

**ON THE OPTIMAL LOCATION OF TRANSMITTERS
FOR
MICRO-CELLULAR RADIO COMMUNICATION SYSTEM DESIGN**

by

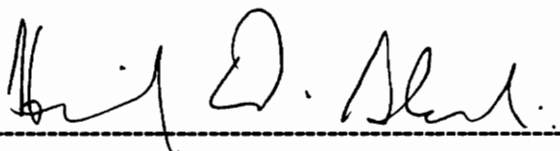
Chandra Mohan Pendyala

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

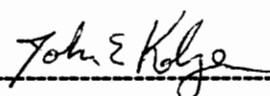
Master of Science

in

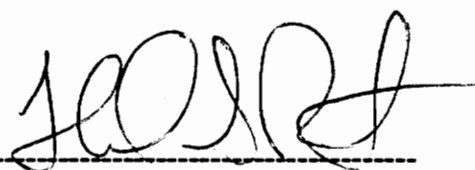
Industrial and Systems Engineering



Dr. Hanif D. Sherali



Dr. John Kobza



Dr. Theodore S. Rappaport

**1st July 1994
Blacksburg, Virginia.**

LD
5655
V855
1994
P463
c.2

On The Optimal Location of Transmitters for Micro-Cellular Radio Communication

System Design.

by

Chandra M. Pendyala

Committee Chairman Dr. Hanif D. Sherali

Industrial & Systems Engineering

(ABSTRACT)

This research aims at solving an engineering design problem encountered in the field of wireless communication systems using mathematical programming techniques. The problem addressed is an indispensable part of micro-cellular radio system design. It involves an optimal location of radio transmitters, given a distribution of receivers and desired signal characteristics. The study has been conducted with the intent of making this problem an integral part of a CAD system for designing radio communication systems. The tool that has been developed for locating a transmitter in such a context is sensitive to different needs of coverage at different locations in the design space.

The physical nature of this problem enables it to be conceptualized as a traditional facility location problem. The transmitter is a service facility responsible for serving all the receivers in the design space. A cost is incurred in terms of path-loss, delay spread, and other separation based measures, whenever service is extended to a receiver. The objective is to place this transmitter in such a way that it optimally serves all the receivers, as measured according to some merit function. However, the nature of the latter merit or objective function, and the nature of the acceptable region for transmitter placement, imparts a special structure to the problem that distinguishes it from traditional facility location problems. The aim of this research effort is to construct a suitable representative mathematical model for this problem, and to design and compare various solution methodologies that are computationally competitive, numerically stable, and accurate.

Dedicated

To my Advisor, Parents, Sister and Brother

Acknowledgments

I wish to express my very deep felt gratitude to the chairman of my advisory committee Dr. Hanif D. Sherali. His guidance, exceptional teaching capabilities, abundant patience and inspiration have enhanced my engineering, academic and communication skills tremendously. I would also like to thank the other members of my committee, Dr. John Kobza and Dr. Theodore S. Rappaport for their friendship, encouragement and inputs to my research. I wish to thank Dr. J.D.Tew for his friendship and inputs. I thank the faculty and the administration at Department of Industrial and Systems Engineering, Virginia Tech for providing me with an environment of constant encouragement for graduate education. I thank the Mobile and Portable Radio Research Group for providing me access to their excellent research facilities.

The research for my master's degree thesis in the Department of Industrial and Systems Engineering has been supported by the Advanced Research Projects Agency, and I thank them for their support. I also wish to thank the College of Engineering for recognizing this research with the "Paul E. Torgersen Graduate Research Excellence Award".

I wish to thank my parents, Krishna Rao and Lakshmi Pendyala, my sister Sharmila Pendyala, my brother Santhi Kiran Pendyala and my friends Veeru, Sunitha, Kalyan and Durga for their encouragement and support all through my graduate education.

Table of Contents

Abstract	ii
Dedication	iii
Acknowledgments	iv
Table of Contents	v
1.0 Introduction	1
2.0 Modelling of the Problem	5
2.1 Single Transmitter Location	5
2.2 Multiple Transmitter Problem	9
3.0 Literature Review	12
3.1 Classification of Location-Allocation Problems	12
3.2 Response Surface Methodology	14
3.3 Nonlinear Programming Techniques	15
4.0 Solution Methodologies	17
4.1 Initial Solution	17
4.2 Solution Procedure	17
4.2.1 Line-search Strategies	21
4.2.2 Improving Directions - Hooke and Jeeves Method	23
4.2.3 Conjugate gradient Methods	25
4.2.4. Quasi-Newton Methods	28
4.2.5 Response Surface Methodology	30
4.3 Multiple Transmitter Problem	34
5.0 Test Problems	38
6.0 Implementation and Results	41
6.1 Effect of Model Objective Functions and Parameters	41

6.2 Implementation Guidelines and Comparisons	43
6.2.1 Line-search	44
6.2.2 Hooke and Jeeves Method	45
6.2.3 Conjugate gradient Method	48
6.2.4 Quasi-Newton Method	53
6.3 Multiple Transmitter Problem	55
7.0 Conclusions & Future Research	76
8.0 References	80
Vita	83

Chapter 1

Introduction

The design of a micro-cellular radio communication system for a given area involves the partitioning of the design area into cells, and the location of transmitters within these cells. The considerations that influence this process are the distribution of receivers (frequency reuse), the topology of the design space, and the goal of improving the utilization of the available resources. Engineering considerations such as the feasibility of locating a transmitter in a given area, and the requirement to provide service of a given quality over the entire design space, also influence the cell division and transmitter location process.

Present procedures to accomplish such a task are not analytical and hence, very often, not optimal. The problem under consideration is to approximately partition the given area of service into cells and to optimally locate transmitters in each cell so that the signal in each cell is of sufficient intensity. Also, a good cell division and a good transmitter location will result in using lesser energy for the transmitter (base station), and hence, can result in lower co-channel interference and in lower costs.

The tool developed for locating a transmitter and designing the cells in such a context must be sensitive to different needs of coverage at different locations in the design space. For example, a region with higher receiver traffic has a greater need for better and higher intensity of coverage. The model should represent the entire design space and ensure that the signal strength at all points in the design space is above a given threshold. Also, the acceptable region for locating transmitters might be a strict subset of the design space.

A model that takes into consideration all the above constraints, meets all the stated demands of the engineer, and produces an optimal solution to the problem, will help in designing an effective communication systems that is reliable and provides a good quality of service. This is the intent of the present research effort. The aim is to construct a suitable representative mathematical model for the foregoing problem, and to design solution methodologies that are computationally competitive, numerically stable, and accurate.

A further discussion of the design of a micro-cellular communication system is given below. This discussion aims at placing the above mentioned research in the context of an overall system design.

Given the topography (outdoor communications) or the building plan (indoor communications), a band of frequencies, and a probable distribution of receivers (receiver density as a function of location), the goal of a system designer is to design a communication system that is reliable and provides all the receivers with clear and sufficiently strong signals. Since the available resources have to effectively serve all the receivers in the design space, an optimal use of the band of frequencies (or the security codes in the case of CDMA) is an essential part of this design process.

A typical communication system partitions the design area into smaller areas to be covered by a transmitter (base station). All the transmitters are linked to a central switching station that controls the flow of calls, frequency (or code) allocations, and manages the identification and tracking of received calls. In reality, the range of frequencies (or codes) is so narrow that very often, many of the frequencies (or codes) have to be simultaneously used by more than one pair of customers. To ensure adequate performance, the frequency allocation should be designed to ensure that any two transmitters using the same frequency are sufficiently separated in distance from each other so as to avoid interference.

Mobile customers and the load on the bands available for each cell change dynamically and the possibility of overloading cells should be taken into consideration during these situations. The cell sizes, and their traffic should also be considered while designing the system. The sequencing of the calls is another problem that can affect the efficiency of the communication system.

An incoming call is first queued, the design space is scanned for the receiver, and a frequency is allocated to service the call. Given an area to be covered (indoor or outdoor), the essential components needed to design a communication system are the hardware, a well optimized call sequencing technique, an effective receiver tracking and locating system, an optimal procedure that designs the shapes and sizes of cells, an optimal transmitter location in every cell, an effective method of call hand-off (call hand-off is the situation arising due to the movement of a customer from one cell to another, leading to a switch-over of the call assignment), and an effective frequency allocation procedure.

