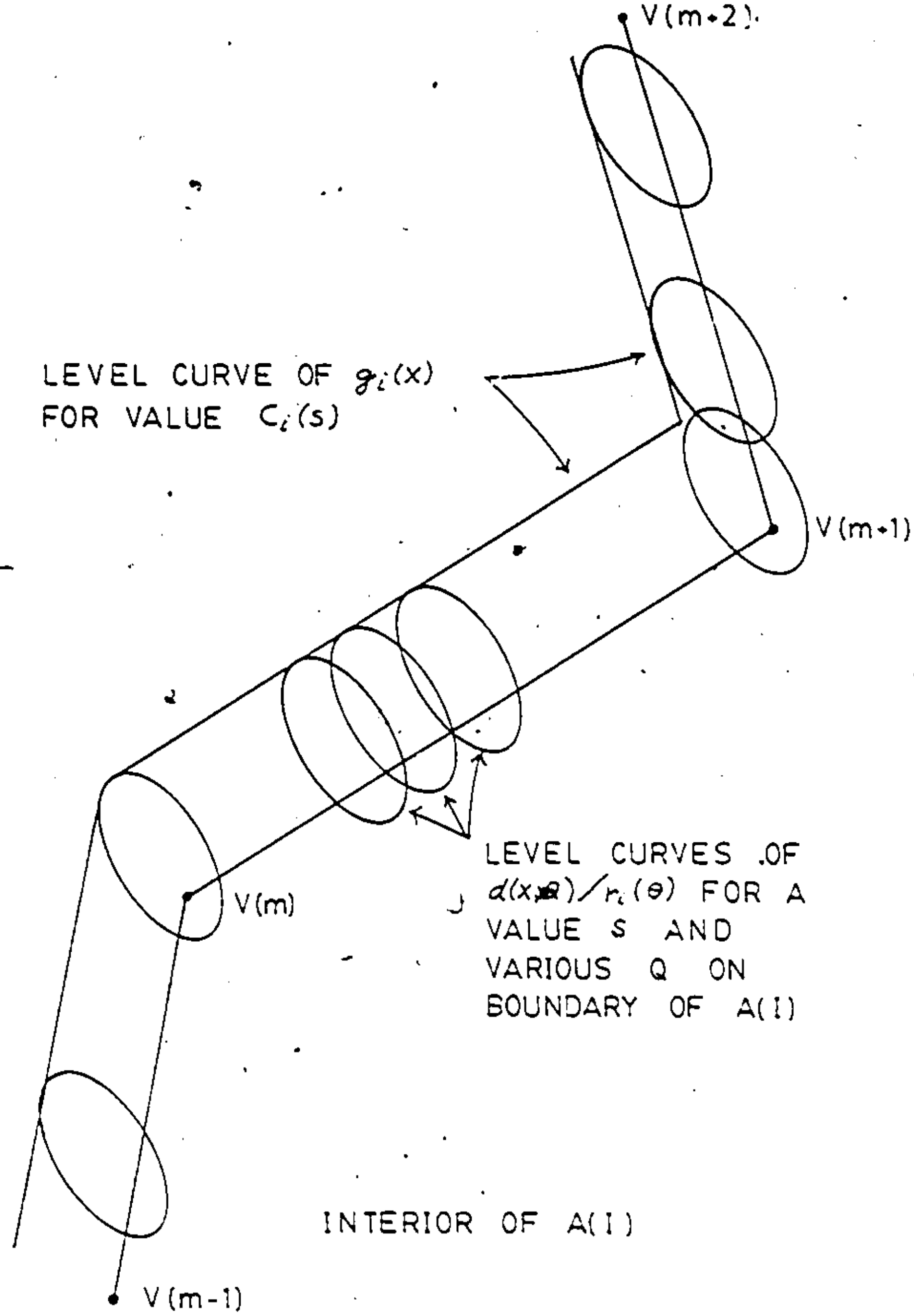

Figure 4.30. Asymmetric distance level curve is envelope of asymmetric circles.

maximum will occur for that Q which minimizes the asymmetric distance $d(X,Q)/r_i(\theta(X-Q))$. The level curves of the function $g_i(X)$ for a value $C_i(s)$ form an envelope of the level curves for the value s of the functions $d(X,Q)/r_i(\theta(X-Q))$ associated with the points Q on the boundary of A(i) (Figure 4.30).

The area between the level curves of $g_i(X)$ can be shaded by shading one parallelogram $P_1P_2P_3P_4$ for each side of the polygonal area demand and one region $P_5P_6P_2P_1$ for each vertex with an exterior angle greater than 180° (Figure 4.31). The corners of the parallelogram are given by

$$P_1 = V(m) + B_i(z_{k-1})\cdot R(m),$$
$$P_2 = V(m) + B_i(z_k)\cdot R(m),$$
$$P_3 = V(m+1) + B_i(z_k)\cdot R(m),$$

and $$P_4 = V(m+1) + B_i(z_{k-1})\cdot R(m)$$

where $B_i(\cdot)$ is the inverse of $C_i(\cdot)$ as defined in subsection 4.3.1. The remaining two corners of the region $P_5P_6P_2P_1$ are given by

$$P_5 = V(m) + B_i(z_{k-1})\cdot R(m-1)$$

and $$P_6 = V(m) + B_i(z_k)\cdot R(m-1)$$

where R(m-1) is the unit vector associated with the m-1'th side. The curved lines P_5P_1 and P_6P_2 are sections of the asymmetric distance circles about V(m) with $s=B_i(z_{k-1})$ and $s=B_i(z_k)$, respectively. Thus, one can shade the region $P_5P_6P_2P_1$ in the same manner as shading a sector of an annulus of an l_p circle and with about the same amount of