This chapter has so far introduced the physics of the problem under consideration and discussed it in the context of an overall system design. The next chapter describes the development of a suitable mathematical model that reflects the above mentioned requirements of the communication system designer. The description of the modelling process starts with the construction of a mathematical model for the location of a single transmitter, given a distribution of receivers in a cell, and a given propagation prediction procedure that measures the adequacy of the signal strength at the different receiver locations, depending on the topology of the cell. The single transmitter model is then extended to the case of locating multiple transmitters, along with the ensuing problem of partitioning the design space into cells and optimally locating the transmitters therein.

Chapter 3.0 is a review of the available literature on facility location problems. The difficulty involved in solving the present problem within the framework of the identified exist-

ing models and procedures is also discussed. Modifications made to various existing nonlinear programming procedures to suitably redesign them for the structure of the present problem are then explained in detail.

Next we present a description of the test problems used to illustrate and validate the model and the prescribed solution procedures. These test problems include a special realistic model for the second floor of Whittemore Hall of Virginia Polytechnic Institute and State University, along with a list of standard test problems intended to analyze the performance of the designed solution procedures.

Following this, we provide a detailed implementation of the various proposed solution procedures, along with computational results on the performance of the proposed methods using the selected test problems. Results for the multiple transmitter problem are also provided.

Finally, a concluding chapter summarizes the research contributions and suggests ideas for further research in this field.

Chapter 2

Modelling of the Problem

We begin this section by developing a mathematical model for the single transmitter problem, and then describe extensions of this model to handle the case of multiple transmitters.

2.1 Single Transmitter Location Problem: The task of optimally locating a single transmitter, given a distribution of receiver points, can be conceptualized as follows. Suppose that the receiver locations are distributed over a designated region in a Euclidean space, and that we need to determine a point, in some restricted subset of the region, at which to locate a particular transmitter. The transmitter location should be sensitive to the coverage needs at the different receiver locations over the design space. Given any location for the transmitter, we can use available propagation prediction techniques to estimate the quality and intensity of coverage over the given area. The purpose of the optimization model and algorithm is to manipulate the transmitter location in order to determine a “best” location site, as measured by a suitable merit or objective function. Hence, the design objective is to provide a coverage of required intensity (minimum power-loss) to the entire design space under consideration and the design or decision variables are the coordinates of the transmitter location.

The problem of numerically representing the quality and intensity of coverage over the design space as a function of the transmitter location can be addressed as follows. The design space is first partitioned into a grid of possible receiver locations. The density of this grid is selected to represent with a desired accuracy. In fact, as we do, the grid density

can be dynamically adjusted to strike a trade-off between the accuracy and computational complexity of the algorithm as the optimization proceeds. More specifically, we commence with a coarse grid, and then sequentially refine the grid to fine tune the location of a transmitter as the algorithm proceeds. Propagation prediction techniques [Seidel and Rapaport (1991)] are used to evaluate the coverage at each of these grid points. The values of these coverage measures are weighted using factors proportional to priorities or the relative importances of different areas in the design space (based on the probable distribution of receivers.) The weighted values of coverage can be used in one of the following forms to construct a merit or objective function that measures the coverage as a function of transmitter location. (Since the coverage is being measured in terms of path-losses (inverse of signal strength), the objective functions defined below are to be minimized.)

MINISUM OBJECTIVE FUNCTION: The sum of all the weighted path-losses or lack of coverage measures is minimized over the design area for this approach. (The path-loss for each receiver can be computed using developed propagation prediction methods, and can be very conveniently implemented using parallel processing.) The optimization of this function with respect to the transmitter location tends to improve the overall coverage. The drawback with this objective function is that it might ignore a few remotely located receiver points, while achieving a good quality of total weighted coverage.

MINIMAX OBJECTIVE FUNCTION: The maximum of all the weighted path-losses or lack of coverage measures is minimized over the design area for this approach. The optimization of this function ensures that even the worst receiver location enjoys an acceptable weighted coverage. The drawback with this model is that it concentrates on the worst-case situation, at the expense of the overall average weighted coverage.

CONVEX COMBINATION OF MINISUM AND MINIMAX OBJECTIVE FUNCTIONS: A convex combination [Bazaraa, Sherali, and Shetty (1993)] of the two foregoing

objective functions can be used to reflect their relative merits, while overcoming their respective shortcomings. The optimization of such a function with respect to the transmitter location can give a good coverage over the entire design space, while maintaining an acceptable coverage at remotely located receivers. The convex combination parameter can be adjusted interactively by the system designer, based on the path-loss profiles over the design space. It should be noted that the minisum is the sum of m weighted path-loss values. This might result in a numerical domination over the minimax objective. Hence the minisum objective is scaled by averaging it using the m values before the convex combination is computed.

The mathematical model accommodates two types of design constraints. The first type of constraint requires that the quality and intensity of coverage at each receiver location over the design space must be above a given threshold value. This constraint is treated as a “soft” constraint, and is incorporated into the model via a penalty term [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)] in the objective function. The penalty term is designed to contribute an added cost factor in the objective function whenever the threshold path-loss is exceeded. This concept can be extended to include more than one break point (different penalty rates as the path-loss crosses different thresholds) in the objective function. The second type of constraint requires the location of the transmitter to be restricted to certain acceptable subsets of the design space. This is treated as a “hard” constraint. We will assume that the region of location is a hyperrectangle in \mathbb{R}^3 . In some cases depending on the structural layout of the design space and the physics and geometry of the propagation prediction technique used, a constrained optimization problem of the type developed here can be solved using each of these hyperrectangles in turn as feasible regions. A comparison of the resulting solutions can be made to prescribe an overall optimum.

The problem can now be mathematically written as,

$$\text{Minimize } f(x, y, z) = \Psi f_1(x, y, z) + (1 - \Psi) f_2(x, y, z) \quad (2.1)$$

Subject to $0 \leq x \leq h_1, 0 \leq y \leq h_2, 0 \leq z \leq h_3$

where $f_1(x, y, z)$ is the minisum objective function given by

$$f_1(x, y, z) = \frac{1}{m} \sum_{i=1}^m w_i \left[g_i(x, y, z) + \mu_i \max \{0, g_i(x, y, z) - s_i\} \right] \quad (2.2)$$

and $f_2(x, y, z)$ is the minimax objective function given by

$$f_2(x, y, z) = \text{Maximum}_{i=1, \dots, m} \left\{ w_i \left[g_i(x, y, z) + \mu_i \max \{0, g_i(x, y, z) - s_i\} \right] \right\} \quad (2.3)$$

and where

$$g_i(x, y, z) = f_a F_i(x, y, z) + c \log [\max(1, D_i(x, y, z))] \text{ for all } i = 1, \dots, m. \quad (2.4)$$

In the above equations, the following notation has been used:

$Q = \{ (x, y, z) \in R^3 \}$ such that $0 \leq x \leq h_1, 0 \leq y \leq h_2, 0 \leq z \leq h_3$ = the feasible hyper rectangle

$i=1, \dots, m$: indices for the given receiver locations

w_i = relative priority weight ascribed to the i^{th} receiver location

$g_i(x, y, z)$ = path-loss at the i^{th} receiver location for a given transmitter location (x, y, z)

s_i = maximum allowable path-loss at the receiver location i , for $i=1, \dots, m$.

μ_i = suitable penalty factor for violating the prescribed path-loss threshold at receiver location i .

$\Psi \in [0, 1]$ = convex combination used to design the objective function

In the formula for computing g_i , the following notation has been used:

f_a = wall-attenuation factor

c = path-loss exponent

$F_i(x, y, z)$ = number of walls separating the i^{th} receiver location and the transmitter when the latter is placed at the point (x, y, z) .

$D_i(x, y, z)$ = distance between the i^{th} receiver location and the transmitter placed at a point (x, y, z)

2.2 Multiple Transmitter Problem: The problem of optimally locating multiple transmitters operating on non-interfering (parallel) frequencies has two components to it, allocation and location. Each transmitter located has to service a subset of the total design space. The mathematical model represents the design space as a grid of receiver locations having assigned priority weights. Hence, each transmitter to be located has to service a set of pertinent receiver locations. The actual allocation of receivers to each transmitter depends on the existing relative locations of the transmitters. Given such an allocation, the corresponding optimal location of each transmitter with respect to the receivers assigned to it is the second integral part of the problem. Naturally, these location and allocation sub-problems need to be coordinated simultaneously in order to determine an overall optimum.

The mathematical model built to represent the multiple transmitter problem has to measure the extent to which the design objectives have been achieved by the set of transmitters. The model should be sensitive both to the location of the transmitters and to the allocations of the receivers to each of the transmitters.

The ideas used in the modelling of the single transmitter problem can be extended to this problem as follows. The objective function used is a sum of all the individual contributions of each of the transmitters toward the design goal. The contribution of each of the

transmitters is computed by allocating it to the receivers that enjoy the strongest signal from it among all the transmitters. Obviously, the allocation scheme is dependent on the location of each transmitter. Hence, the idea is to locate the transmitters such that the ensuing allocation scheme will provide the best overall measure of service to the set of receivers.

The problem can now be written as follows.

Let $j = 1, \dots, n$: indices given for the transmitters to be located.

$Q = \{ (X, Y, Z) \in R^3 \}$ such that $0 \leq X \leq h_1, 0 \leq Y \leq h_2, 0 \leq Z \leq h_3$ = the feasible hyper rectangle

w_i = relative priority weight ascribed to the i^{th} receiver location

$g_{ij}(x_j, y_j, z_j)$ = path-loss at the i^{th} receiver location for a given j^{th} transmitter located at (x_j, y_j, z_j)

s_i = maximum allowable path-loss at the i^{th} receiver location.

μ_i = suitable penalty factor for violating the prescribed path-loss threshold at receiver location i .

S_j = set of all receivers allocated to the transmitter j .

ALLOCATION PROBLEM (for a given set of transmitter locations):

Let, $p_i(X, Y, Z)$ = the path-loss at the i^{th} receiver location defined as

$$p_i(X, Y, Z) = \min_j (g_{ij}(x_j, y_j, z_j)) \quad \forall \quad i = 1, \dots, m$$

Hence, effectively, given the transmitter locations, each receiver i is allocated to

$\text{argmin}_j \{g_{ij}(x_j, y_j, z_j)\}$ with ties broken arbitrarily. Consequently, this determines the sets

S_j , where $\bigcup_{j=1}^n S_j = \{1, \dots, m\}$ and $S_{j_1} \cap S_{j_2} = \emptyset \quad \forall (j_1 \neq j_2)$.

OVERALL MULTIPLE TRANSMITTER LOCATION-ALLOCATION MODEL:

With the usual notation, the problem is to

$$\text{Minimize } f(X, Y, Z) = \Psi f_1(X, Y, Z) + (1 - \Psi) f_2(X, Y, Z) \quad (2.5)$$

Subject to $0 \leq x_j \leq h_1, 0 \leq y_j \leq h_2, 0 \leq z_j \leq h_3$ for all $j=1, \dots, n$.

where, $\mathbf{X} = (x_1, \dots, x_n)$, $\mathbf{Y} = (y_1, \dots, y_n)$, $\mathbf{Z} = (z_1, \dots, z_n)$, and where,

$$f_1(X, Y, Z) = \frac{1}{m} \cdot \sum_{i=1}^m w_i \left[p_i(X, Y, Z) + \mu_i \max \{0, p_i(X, Y, Z) - s_i\} \right] \quad (2.6)$$

and

$$f_2(X, Y, Z) = \frac{1}{n} \cdot \sum_{j=1}^n \text{Maximum}_{i \in S_j} w_i \left[p_i(X, Y, Z) + \mu_i \max \{0, p_i(X, Y, Z) - s_i\} \right] \quad (2.7a)$$

Note that, similar to the single transmitter case the objective function is a convex combination of the minisum and minmax objective functions f_1 and f_2 . The function f_1 represents the average of all path-losses at the receiver locations. The function f_2 represents the average of the path-losses of the worst served receiver over all the transmitters. Alternatively, if desired, the second objective function can be modified to measure the worst service over the entire design space as follows

$$f_2(X, Y, Z) = \text{Maximum}_{i=1, \dots, m} w_i \left[p_i(X, Y, Z) + \mu_i \max \{0, p_i(X, Y, Z) - s_i\} \right] \quad (2.7b)$$

We now proceed to discuss the framework in which the foregoing problem can be cast, and present a brief description of the available related literature.

Chapter 3

Literature Review

In the foregoing discussion, we have mentioned certain similarities and differences between traditional facility location-allocation problems and the present problem under consideration. This chapter aims at providing a brief overview and a classification of traditional facility location-allocation problems. It also provides a discussion of solution procedures that have been considered for solving these problems. The aim of this chapter is to put the present research in perspective with existing literature.

3.1 Classification of Location-Allocation Problems: Facility location and allocation and layout problems have drawn a great deal of research interest and this has resulted in a vast library of literature discussing these problems. Relevant discussion on such problems can be found in the book by Francis, McGinnis and White (1992) based on a variety of criteria. Handler and Mirchandani (1979) discuss the corresponding problem when the location space is a network. Tansel, Francis, and Lowe (1983) provide a more recent survey on the same topic. Nordai (1985) has attempted to classify location-allocation problems in his doctoral dissertation and provides an extensive literature review, and Tuncbilek's (1990) masters thesis presents an update on this survey. We refer the reader to these surveys for details on this subject matter. A conceptual problem classification scheme is given below.

1) **SOLUTION SPACE**: The location problem can be classified based on the permissible feasible region for locating the service facilities. The feasible region can be continuous or discrete. In the case of a continuous solution space, the variables describing the possible

locations of the facilities are defined over a continuous set or a union of such subsets. In the case of a discrete solution space, the possible locations of the facilities are restricted to a set of discrete points in a plane or on a network. The solution space is further restricted in most cases by other side-constraints as well. The present problem under consideration is a continuous problem.

2) **MEASURE OF DISTANCE**: The optimality of the location of a facility is often a function of its distance from other facilities or from the demand locations. The nature of this measure of distance is dictated by the modelling of the problem. The frequently used measures are rectilinear, Euclidean / squared Euclidean, the general l_p norm [Francis, McGinnis, and White (1992), Nordai (1985)], or, in the case of network-based facility location problems, shortest paths in the underlying network. The present problem under consideration uses a function of Euclidean distances for the model. It should be noted that often, a function of these distance measures is more pertinent to the problem than the distances themselves. The complexity and the nature of the distance functions is often the influencing factor in designing suitable solution procedures, and determines the degree of analytical tractability of the problem.

3) **OPTIMALITY CRITERION**: The physical nature of the problem under consideration often determines the choice of a suitable optimality criterion. The frequently used optimality criteria are the minimax, and the minisum objective functions, and a convex combination of these criteria. In modelling the present problem, (Chapter 2), a convex combination of the minimax and minisum objectives has been used.

4) **DEMAND LOCATIONS**: The modelling of a problem is often dependent on the nature of the locations of the demand that the facility has to serve. The demand locations can be discrete or might be continuously distributed over some region or network, sometimes represented by a probability distribution. The present problem is inherently a continuous

demand problem, but in modelling the problem, we have discretized the demand for reasons of computational tractability.

In addition to the above aspects of the problem, a variety of other classifications can be made, such as the number of facilities to be located, i.e., a single facility problem or a multiple facility problem. In the case of multiple facility problems, some instances have the number of facilities as a decision variable, while other instances use the number of facilities as a specified parameter. Other classifications are based on the nature of demand (static or dynamic, probabilistic or deterministic), or based on the physical significance of the problem such as competitive or non-competitive.

The major research thrust in this thesis, aside from modelling the problem at hand, is to design computationally competitive, numerically stable and accurate solution procedures for the proposed model. Linear programming, integer programming, nonlinear programming, network flows and dynamic programming techniques are the popular approaches for solving facility location problems, depending on their structure. The present instance of the problem is a nonlinear programming problem and we have specially designed search algorithms to solve the problem. Response Surface Methodology has also been investigated with the intent of dealing with future stochastic propagation prediction techniques. The rest of this chapter reviews the solution procedures considered for the proposed of a problem.