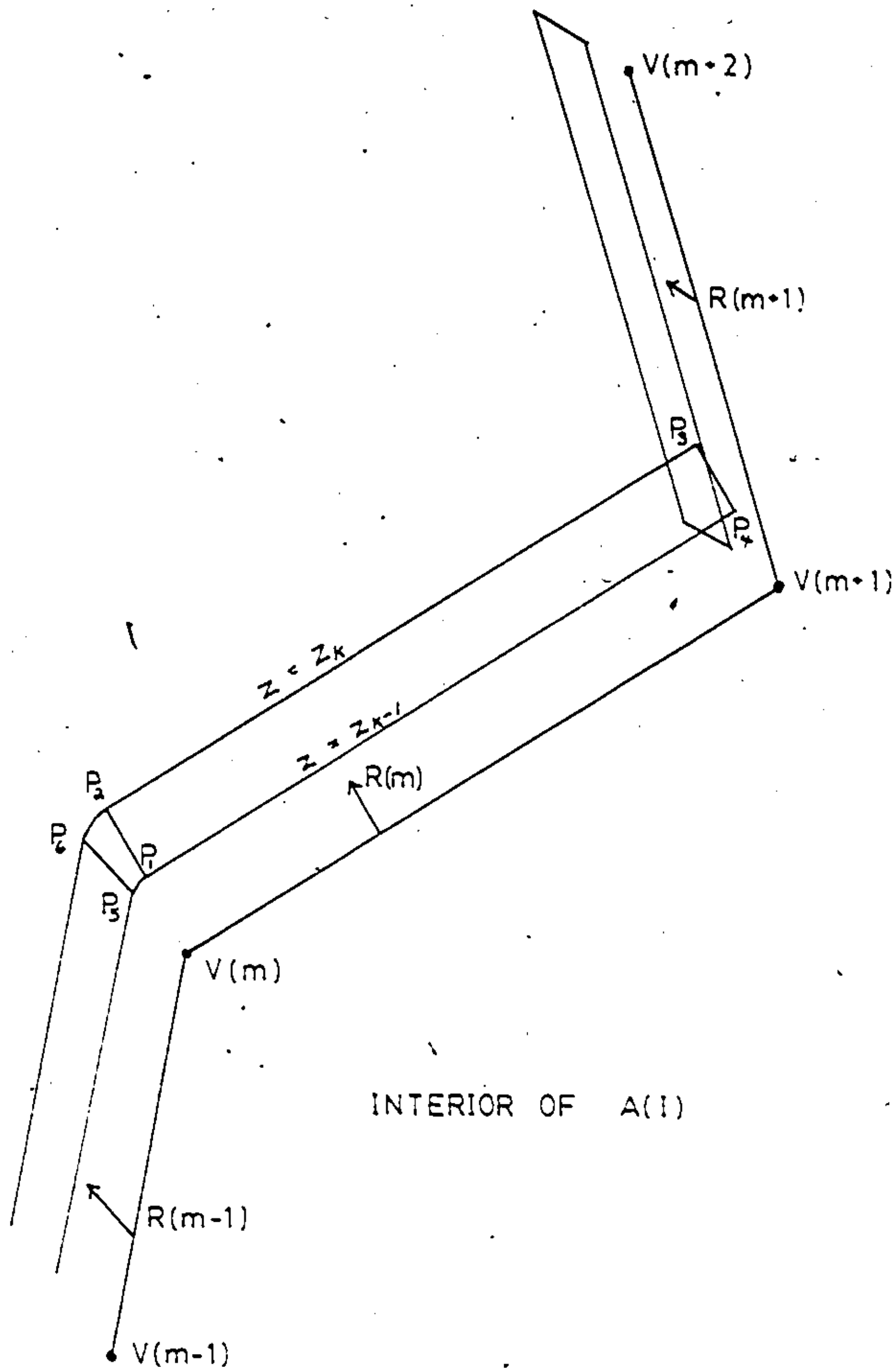

Figure 4.31. Parallelogram and sector of asymmetric annulus.

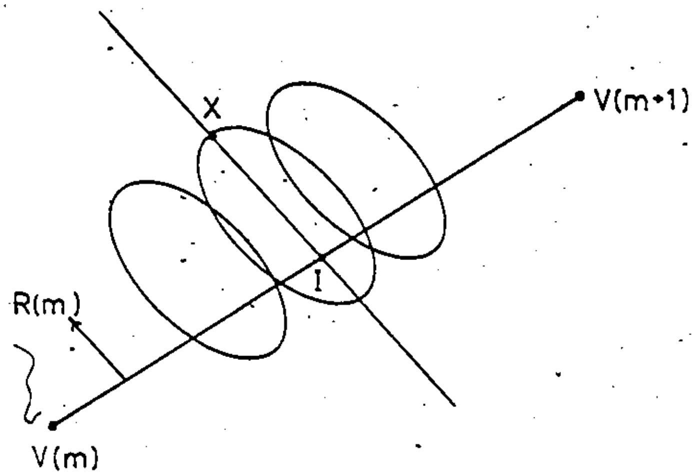

Figure 4.32. Asymmetric distance to line segment.

difficulty. Similarly, a sector of an asymmetric distance circle or an entire asymmetric distance circle could be drawn about the vertex V(m) without affecting the solution of the minimax asymmetric cost problem.