3.2 Response Surface Methodology: Response Surface Methodology RSM is a simulation-optimization [Box and Draper (1987), Khurii and Cornell (1987), Raymond (1976), Sheldon (1989), William and Tamer (1987)] procedure. The present problem under consideration, as modelled in the previous chapter, is a complex problem since it is not easy to analyze it using traditional deterministic procedures. Response Surface Methodology inherently does not make any assumptions about the nature of the function. It is designed

to handle situations arising because of randomness in simulation [Box and Draper (1987), Khuri and Cornell (1987), Raymond (1976)]. These characteristics make RSM an attractive procedure to solve the present problem if the propagation prediction procedure is stochastic.

In the present instance of the problem, the coordinates of the transmitter location are the decision variables and the response (objective function value) is a function of these decision variables. Response Surface Methodology starts with a good initial solution, approximates the function by a polynomial (using experimental design and regression), and uses a search strategy based on this approximation to identify an optimum. Typically, the vicinity of a stationary solution (gradient zero) is identified, and then a higher degree polynomial (more accurate design) is used to either more accurately characterize this local optimum, or to analytically describe the vicinity of this best solution found.

The general applicability and the robustness to deal with randomness is a very attractive feature of Response Surface Methodology, but this technique has traditionally used weak search algorithms [Sheldon (1978), William and Tamer (1987)] in terms of computational competitiveness. There is an exponential increase in the number of functional evaluations with an increase in the number of [Raymond (1976)], and Hence, the algorithm is computationally very expensive. If the interest of the designer lies solely in finding an optimal solution and not in describing the vicinity of the optimum, RSM tends to use more computations [Sheldon (1989)] for no extra benefit. This has prompted an interest in the exploration of more effective nonlinear programming techniques to improve the computational competitiveness of RSM, than have been heretofore employed in this context.

3.3 Nonlinear Programming Techniques: These techniques include various derivative-free and derivative-based, single-dimensional and multi-dimensional, unconstrained optimization procedures [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)]. Con-

strained optimization problems are also frequently solved as a sequence of unconstrained optimization problems using suitable penalty functions.

The unconstrained optimization procedures referred to above guarantee convergence to a stationary point under mild assumptions [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)]. Some of the procedures offer convergence in a finite number of steps (same as the number of variables) if the objective function is quadratic, and many of the procedures offer superlinear rates of convergence. This feature is worthwhile to pursue, although Response Surface Methodology techniques addressed in the foregoing subsection have largely ignored it. (See Joshi, Sherali, and Tew (1993) for an enhancement in RSM emerging from this view point.)

In practical implementation contexts, many of these procedures can be easily extended to cases that do not satisfy the strict assumptions required to guarantee convergence. Modifications of this kind with appropriate termination criteria and gradient evaluation procedures have shown great promise in solving the present problem in a very stable and competitive fashion. The following chapter will elaborate on these solution procedures, their modifications, and the redesign of these procedures for the present problem. This research effort is therefore expected to contribute toward improving the performance of the search algorithms in the context of optimizing complex functions whose behavior is largely unknown, and where the evaluation of the objective function itself is computationally expensive.

Chapter 4

Solution Methodologies

The following sections of this chapter describe the solution procedures proposed for the single transmitter problem. The multiple transmitter location-allocation problem will be considered subsequently.

4.1 Initial Solution: The performance and convergence characteristics of optimization algorithms are usually very sensitive to the starting solution used. An initial point in the close vicinity of the optimum can reduce computations drastically, and accelerate the convergence of the algorithm. Moreover, for nonconvex problems, a good initial solution can promote the convergence toward a global optimum rather than a local optimum. In our context, the minimization of the sum of weighted squared Euclidean distances [Francis, McGinnis, and White (1992)] of the receiver locations from the transmitter location can be used to obtain an initial point. This approach yields a center of gravity solution in closed-form, and in our computational experiments, the resulting starting solution has demonstrated a significant advantage in terms of computational savings and convergence properties for all the different algorithms.

4.2 Solution Procedure: The proposed search algorithms solve the problem by starting with a good quality initial feasible solution, and then iteratively, they generate an improving direction, identify an optimal step length along this direction, and hence arrive at a new solution [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)]. This process continues until a suitable termination criterion is satisfied. The idea is to obtain the final solu-

tion after having evaluated the fewest number of alternative transmitter locations along the way.

As mentioned in the chapter on problem modelling, the problem is to be solved with box constraints. This involves redesigning the above mentioned unconstrained procedures to incorporate the box constraints. These constraints require two algorithmic modifications. The first modification is required when the procedure used for developing the improving directions generates a direction that is infeasible to the box constraints. This problem is easily solved either by projecting the direction onto the box to obtain a feasible direction, or by projecting the point resulting after taking a step length along this direction back onto the box to obtain a feasible solution. We will adopt the former approach.

Given a point, an improving feasible direction and the corresponding box constraints, the maximum allowable step-length along the direction before the box constraints are violated is easily computed. This is used to limit the step-length that is prescribed by a suitable unconstrained line-search technique. The value of this maximum allowable step-length λ_{max} from a point $(x, y, z) = q = (q_1, q_2, q_3)$ along a direction $d = (d_1, d_2, d_3)$ is given by,

$$\lambda_{max} = \min (\lambda_1, \lambda_2, \lambda_3)$$

where,

$$\lambda_i = \begin{bmatrix} \infty, & \text{if } (d_i = 0) \\ \frac{q_i}{(-d_i)}, & \text{if } (d_i < 0) \\ \left(\frac{h_i - q_i}{d_i} \right), & \text{if } (d_i > 0) \end{bmatrix}$$

(4.8)

Another problem encountered in the design of solution procedures is the evaluation of the gradient at a given point. Since the objective function is not analytically differentiable, the following techniques are employed to approximate the gradient at any given point.

FINITE DIFFERENCES: The gradient of the function at the point q is computed using one of the following two techniques. Let q_j be the j th component of the point q , let δ be a parameter, and let ξ_j be the j th component of the gradient being evaluated at the point q . The first technique employs *forward differences* and computes

$$\xi_j = \frac{f(q_j + \delta) - f(q_j)}{\delta} \quad \forall j = 1, \dots, n \quad (4.9)$$

The second technique employs *central differences* and computes

$$\xi_j = \frac{f(q_j + \delta) - f(q_j - \delta)}{2\delta} \quad \forall j = 1, \dots, n \quad (4.10)$$

Note that the forward difference method requires $(n+1)$ functional evaluations, while the central difference method requires $2n$ functional evaluations, while perhaps yielding more reliable estimates.

FIRST-ORDER MODEL: This technique uses suitable functional evaluations to fit a good first-order model [Box and Draper (1987), Khuri and Cornell (1987), Raymond (1976)], and then differentiates this function to estimate the gradient. The first-order model employed is a linear function whose parameters are determined via a least-squares fit based on the functional evaluations conducted at the points shown by asterisks below. (Here (x, y, z) represents the current iterate.) The particular eight points selected below in Figure (4.1) are desirable in that they give uncorrelated estimates for the parameters.

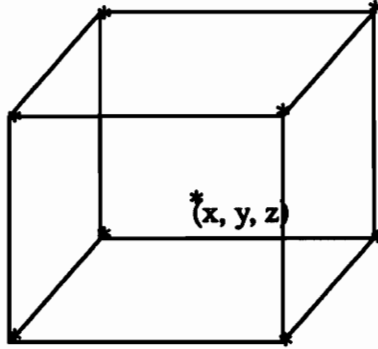


Figure (4.1); Function Evaluations for a First Order Fit.

Initial experiments seem to indicate that the accuracy of all the foregoing methods is comparable, although the finite forward differencing method uses the fewest functional evaluations. This becomes more critical as the number of variables increases, as well as when the number of iterations of the algorithm increases. Hence, this is the technique that we will adopt in our implementation.

The following sections elaborate on details regarding the implementation of the various generic steps used by the nonlinear search procedures. As mentioned above, the proposed search algorithms solve the problem by starting with a good quality initial feasible point, and then iteratively, they generate an improving direction, identify a step-length along this direction, and hence arrive at a new solution [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)]. This process continues until a suitable termination criterion is satisfied. The rest of this chapter elaborates on the theoretical motivation for each of the methodologies used in the above steps. Also presented are the assumptions made in each case, the modifications made to these procedures, along with the motivation thereof, to accommodate the violations of the assumptions, the theoretical rates of convergence of these procedures, and their convergence properties are presented below.

4.2.1 Line-Search Strategies: This part of the algorithm involves identifying a suitable step length to take along any given improving direction, in order to determine a new iterate. Research efforts on this topic are focussed on comparing the computational competitiveness of the exact and inexact line-searches described below [Bazaraa, Sherali, and Shetty (1993)].

EXACT QUADRATIC INTERPOLATION LINE-SEARCH WITH BOX CONSTRAINTS: This method iteratively approximates the function along a search direction by a univariate quadratic function, and then minimizes this approximation to determine a revised step-length. Specifically, the procedure starts at an initial point, reads the algorithmic parameters, and achieves a Three Point Pattern (TPP). A Three Point Pattern (TPP) [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)] is a set of three points such that for step-lengths $0 \leq \lambda_1 < \lambda_2 < \lambda_3$, the corresponding function values satisfy $f_1 \geq f_2 \leq f_3$. In order to achieve a TPP, an initial step-length is taken to determine two of these points and a search is conducted to identify the third point, revising the first two points in the process if necessary. Initially the maximum allowable step-length in the direction of search is computed, and this is used to enforce the box constraints. A quadratic function is fit onto the three points thus obtained and this function is minimized to identify a new step-length. The function value at this newly identified step length is compared with the existing TPP, to obtain a revised three point pattern in a manner that reduces the interval of uncertainty. The process is terminated when the interval of uncertainty is sufficiently small. The step-length that yields the least objective function value at termination is taken as the optimal step-length. The algorithm requires one functional evaluation per iteration of the TPP search, and one per iteration of interpolation.

This procedure is theoretically sound in the case of minimizing continuous and strictly quasi-convex functions, but is widely used even in cases when such assumptions might be violated.

INEXACT QUADRATIC INTERPOLATION LINE-SEARCH WITH BOX CONSTRAINTS: The inexact version of the quadratic interpolation technique [Bazaraa, Sherali, and Shetty (1993)] is based on the same principle as the exact version, except that it terminates after an initial quadratic interpolation, unlike the exact line-search which proceeds on to reduce the interval of uncertainty to more refined levels. If the quadratic approximation of the objective function is accurate enough, the first interpolation technique gives a good estimate for the optimal step-length, and hence, further functional evaluations that attempt to reduce the interval of uncertainty can be avoided. In the vicinity of the optimal solution for the problem, a quadratic approximation of the function is a reasonable approximation to make, and this provides a principal motivation for this approach.

An inexact line-search is computationally inexpensive in comparison with the exact search but it might affect the convergence of the overall algorithm if a sufficient degree of descent is not achieved. In the present instance of the problem, the use of a good initial solution (see Section 4.2) makes the inexact procedure reasonably accurate. Computational experience has shown that the inexact version of the quadratic interpolation line-search exhibits an overall computational advantage over the exact version of the line-search in solving these problems.

INEXACT PRESCRIBED STEP-LENGTH RULE WITH BOX CONSTRAINTS: This procedure aims at prescribing a step-length [Choi and Sherali (1992)] along the given direction based on the functional evaluation at the current iterate, the adopted direction of motion, and a target value that estimates the optimum objective function value. An initial target value is selected and step-lengths are prescribed to identify the point nearest to the

target along the given direction. If this search prescribes a step-length that results in an infeasible point to the box constraints, the point is projected [Bazaraa, Sherali, and Shetty (1993)] back onto the feasible region. Now, a new direction is developed based on methods used to identify improving directions, and the procedure is repeated until an acceptable termination criterion is met. As the algorithm proceeds in this fashion, and more information regarding the optimum value is revealed, the target value used in the prescribed step-size formula is dynamically updated.

This procedure is very flexible and does not use any line-searches, and can accommodate many different direction finding routines. Hence, it can be expected to be very computationally competitive. Many true-life problems can provide a good estimate of the final target based on the physics of the problem, and this can enhance the computational capabilities tremendously.

The procedure has been tried with a few conjugate gradient and memory-less quasi-Newton [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)] direction generation procedures (detailed later). The availability of a good initial solution and the structure of the function have made inexact quadratic interpolation a very competitive procedure, as compared to this prescribed step-length technique. The experience with the inexact fixed prescribed step-length procedure has led us to infer that this procedure is better suited to solving problems having a greater number of variables (the present single transmitter problem is a three variable problem).

4.2.2 Improving Directions - Hooke and Jeeves' Method: This algorithm is a derivative free multidimensional search technique that is theoretically sound if the set over which the search is being conducted is compact [Bazaraa, Sherali, and Shetty (1993)]. (Additionally the convergence requirements of the line-search have to be met.) It should also be noted that the version of the algorithm described in the foregoing section is a modified version of

the discrete step version of the original Hooke and Jeeve's (1961) algorithm. This algorithm generates two types of directions, namely, exploratory directions and acceleration directions, in order to produce revised iterates. The box constraints of the problem are built into the line-search employed along the generated directions.

The exploratory directions are a set of orthogonal directions (coordinate axes parallel to the edges of the feasible hyperrectangular region). A fixed step-length along each of these directions is used to identify if the direction is improving. The negative extensions of these directions are also explored if the positive extensions are non-improving. After an improving direction has been identified, a suitable step length along this direction is identified to yield a new point, and the process continues along the next exploratory direction. The performance of the algorithm is considerably improved by using the step-length found for the earlier iteration as an initial trial step at the current iteration.

The second type of direction employed is a pattern search or acceleration direction, and this is used after the exploratory searches have been conducted. This pattern direction is determined by joining the initial point of the current iteration to the final point obtained after conducting the exploratory searches. An optimal step-length is identified along this direction (after projecting onto the feasible region) and the algorithm moves to the resulting iterate. This pattern direction is related to the conjugate direction strategy described next.

A pictorial representation of this procedure is depicted in two dimensions in Figure (4.2). The illustrated iteration starts at a point (x, y) , takes a step along each of the exploratory directions (axes), and then takes a step along the acceleration direction (pattern-search) toward the optimum.

The algorithm proceeds in this fashion until a termination criterion indicates that a (near) optimum has been identified. This termination criterion is based on the improvement between consecutive iterations being within a prescribed tolerance.