Evaluating the cost function $g_i(X)$ at a point X outside the area demand A(i) reduces to finding the minimum asymmetric distance from a point on a line segment to X (Figure 4.32). In order to find the minimum asymmetric distance from m'th side to a point X, find the intersection I of the line through V(m) and V(m+1) with the line through X having the same slope as R(m). If I lies between V(m) and V(m+1), then the minimum distance is the asymmetric distance from I to X. Otherwise, the minimum distance is the asymmetric distance from the closest end point, V(m) or V(m+1), to X.

4.4 Demonstration Program

4.4.1 Description

The graphical algorithm for single facility maximin distance location with area demands was implemented on a Digital Equipment Corporation (DEC) Professional 350 personal computer. The program was written in FORTRAN 77 and utilized DEC's Core Graphics Library of graphic subroutines which follow the proposed Core Graphics System standard of the ACM SIGGRAPH Graphics Standards Planning Committee (1979). DEC's implementation of CORE includes several extensions which were used in the demonstration program. In particular, the routine which fills a polygon given its vertices, the routine which fills a sector of a circle, and the routine which draws a curved line and fills from a point to that curved line were used in the demonstration program.

The demonstration program consists of approximately 2200 lines of not very tightly written FORTRAN. A maximum of 20 area demands are allowed and each demand may have a different weight and l_p distance function associated with it. There may be up to a total of 200 vertices associated with the area demands and they may be divided among the individual area demands in any manner as long as the total number does not exceed 200. The feasible region is assumed to consist of one or more polygons up to a maximum of 20 with the total number of vertices not exceeding 200. Point demands are allowed and treated as a special case of polygonal area demands.

The information on the area demands and the feasible region is input from data files. In a frequently used application not requiring high accuracy, this could be done more conveniently using a graphics tablet as an input device. Positions on the screen are indicated by moving the cursor about using the keyboard.

Since there may be more than one small region which the user would like to expand in Step 4 of the algorithm, the program allows more than one region to be picked for expansion. The most recently picked region is expanded and the information on the other regions is placed on a stack for later retrieval.

4.4.2 Examples

An example with realistically shaped geographical regions was generated from a road map of Ontario (Figure 4.33). The feasible region is a polygon with 196 vertices which approximates the shape of Southern Ontario. Five area demands were generated which approximate Essex County, Hamilton-Wentworth Region, Metropolitan Toronto, Ottawa-Carleton Region, and Algonquin Provincial Park. The polygons representing the demands have a combined total of 107 vertices. The coordinates of the vertices of the feasible region and the area demands are given in Appendix B.

For Euclidean distances and all weights equal to one, the best location found was at (15.975, 54.238). This is very close to the feasible region vertex with coordinates (15.9, 54.2) which represents Cape Hurd on the north-west tip of the Bruce Peninsula. The minimum weighted distance to the nearest area demand, Algonquin Provincial Park, was 26.650 units. The next closest area demand was Metropolitan Toronto at a distance of 28.968 units.