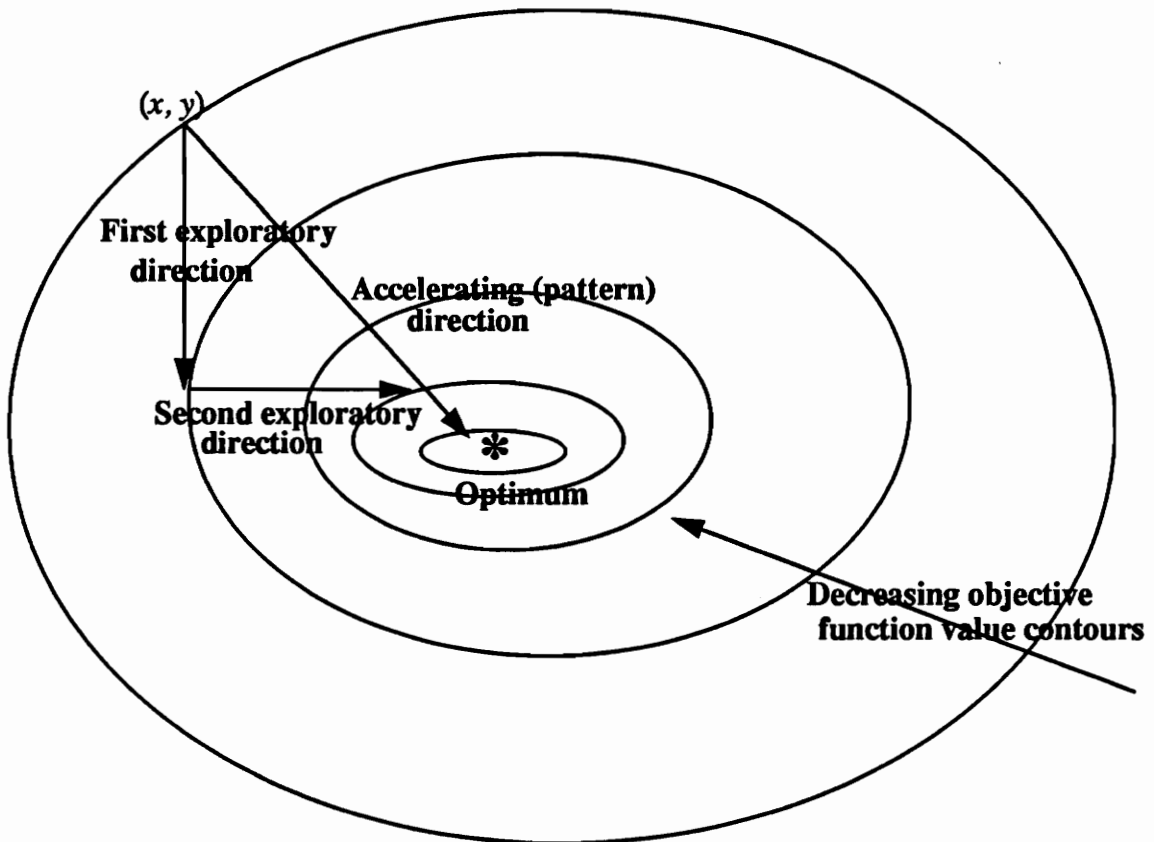


Figure (4.2); Hooke and Jeeve's Search.

4.2.3 Conjugate gradient Method: Conjugate gradient methods [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)] are derivative-based conjugate direction or gradient deflection methods that have an improved convergence behavior as compared with the steepest descent method, and need less memory than a Newton or a quasi-Newton method. If the quadratic approximation of a function is reasonably accurate, a conjugate gradient method converges in at most n iterations, where n is the number of independent variables.

Often, for convergence purposes in nonquadratic situations, such algorithms are restarted with steepest descent direction steps finitely often.

In our context, since the objective function is not differentiable, and theoretical extensions of conjugate gradient methods exist for even sub-differentiable functions (See Camerini et al., 1973, and Sherali and Ulular, 1998), we decided to investigate this technique in comparison with the quasi-Newton procedure, although we have a low dimensional problem to solve in the present context.

The algorithm starts at an initial point and performs a line-search in the negative gradient direction (for a minimization problem). This line-search results in a new iterate. The direction of search from this point is determined by deflecting the current negative gradient direction with the previous direction of search, using an additive deflection parameter. The deflecting factors are derived based on functional and gradient evaluations. The process repeats after conducting a line-search, until a suitable termination criterion similar to that used for Hooke and Jeeves' method is satisfied.

A few of the deflection strategies used compute the direction d_k at iteration k according to

$$d_k = -\xi_k + \beta_k d_{k-1} \quad (4.11)$$

where,

x_k is the present iterate, and x_{k+1} is the iterate obtained following a line-search that yields a prescribed step-length of λ_k along the direction d_k . This direction of search at the present iteration is obtained by deflecting the antigradient $-\xi_k$ at the present iteration with the previous direction using an additive deflection factor β_k .

One choice of the deflection factor can be derived by simply bisecting the angle between the present gradient and the previous anti-gradient. This is called the average direction strategy (ADS) [Sherali and Ulular(1989)] and is given by,

$$\frac{\|\xi_k\|}{\|d_{k-1}\|} \quad (4.12)$$

An alternative choice derived from the quadratic approximation of the function using a Hessian conjugacy requirement for consecutive directions and assuming inexact line-searches but using quasi-Newton types of updates [see Shanno (1978), Sherali and Ulular(1990)] is given by

$$\frac{\xi_k^t \sigma_k - \left(\frac{1}{\lambda_{k-1}}\right) \xi_k^t \rho_{k-1}}{d_{k-1}^t \sigma_k} \quad (4.13)$$

Here, $\sigma_k = (\xi_k - \xi_{k-1})$ and $\rho_k = (x_k - x_{k-1})$. The derivation of this deflection parameter recognizes the inaccuracies in the line-searches, and attempts to approximate the Newton-based direction at x_k using d_{k-1} and ξ_k . Note that a combination of the above two deflection strategies can be used.

Note also that the problem under consideration is not an unconstrained optimization problem. The constrained nature of the problem creates situations where infeasible directions might result. If an updated direction is infeasible, it is projected onto the feasible region before conducting the line-search. If this projected direction is null, the algorithm is restarted using the projection of the negative gradient direction. The well known Karush-Kuhn-Tucker optimality conditions [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)] are used to assert that if the projected negative gradient is also null, then under the assumption of local pseudo-convexity of the objective function, the current point is a local optimum, and hence, the procedure can be terminated. A restart [Powell (1977),

Sherali and Ulular (1990)] of the algorithm after every n iterations, or based on the progress attained, significantly improves the rate of convergence.

A pictorial representation of the conjugate gradient method is presented in Figure (4.3) in two dimensions. The algorithm starts at a point (x_1, y_1) and moves along the negative gradient direction to arrive at the next iterate (x_2, y_2) . At this point, the revised negative gradient direction is deflected using the previous direction of motion. The procedure then moves along the deflected direction from the point (x_2, y_2) . If the function being optimized is quadratic in this two dimensional illustration, this latter direction would directly lead to the optimum.

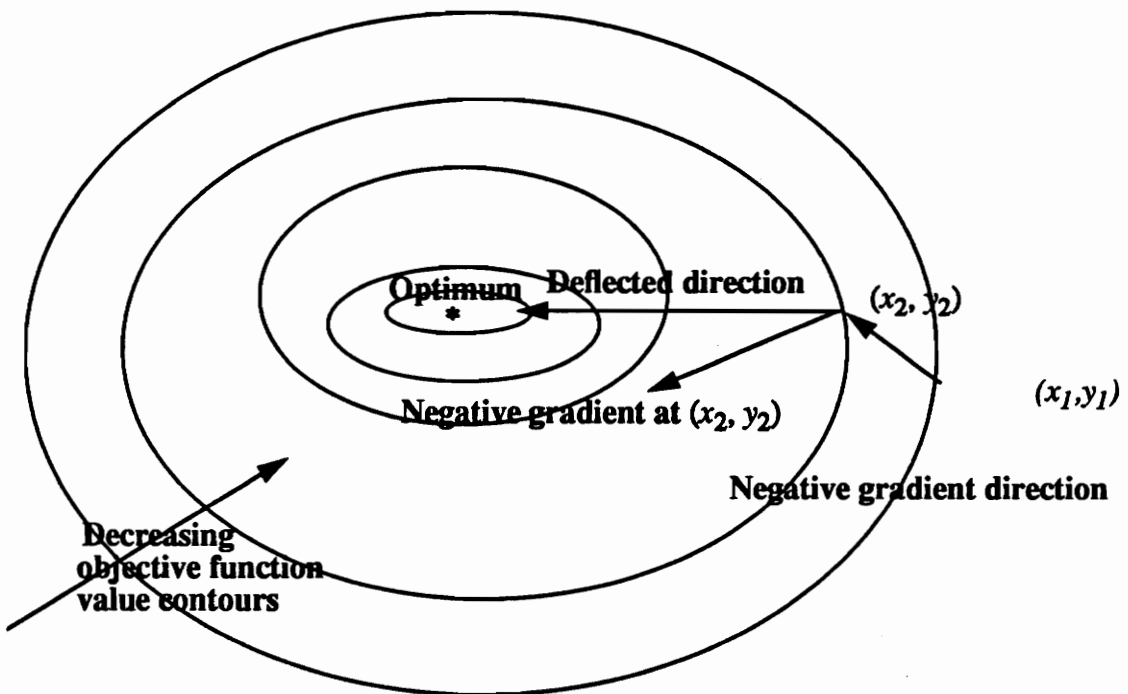


Figure (4.3); Conjugate Gradient Search.

4.2.4 Quasi-Newton Method: These are conjugate direction methods that employ a multiplicative deflection strategy. Their motivation lies in constructing a series of approxima-

tions that converge (in at most n iterations for the quadratic case) to the actual (inverse) Hessian of the objective function, hence effectively producing a multi-step Newton direction [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)].

A brief description of the Newton method and the Levenberg-Marquardt modification of the Newton method [Bazaraa, Sherali, and Shetty (1993)] are described below to motivate the quasi-Newton methods.