For rectilinear distances and all weights equal to

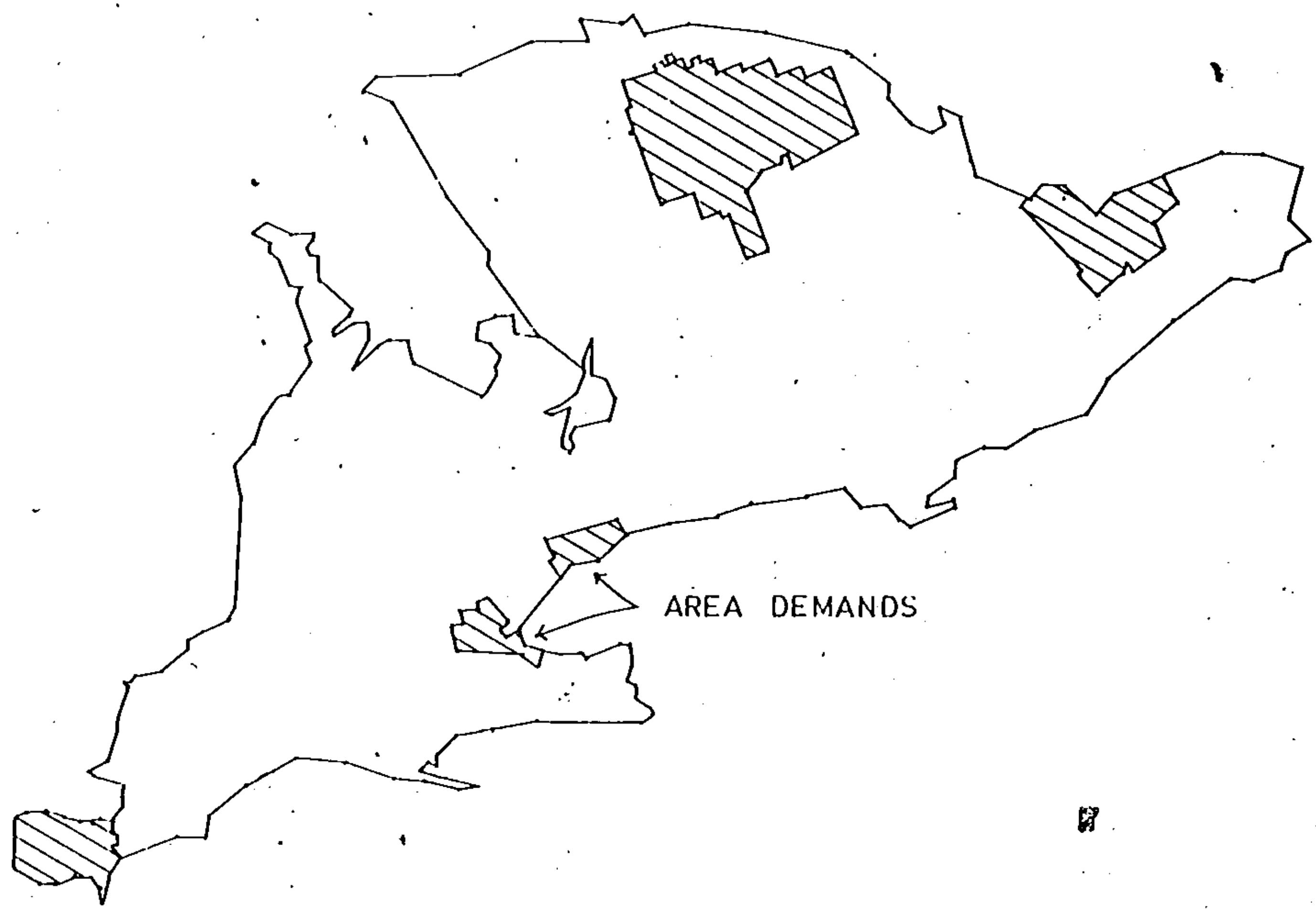

Figure 4.33. Southern Ontario example.

one, the best location found was at (19.337, 49.351) which corresponds to a point on the west side of the Bruce Peninsula about halfway down from the tip of the peninsula. The minimum weighted distance to the nearest area demand, Algonquin Provincial Park, was 31.612 units and the next closest area demand was Hamilton-Wentworth Region at a distance of 31.613 units.

For general l_p distances with $p = 1.5$ and all weights equal to one, the best location found was at (15.948, 54.221). Like the Euclidean case, this is very close to the feasible region vertex with coordinates (15.9, 54.2) which represents Cape Hurd. The minimum weighted l_p distance to the nearest area demand, Algonquin Provincial Park, was 28.118 units. The next closest area demand was Hamilton-Wentworth Region at a distance of 32.268 units.

CHAPTER 5

SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH

This chapter summarizes the research performed and presents recommendations for future research related to the topics investigated.

5.1 Summary

This research presents an extension of graphical and interactive computer graphical approaches to previously unsolved location problems. The problems studied were of two distinct types. The first type involved the maximization of the minimum distance or the minimization of the maximum distance to a set of point demands where the underlying distances were rectilinear and a number of right-angled barriers to travel were present. The second type involved the maximization of the minimum distance to a number of polygonal area demands.

Chapter 1 presents a brief introduction to facility location problems as well as the maximin and minimax criterions, and provides a simple example of the graphical approach. Chapter 1 also points out the importance of barriers to travel in practical location problems. Chapter 2 presents a literature survey of previous research related to the two types of location problems studied.

Chapter 3 considers maximin and minimax location problems with rectilinear distances and right-angled barriers aligned with the travel directions. We present a division of the plane into cells and prove certain properties of these cells which give us a practical method of applying the graphical approach to these problems. An algorithm for the maximin problem is presented and those changes needed to convert it to an algorithm for minimax problems are also given. The maximin algorithm was programmed and an optimal solution to a large example problem was obtained. For unweighted maximin problems which have feasible regions in the form of one or more polygons, a purely numerical method is given.

Chapter 4 considers the problem of locating a noxious facility so that minimum distance to a set of polygonal area demands is maximized. Rectilinear, Euclidean, and general l_p distances are treated and extensions to asymmetric distances and non-linear cost functions are discussed. An interactive computer graphic algorithm for maximin location with rectilinear, Euclidean, or general l_p distances is presented. The algorithm was programmed and optimal solutions were obtained for a large example problem using various distance metrics.

5.2 Recommendations for Future Research

Based on this research, four areas suggest themselves for further investigation. We present them in the order of expected difficulty.

5.2.1 Cell Consolidation

We have proved that the distance function inside a cell as defined in Chapter 3 has a particularly simple form and that the level curves also have a simple form inside a cell. Although necessary for Phase I, this division into cells may be finer than necessary for Phase II of the algorithm. We have observed that a row or rectangular block of cells can often be replaced by a single large cell since the same simple pattern of level curves often extends over many cells. Whether two adjacent cells can be consolidated into a larger single cell can be determined by examining the distance function at the corners of the cells. The potential benefits of consolidating cells at end of Phase I of algorithm are increased execution speed in Phase II of the algorithm and reduced storage requirements if a sufficient number of cells can be consolidated to offset increased storage requirements per cell. We recommend that the advantages and disadvantages of consolidating cells as the final stage of Phase I of the algorithm be investigated.

5.2.2 Numerical Minimax With Barriers

A purely numerical algorithm for unweighted minimax location problems with rectilinear distances and barriers appears to be more difficult than the algorithm for unweighted maximin problems. In the maximin case, we were able to greatly simplify the problem by interchanging the minimization over the demand points and the minimization over the cell corners. In the minimax case, we are faced with a maximization over demand points and a minimization over cell corners so that no comparable interchange and simplification is possible. However, the set of distances from each demand point to each of the cell's corners still contains in a relatively simple form all of the information required to calculate the objective function at any point inside the cell and to determine the minimum objective function inside the cell. Accordingly, we recommend that a purely numerical algorithm for unweighted minimax location problems with rectilinear distances and barriers to travel be developed which is based on the division of the plane into cells as presented in Chapter 3.

5.2.3 General Polygonal Barriers

Larson and Li's theory and algorithm were developed for general polygonal barriers to travel but the barriers that we considered in Chapter 3 were restricted to be right-angled polygons aligned with the travel directions. Restricting the allowed barrier shapes to this common type of barrier shape simplified many aspects of applying the graphical approach to the problems considered. We recommend that the extension of the methods of Chapter 3 to more general polygonal barriers be investigated.

5.2.4 Euclidean Distances With Barriers

Euclidean distance problems with barriers to travel appear to be more difficult than rectilinear distance problems because there appears to be no simple division of the plane into an equivalent of the cells used for rectilinear distance problems. We recommend that a graphical approach to weighted maximin location problems with Euclidean distances and polygonal barriers be investigated.

APPENDIX A

BARRIER TO TRAVEL EXAMPLE

In clockwise order, the 7 vertices of the feasible region are:

(1.0,20.0) (19.0,19.0) (19.0, 0.0) (13.0, 1.0) (0.0, 2.0) (0.0,10.0) (1.0,11.0).

The 30 demand points, each with a weight equal to one, are:

(7.0,10.0) (10.3, 8.5) (11.0,12.5) (13.0,13.5) (12.0, 9.5)
(13.5,11.5) (13.5, 9.0) (14.5, 8.0) (9.0, 6.5) (5.5, 6.0)
(8.5, 8.5) (10.0,17.0) (12.0,15.5) (19.0,14.0) (17.5,11.0)
(14.0, 2.0) (18.0, 5.0) (3.0, 4.0) (5.0, 2.0) (6.0, 3.0)
(2.0, 2.0) (1.5, 8.5) (3.5,15.0) (7.0,16.5) (5.0,12.0)
(4.5, 8.0) (2.0,16.0) (2.0,17.0) (6.5,12.0) (7.0,12.0).

In clockwise order, the 6 vertices of the first of four barriers to travel are:

(3.0,15.0) (4.5,15.0) (4.5, 6.0) (14.5, 6.0) (14.5, 4.0) (3.0, 4.0).

In clockwise order, the 8 vertices of the second of four barriers to travel are:

(5.0,16.5) (8.0,16.5) (8.0, 9.0) (7.5, 9.0) (7.5,13.0) (6.0,13.0) (6.0, 7.0) (5.0, 7.0).

In clockwise order, the 8 vertices of the third of four barriers to travel are:

(9.0,15.5) (17.5,15.5) (17.5, 3.5) (15.0, 3.5) (15.0,13.5) (10.0,13.5) (10.0, 9.0) (9.0, 9.0).

In clockwise order, the 8 vertices of the fourth of four barriers to travel are:

(10.5,11.5) (13.5,11.5) (13.5, 7.0) (7.5, 7.0)(7.5, 8.0) (12.5, 8.0) (12.5,10.0) (10.5,10.0).

APPENDIX B

NOXIOUS FACILITY EXAMPLE

In clockwise order, the 196 vertices of the feasible region which approximates Southern Ontario are:

(34.4,27.5)	(34.8,26.4)	(37.2,25.7)	(38.4,25.8)	(39.0,25.5)
(41.2,26.6)	(41.8,26.6)	(42.0,24.9)	(41.7,24.0)	(42.3,23.7)
(42.2,22.8)	(43.3,22.3)	(43.5,21.8)	(43.0,21.3)	(35.4,21.1)
(30.1,20.1)	(28.9,18.9)	(28.8,18.2)	(28.2,18.4)	(27.7,17.8)
(29.5,17.3)	(31.9,16.9)	(30.5,16.7)	(28.2,17.2)	(25.9,17.3)
(23.6,18.4)	(19.2,18.7)	(16.0,16.8)	(13.5,14.7)	(13.2,13.3)
(11.2,13.5)	(7.4,11.5)	(6.6,10.5)	(6.3, 8.8)	(6.0, 9.8)
(5.4,10.5)	(4.2,10.6)	(3.0,10.1)	(2.0,10.0)	(0.1,11.0)
(0.4,14.3)	(1.0,14.8)	(2.1,15.1)	(3.7,14.6)	(4.7,14.4)
(6.0,14.6)	(6.9,14.5)	(7.4,15.3)	(7.4,16.6)	(5.7,16.9)
(5.0,17.6)	(5.7,17.8)	(6.5,18.5)	(7.2,20.6)	(7.1,21.4)
(7.8,23.5)	(7.7,23.8)	(8.2,24.1)	(10.0,24.5)	(11.8,26.1)
(11.9,26.6)	(12.4,26.6)	(13.6,27.2)	(14.5,27.9)	(15.0,28.7)
(15.3,36.5)	(15.0,38.4)	(16.1,39.8)	(16.7,41.8)	(17.8,43.0)
(18.5,43.3)	(20.1,45.8)	(19.7,46.8)	(20.0,47.0)	(19.2,49.8)
(19.4,50.6)	(18.7,50.7)	(17.8,52.5)	(17.2,53.2)	(17.3,53.7)
(15.9,54.2)	(16.6,54.8)	(18.6,54.3)	(20.3,54.4)	(20.3,54.1)
(19.8,53.3)	(19.9,52.8)	(20.4,52.7)	(20.6,51.2)	(22.9,49.2)
(22.7,48.7)	(21.6,47.8)	(21.6,47.4)	(22.9,48.3)	(23.9,48.1)
(24.0,47.2)	(23.4,45.2)	(24.6,46.7)	(25.3,47.1)	(26.5,47.0)
(26.8,45.6)	(31.6,43.4)	(32.4,44.0)	(32.8,44.9)	(32.6,45.2)
(32.9,46.1)	(32.4,46.8)	(31.4,47.3)	(31.5,48.2)	(32.5,48.7)
(33.6,48.8)	(34.2,47.7)	(35.6,47.4)	(38.7,45.0)	(38.0,43.5)
(36.9,42.4)	(35.9,42.2)	(36.4,42.1)	(37.6,42.6)	(37.3,40.5)
(37.4,39.7)	(37.7,39.6)	(37.9,40.1)	(37.6,40.6)	(38.0,41.2)
(40.6,41.7)	(40.9,43.2)	(40.2,44.5)	(39.0,44.9)	(39.2,47.1)
(38.6,45.8)	(38.7,45.0)	(35.6,47.4)	(31.9,52.3)	(32.0,53.2)
(29.1,56.8)	(25.5,63.2)	(23.4,64.0)	(24.2,64.9)	(30.1,65.1)
(34.9,67.4)	(38.7,67.6)	(38.2,68.9)	(41.0,68.7)	(41.6,69.2)
(42.5,68.5)	(45.6,68.7)	(50.9,68.1)	(56.3,66.7)	(59.3,64.7)
(59.1,63.8)	(60.8,62.1)	(62.5,61.4)	(63.1,62.0)	(62.8,63.0)
(64.3,62.6)	(65.4,58.6)	(69.2,56.9)	(69.8,58.0)	(71.4,57.9)
(73.6,55.9)	(74.9,57.4)	(79.2,59.2)	(82.2,60.2)	(84.9,60.3)
(87.7,59.4)	(86.9,55.7)	(88.3,54.4)	(86.7,53.3)	(86.4,52.4)
(84.4,51.5)	(82.9,51.7)	(78.9,48.8)	(74.5,44.5)	(73.0,42.4)
(69.4,41.2)	(67.6,40.0)	(66.1,40.0)	(64.1,38.9)	(54.1,38.0)
(62.3,36.5)	(62.2,35.9)	(63.9,36.4)	(64.0,36.0)	(61.0,34.7)