The Newton method deflects the negative gradient of the function by pre-multiplying it with the Hessian inverse. This direction ensures convergence to the local optimal in one unit step if the function is quadratic, and if the Hessian is positive definite. For nonquadratic functions, if the iterate is initialized close enough to a local minimum, and if the Hessian at the solution is positive definite, and satisfies certain boundedness and regularity conditions, the sequence of iterates generated by taking such steps iteratively converges to this local minimum at an order-two (superlinear) rate of convergence. The problem encountered with this method is that the Hessian may not be nonsingular or positive definite. To resolve this difficulty, a modification of this method solves this problem by deflecting the negative gradient with

$$B = (\epsilon I + H)^{-1} \tag{4.14}$$

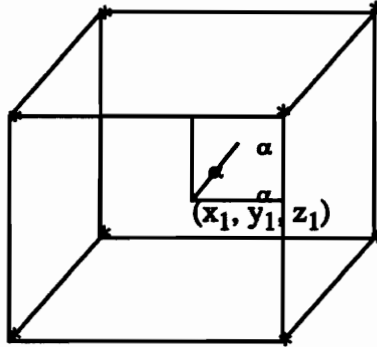
where H is the Hessian and ϵ is chosen to ensure that all the eigenvalues of B are greater than or equal to some $\delta > 0$. This feature makes sure that B is positive definite and invertible. Assuming that the search is contained in a compact set, and the algorithm is reset to use the antigradient direction finitely often, the convergence of the algorithm can be claimed because of the convergence of the steepest descent method, along with the fact that the procedure generates a sequence of improving directions. It should be noted that the algorithm is very sensitive to the choice of ϵ and δ . As the value of δ increases, it forces an increase in the value of ϵ and reduces the rate of convergence to order-one. The

smaller the value of ϵ , the closer the method resembles the Newton method and hence tends to an order-two convergence. The method dynamically updates the value of ϵ using a comparison of how closely a quadratic approximation describes the actual function.

A natural extension of the above ideas incorporating the idea of conjugacy results in the quasi-Newton methods [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)]. The negative gradient at each iteration is deflected using a positive definite deflection matrix D_k that represents the current approximation of the Hessian inverse. The idea is to sequentially update D_k to D_{k+1} at iteration k , by accumulating the additional linearly independent current search direction step value as an eigenvector of the product $D_{k+1}H$ having a unit eigenvalue, where H is the Hessian of the (assumed) quadratic objective function. In this case, the Hessian inverse approximation is obtained at the end of n iterations. As above, direct modifications are possible to accommodate nonquadratic functions or to account for the inaccuracies due to inexact line-searches. The idea of periodic restarts enhances the convergence of these algorithms. The path adopted by the algorithm looks similar to that of the conjugate gradient algorithm, and usually has faster convergence than the conjugate gradient algorithm.

4.2.5 Response Surface Methodology: This technique uses functional evaluations to fit response surfaces and uses an appropriate optimization technique [William and Tamer (1987)] to identify an optimal solution. The details of this method can be summarized as follows.

The algorithm starts with an initial point and a parameter α . A 2^k factorial first-order model [Box and Draper (1987), Khuri and Cornell (1987), Raymond (1976)] is fit around the initial point (x_1, y_1, z_1) , using a hypercube of side 2α as shown below, where the asterisks represent the points at which functional evaluations are made.



Function Evaluations for a First Order Fit.

The response is given by

$$\eta = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z + \varepsilon \quad (4.15)$$

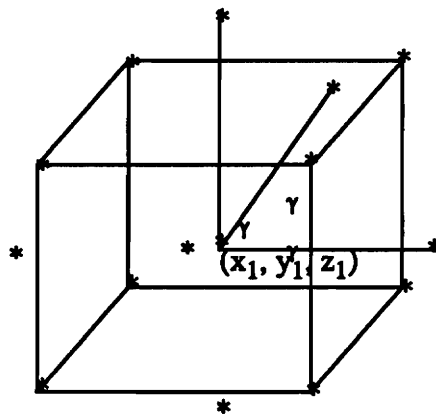
where ε represents the error with respect to the first-order approximation. Let $\beta = [\beta_0, \beta_1, \beta_2, \beta_3]'$ denote the vector of function parameters, $R = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6, \eta_7, \eta_8]'$ denote the vector of responses, and let us translate the axis to the center of the cube and scale it by a factor of α to simplify the calculations. Hence the design matrix is given by

$$D = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Now, using $\eta = D\beta + \varepsilon$, where ε is a vector of errors, when $\eta = R$ and the variable values are given via D , we can determine a least-squares estimate $\hat{\beta}$ for β by minimizing $\|e\|^2$. Since $(D'D)^{-1} = I_4/8$, where I_4 is an identity matrix of size 4, we obtain $\hat{\beta} = (D'R)/8$ and moreover, the coefficients of this design are uncorrelated, i.e., their covariance is zero. The

above design for the first-order model is orthogonal, hence it is a minimum variance design. Using this linear model, a search is conducted in an improving direction using any of the nonlinear programming techniques where the gradient is given by β , leading to a new point. If the new point is inside the cube of dimension 2α , this indicates the presence of a curvature; otherwise, the value of α is updated by expanding it by a multiplicative factor. After a few iterations of fitting a first-order model, a curvature will be detected (when the algorithm has moved into a close neighborhood of the optimum). The curvature is then taken into consideration and a central composite design is used to fit a second-order model.

The functional evaluations needed for this second-order model can be represented as follows, where the asterisks represent the points at which the functional evaluations are made.



Function Evaluations for a Second Order Fit.

The response is now given by

$$\eta = \beta_0 + \beta_1x + \beta_2y + \beta_3z + \beta_{11}(x^2 - \theta_x) + \beta_{22}(y^2 - \theta_y) + \beta_{33}(z^2 - \theta_z) + \beta_{12}xy + \beta_{23}yz + \beta_{13}xz \quad (4.16)$$

where θ_x , θ_y , and θ_z are, respectively, the means of the squares of the x , y , and z coordinates of the points indicated by asterisks above [Raymond (1976)]. The response

uses mean corrected variables in order to enable a diagonalization of the design matrix. As before,

$$\text{let } \beta = [\beta_0, \beta_1, \beta_2, \beta_3, \beta_{11}, \beta_{22}, \beta_{33}, \beta_{12}, \beta_{23}, \beta_{13}]'$$

denote the vector of function parameters, let

$$R = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6, \eta_7, \eta_8, \eta_9, \eta_{10}, \eta_{11}, \eta_{12}, \eta_{13}, \eta_{14}, \eta_{15}]'$$

denote the vector of responses, and let us translate the axis to the central point to simplify the calculations. Hence the design matrix is given by

$$D = \begin{bmatrix} 1 & -1 & -1 & -1 & 1-c & 1-c & 1-c & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1-c & 1-c & 1-c & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1-c & 1-c & 1-c & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1-c & 1-c & 1-c & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1-c & 1-c & 1-c & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1-c & 1-c & 1-c & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1-c & 1-c & 1-c & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1-c & 1-c & 1-c & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & -c & -c & -c & 0 & 0 & 0 \\ 1 & -\gamma & 0 & 0 & \gamma^2 - c & -c & -c & 0 & 0 & 0 \\ 1 & \gamma & 0 & 0 & \gamma^2 - c & -c & -c & 0 & 0 & 0 \\ 1 & 0 & -\gamma & 0 & -c & \gamma^2 - c & -c & 0 & 0 & 0 \\ 1 & 0 & \gamma & 0 & -c & \gamma^2 - c & -c & 0 & 0 & 0 \\ 1 & 0 & 0 & -\gamma & -c & -c & \gamma^2 - c & 0 & 0 & 0 \\ 1 & 0 & 0 & \gamma & -c & -c & \gamma^2 - c & 0 & 0 & 0 \end{bmatrix}$$

(4.17)

where $c=7(8+2\gamma^2)$ for a three variable response. The above design can be made orthogonal by an appropriate choice of γ (1.2 for a three variable problem). Now a least-squares estimate $\hat{\beta}$ for β can be computed by minimizing the square of errors. This yields $\hat{\beta} = (D'D)^{-1}D'R$. Also since $(D'D)^{-1}$ is diagonal, the coefficients of this design are uncorrelated, i.e., their covariance is zero. The above design for the second-order model is orthogonal, hence, it is a minimum variance design.