(60.1,34.9) (59.2,36.1) (57.6,35.8) (56.3,37.1) (53.8,36.4)
(50.0,36.0) (47.9,35.3) (44.5,34.8) (41.7,34.0) (39.8,32.3)
(37.9,31.7).

In clockwise order, the 20 vertices of the area demand approximating Essex County are:

(7.4,11.5) (6.6,10.5) (6.3, 8.8) (6.0, 9.8) (5.4,10.5)
(4.2,10.6) (3.0,10.1) (2.0,10.0) (0.1,11.0) (0.4,14.3)
(1.0,14.8) (2.1,15.1) (3.7,14.6) (4.7,14.4) (6.0,14.6)
(7.2,14.2) (6.9,13.7) (7.1,13.4) (7.1,12.7) (6.9,12.7).

In clockwise order, the 16 vertices of the area demand approximating Hamilton-Wentworth Region are:

(34.4,27.6) (34.8,26.5) (36.1,26.1) (35.8,25.1) (35.3,25.3)
(34.8,23.9) (30.4,26.1) (29.9,27.8) (30.5,27.9) (30.3,28.7)
(31.6,28.9) (31.6,29.2) (32.0,29.6) (33.8,27.9) (33.3,27.4)
(33.6,27.0).

In clockwise order, the 9 vertices of the area demand approximating Metropolitan Toronto are:

(39.8,32.3) (37.9,31.7) (37.3,31.0) (36.5,32.1) (36.7,32.3)
(36.2,33.5) (40.9,34.9) (41.1,34.3) (41.6,33.9).

In clockwise order, the 17 vertices of the area demand approximating Ottawa-Carleton Region are:

(69.1,56.9) (69.7,58.0) (71.3,57.9) (73.6,55.8) (75.0,57.3)
(78.1,58.7) (79.3,56.3) (77.6,55.5) (78.4,53.7) (76.1,52.2)
(75.8,52.6) (75.6,52.0) (74.0,50.5) (72.5,51.9) (72.6,52.0)
(68.5,56.5) (68.9,56.9).

In clockwise order, the 45 vertices of the area demand approximating Algonquin Provincial Park are:

(55.2,65.9) (57.0,61.3) (52.6,59.4) (52.5,59.8) (51.9,59.6)
(51.9,59.4) (50.1,58.6) (49.6,57.3) (51.1,53.3) (49.7,52.8)
(48.7,55.5) (48.5,55.5) (48.4,55.7) (48.1,55.5) (47.9,55.9)
(46.8,55.5) (46.1,57.2) (43.9,56.4) (42.1,61.1) (41.8,61.0)
(41.7,61.2) (42.0,61.4) (41.5,62.9) (41.7,63.0) (41.1,64.7)
(43.4,65.6) (43.3,65.8) (44.0,66.1) (43.9,66.4) (44.4,66.6)
(44.6,66.2) (45.0,66.4) (45.3,65.8) (45.7,65.9) (45.6,66.0)
(46.0,66.2) (46.1,66.0) (47.0,66.4) (47.4,65.4) (49.0,66.0)
(49.1,65.8) (51.0,66.4) (51.4,65.3) (53.0,65.9) (53.4,65.2).

BIBLIOGRAPHY

Abhinorasaeth, P. and E. Melachrinoudis. "Locating a Facility Using Two Objectives: Maximum & Minimum," TIMS/ORSA Joint National Meeting, Boston, April, 1985.

ACM SIGGRAPH Graphics Standards Planning Committee. "Status Report of the Graphics Standards Committee," Computer Graphics, 13 (1979).

Bazaraa, M. S. and C. M. Shetty. Nonlinear Programming: Theory and Algorithms, Wiley, New York, 1979.

Belter, S. E. "Computer-Aided Routing of Printed Circuit Boards," Byte, June 1987, pp. 199-208.

Benhamou, S. and E. Melachrinoudis. "A Single Facility Rectilinear Location Problem Using Two Criteria: Minisum and Maximin," TIMS/ORSA Joint National Meeting, New Orleans, May, 1987.

Bennett, C. D., and A. Mirakhor. "Optimal Facility Location With Respect to Several Regions," Journal of Regional Science 14 (1974), pp. 131-136.

Brady, S. D. and R. E. Rosenthal. "Interactive Computer Graphical Solutions of Constrained Minimax Location Problems," AIIE Transactions 12 (1980), pp. 241-248.

Brady, S. D., R. E. Rosenthal, and D. Young. "Interactive Graphical Minimax Location of Multiple Facilities with General Constraints," IIE Transactions 15 (1983), pp. 242-254.

Cavalier, T. M. and H. D. Sherali. "Euclidean Distance Location-Allocation Problems with Uniform Demands over Convex Polygons," Transportation Science 20 (1986), pp. 107-116.

Chatelon, J., D. Hearn, and T. J. Lowe. "A Subgradient Algorithm for Certain Minimax and Minisum Problems - The Constrained Case," SIAM Journal on Control and Optimization 20 (1982), pp. 455-469.

Chein, O. and L. Steinberg. "Routing Past Unions of Disjoint Linear Barriers," Networks 13 (1983), pp. 389-398.

Church, R. L. and R. S. Garfinkel. "Locating an Obnoxious Facility on a Network," Transportation Science 12 (1978), pp. 107-118.

Cohen, B. Before It's Too Late: A Scientist's Case For Nuclear Energy, Plenum Press, New York, 1983.

Cooper, L. "Location-Allocation Problems," Operations Research, 11 (1963), pp. 331-344.

Courant, R. and H. Robbins. What Is Mathematics?, Oxford University Press, Oxford, 1941.

Dasarathy, B. and L. J. White. "A Maxmin Location Problem," Operations Research 28 (1980), pp. 1385-1401.

Dearing, P. M. and R. L. Francis. "A Network Flow Solution to a Multifacility Minimax Location Problem Involving Rectilinear Distances," Transportation Science 8 (1974), pp.126-141.

Domschke, W. and A. Drexl. Location and Layout Planning: An International Bibliography, Springer-Verlag, New York, 1985.

Drezner, Z. "On Minimax Optimization Problems," Mathematical Programming 22 (1982), pp. 227-230.

________. "Constrained Location Problems in the Plane and on a Sphere," IIE Transactions 15 (1983), pp. 300-304.

________. "Location of Regional Facilities," Naval Research Logistics Quarterly 33 (1986), pp. 523-529.

Drezner, Z. and G. O. Wesolowsky. "A Note on Optimal Facility Location With Respect to Several Regions," Journal of Regional Science 18 (1978), p. 303.

________. "Single facility lp-distance minimax location," SIAM Journal on Algebraic and Discrete Methods 1 (1980a), pp. 315-321.

________. "A Maximin Location Problem with Maximum Distance Constraints," AIIE Transactions 12 (1980b), pp. 249-252.

________. "Optimal Location of a Facility Relative to Area Demands," Naval Research Logistics Quarterly 27 (1980c), pp. 199-206.

________. "Minimax and Maximin Facility Location Problems on a Sphere," Naval Research Logistics Quarterly 30 (1983a), pp. 305-312.

________. "The Location of an Obnoxious Facility with Rectangular Distances," Journal of Regional Science 23 (1983b), pp. 241-248.

________. "Location of Multiple Obnoxious Facilities," Transportation Science 19 (1985), pp. 193-202.

________. "Location with Asymmetric Euclidean Distances," ORSA/TIMS Joint National Meeting, Miami Beach, October, 1986.

Eilon, S., C. D. T. Watson-Gandy, and N. Christofides. Distribution Management: Mathematical Modelling and Practical Analysis, Hafner, New York, 1971.

Elzinga, J. and D. W. Hearn. "Geometrical Solutions for Some Minimax Location Problems," Transportation Science 6 (1972a), pp. 379-394.

________. "The Minimum Covering Sphere Problem," Management Science, 19 (1972b), pp. 96-104.

________. "A Note on a Minimax Location Problem," Transportation Science 7 (1973), pp. 100-103.

Foley, J. D. and A. Van Dam. Fundamentals of Interactive Computer Graphics, Addison-Wesley, Reading, Massachusetts, 1982.

Francis, R. L. "Some aspects of a minimax location problem," Operations Research 15 (1967), pp. 1163-1169.

________. "A Geometrical Solution Procedure for a Rectilinear Distance Minimax Location Problem," AIIE Transactions 4 (1972), pp. 328-332.

Francis, R. L., L. F. McGinnis, and J. A. White. "Locational Analysis," European Journal of Operational Research, 12 (1983), pp. 220-252.

Francis, R. L. and J. A. White. Facility Layout and Location, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

Glasstone, S. and P. Dolan, eds. The Effects of Nuclear Weapons, 3rd ed., U.S. Department of Defense and Department of Energy, Washington, D.C., 1977.

Handler, G. Y. and P. B. Mirchandani. Location on Networks: Theory and Algorithms, The MIT Press, Cambridge, Massachusetts, 1979.

Hansen, P., D. Peeters, D. Richard, and J.-F. Thisse. "The Minisum and Minimax Location Problems Revisited," Operations Research 33 (1985), pp. 1251-1265.

Hansen, P., D. Peeters, and J.-F. Thisse. "On the location of an obnoxious facility," Sistemi Urbani 3 (1981), pp. 299-317.

________. "Public Facility Location Models: A Selective Survey," in Thisse, J. F. and H. G. Zoller (eds.), Location Analysis of Public Facilities, North Holland, Amsterdam, 1983.

Katz, I. N. and L. Cooper. "Facility Location in the Presence of Forbidden Regions," Report OREM 79011, Department of Operations Research, Southern Methodist University, July 1979a.

________. "Facility Location in the Presence of Forbidden Regions, II: Euclidean Distance and Several Forbidden Circles," Report OREM 79006, Department of Operations Research, Southern Methodist University, March 1979b.

________. "Facility location in the presence of forbidden regions, I: Formulation and the case of Euclidean distance with one forbidden circle," European Journal of Operational Research 6 (1981), pp. 166-173.

Krarup, J. and P. M. Pruza. "The simple plant location problem: Survey and synthesis," European Journal of Operational Research 12 (1983), 36-81.

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. 21-33.

Larson, R. C. and G. Sadiq. "Facility Locations with the Manhattan Metric in the Presence of Barriers to Travel," Operations Research 31 (1983), pp. 652-669.

Larson, R. C. and V. O. K. Li. "Finding Minimum Rectilinear Distance Paths in the Presence of Barriers," Networks 11 (1981), pp. 285-304.

Lee, D. T. and F. P. Preparata. "Euclidean Shortest Paths in the Presence of Rectilinear Barriers," Networks 14 (1984), pp. 393-410.

Litwhiler, D. W. "Large Region Location Problems," Ph.D. dissertation, University of Oklahoma, 1977.

Love, R. F. "A Computational Procedure For Optimally Locating A Facility With Respect To Several Rectangular Regions," Journal of Regional Science 12 (1972), pp. 233-241.

Love, R. F., J. G. Morris, and G. O. Wesolowsky. Facilities Location: Models and Methods, Publications in Operations Research Series, Elsevier Science Publishing Co., 1988.

Lozano-Perez, T. and M. Wesley. "An algorithm for planning collision-free paths among polyhedral obstacles," Comm. ACM 22 (1979), pp. 560-570.

Marucheck, A. S. and A. A. Aly. "An Efficient Algorithm for the Location-Allocation Problem with Respect to Several Rectangular Regions," Naval Research Logistics Quarterly 28 (1981), pp. 309-323.

Mehrez, A., Z. Sinuany-Stern, and A. Stulman. "The One-Dimensional Single Facility Maximin Distance Location Problem," Journal of Regional Science 23 (1983), pp. 233-239.

________. "An Enhancement of the Drezner-Wesolowsky Algorithm for Single-facility Location with Maximin of Rectilinear Distance," Journal of the Operational Research Society 37 (1986), pp. 971-977.

Melachrinoudis, E. "The MAXIMIN Location Problem Using a Euclidean Metric," Ph.D. Dissertation, University of Massachusetts, September, 1980.

Melachrinoudis, E. and T. P. Cullinane. "Locating an Undesirable Facility Within a Geographical Region Using the Maximin Criterion," Journal of Regional Science 25 (1985a), pp. 115-127.

________. "A heuristic approach to the single facility maximin location problem," International Journal of Production Research, 23 (1985b), pp. 523-532.

________. "Locating an undesirable facility with a minimax criterion," European Journal of Operational Research 24 (1986), pp. 239-246.

Minieka, E. Optimization Algorithms for Networks and Graphs, Marcel Dekker, New York, 1978.

Morris, J. G. "A Linear Programming Approach to the Solution of Constrained Multi-Facility Minimax Location Problems where Distances are Rectangular," Operational Research Quarterly 24 (1973), pp. 419-435.

Muralimohan, R. and A. J. G. Babu. "Facility Location in the Presence of Convex Forbidden Regions," TIMS/ORSA Joint National Meeting, Chicago, April, 1983a.

Muralimohan, R. and A. J. G. Babu. "Mathematical Modeling of the Weber Problem in the Presence of Two Convex Nonintersecting Forbidden Regions--Application to 2 Ellipses," ORSA/TIMS Joint National Meeting, Orlando, Florida, November, 1983b.

Oommen, B. J. "An Efficient Geometrical Solution to the Minimum Spanning Circle Problem," Operations Research 35 (1987), pp. 80-86.

Paroush, J. and C. S. Tapiero. "Optimal Location of a Polluting Plant on a Line Under Uncertainty," Journal of Regional Science 16 (1976), pp. 365-374.

Plastria, F. "Solving general continuous single facility location problems by cutting planes," European Journal of Operational Research 29 (1987), pp. 98-110.

Ravindranath, K., P. Vrat, and N. Singh. "Bicriteria Single FacilityRectilinear Location Problems in the Presence of a Single Forbidden Region," Opsearch 22 (1985), pp. 1-16.

Rogers, C. A. Packing and Covering, Cambridge University Press, Cambridge, 1964.

Shamos, M. I., and D. Hoey. "Closest-Point Problems," 16th Annual IEEE Symposium on Foundations of Computer Science, (1975), pp. 151-162.

Smallwood, R. D. "Minimax Detection Station Placement," Operations Research 13 (1965), pp. 632-646.

Viegas, J. and P. Hansen. "Finding shortest paths in the plane in the presence of barriers to travel (for any lp-norm)," European Journal of Operational Research 20 (1985), pp. 373-381.

Wangdahl, G. E., S. M. Pollock, and J. B. Woodward. "Minimum-Trajectory Pipe Routing," Journal of Ship Research 18 (1974), pp. 46-49.

Weiszfeld, E. "Sur le Point pour Lequel la Somme des Distances de n Points Donnes est Minimum," Tohoku Mathematics Journal, 43 (1936), pp. 355-386.

Wesolowsky, G. O. "Rectangular Distance Location under the Minimax Optimality Criterion," Transportation Science 6 (1972), pp. 103-113.

Wesolowsky, G. O. and R. F. Love. "Location of Facilities with Rectangular Distances Among Point and Area Destinations," Naval Research Logistics Quarterly, 18 (1971), pp. 83-90.