Having obtained this second-order model, we use it to analytically derive an improving direction (or solution). Termination is identified when the decrease in the response over two consecutive iterations is less than the allowable error.

The number of functional evaluations needed, the necessity for uncorrelated estimates of the parameters [Box and Draper (1987), Khuri and Cornell (1987), Raymond (1976)], and the advantages of having a zero variance model motivates the selection of the above mentioned designs for the first-order and second-order models. This solution procedure has been recently enhanced by Joshi, Sherali, and Tew(1993) using a variety of memory-less quasi-Newton based and conjugate gradient based search procedures to improve its computational competitiveness.

Having discussed the single transmitter problem, we now move on to discuss extensions of the single transmitter problem solution methodologies for solving the multiple transmitter problem.

4.3 Multiple Transmitter Problem: The solution procedure starts with an initial allocation and location and goes through iterations of improvement to approach an optimum. A schematic description of the procedure is presented in Figure (4.4).

INITIAL SOLUTION: The aim of this scheme is to allocate the receivers to the given number of transmitters as evenly as possible (in terms of priority weights) and to optimally (as close to it as possible) locate the transmitters with respect to their assigned receivers using a limited computational effort.

The initial solution starts with the given feasible hyperrectangle, an initial grid, and priority weights for these grid points. This hyperrectangle is divided into 2 hyperrectangles by

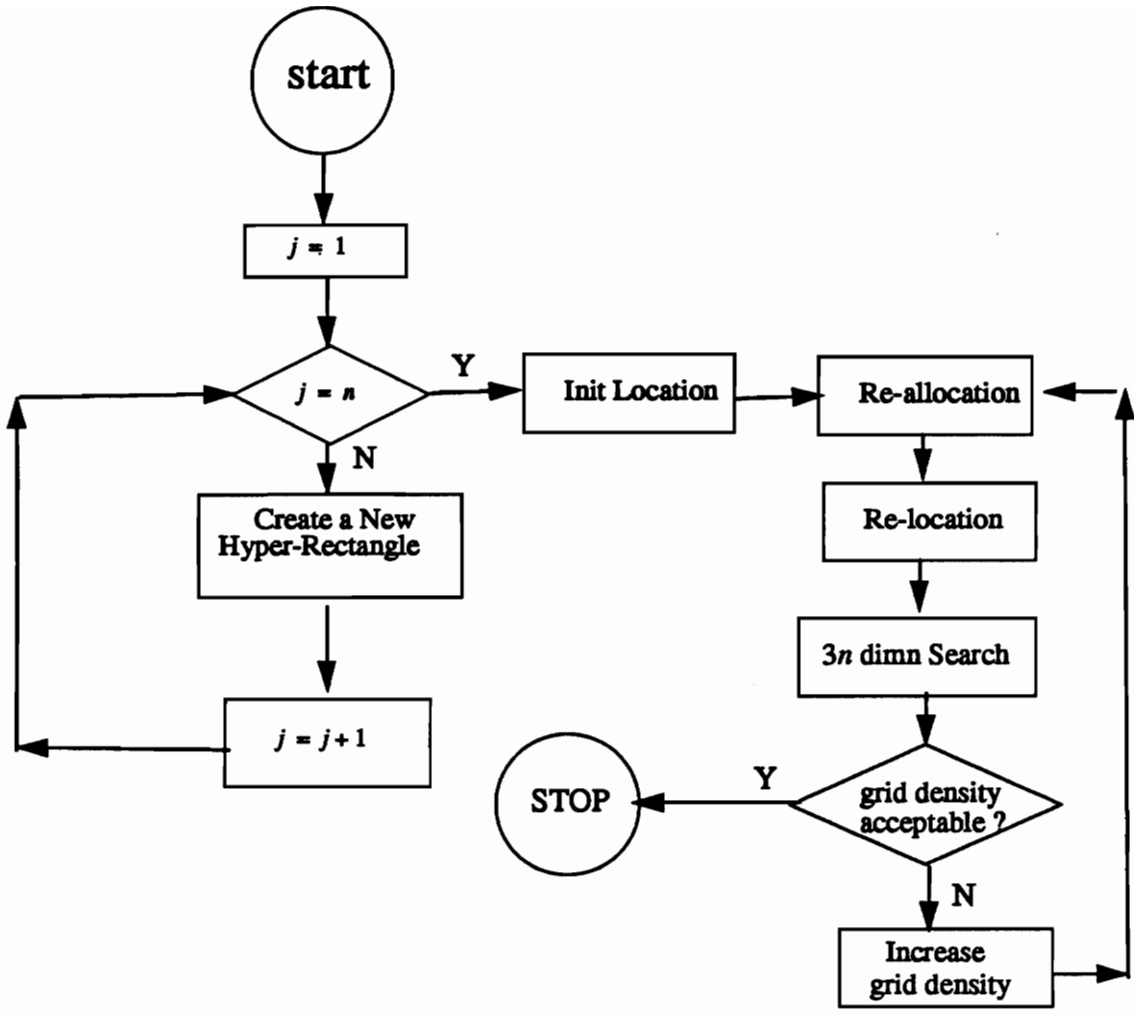
partitioning its longest dimension at the center of gravity point (solution for the squared Euclidean distance minimization problem) with respect to the priority weights. The procedure now continues by iteratively partitioning the hyperrectangle having the highest summation of weights (with ties broken arbitrarily) along its longest dimension at the center of gravity point, until n such hyperrectangles result. The initial transmitter locations are obtained by placing each of the transmitters at the center of gravity of each of the partitioned sub hyperrectangles, by solving the corresponding squared Euclidean distance minimization problem.

REFINING THE INITIAL SOLUTION: Given the above locations of the n transmitters, we first compute the values of $p_i(X, Y, Z)$, the path-loss at each of the grid points $i=1, \dots, m$, with respect to its allocated transmitter. Next we identify, the set of candidates R that are eligible for reallocation as follows.

$$R = \{ i: (p_i(X, Y, Z) > s_i) \text{ or } D_{i,j} > D_{i,min} \} \quad (4.18)$$

where, $D_{i,j}$ is the Euclidean distance from the receiver i to the transmitter j to which it is presently allocated and $D_{i,min}$ is the Euclidean distance from the receiver i to the transmitter that is closest to it. For each receiver in the set R , the six nearest transmitters, excluding the transmitter to which it is presently allocated (four in the case of two-dimensional region problems) are identified. The path-loss values for each of these receivers with respect to the above identified transmitters is computed. These receivers are now reallocated to the transmitter that provides the best coverage. This completes the reallocation phase. The location of each transmitter is now refined by solving the squared Euclidean distance minimization problem for the revised allocation. Alternatively, the solution can be enhanced by solving n single transmitter problems using their present receiver assignments. Our choice of the former scheme is motivated by the desire to produce a quick, reasonably good quality, starting solution. Also, for the same reason, we do not iterate between this location and allocation scheme until a fixed point results, although this is another viable option.

SEARCH TECHNIQUE: Given the initial solution, the search techniques developed to solve the single transmitter problem are used to solve the $3n$ variable problem ($2n$ in the case of 2-dimensional region problems) stated by Equations 2.5, 2.6 and 2.7. Here, it should be noted that we adopt the strategy of solving a sequence of problems, each having a further refined grid representation. Hence, at the beginning of each iteration, i.e., after the $3n$ dimensional problem has been solved to a certain degree of accuracy using the present grid, the grid density is increased and the new grid points are assigned based on the refinement procedure described above. Also, during the computation of the objective function, the path-losses are determined based on the reallocation procedure. This reallocation procedure involves identifying the set R using Equation (4.18), computing the path-loss values at each receiver location in this set R with respect to the six nearest transmitters (four in the case of two-dimensional region problems), excluding its present allocation, and then reassigning each receiver to the transmitter that provides the best coverage. Optimality is declared when the nonlinear search procedure terminates after solving the problem having the highest required grid density. This final grid density is a user specified option, based on the designer's desired accuracy.



Figure(4.4); Schematic Representation of the Multiple Transmitter Problem Solution Procedure.

Chapter 5

Test Problems

This chapter documents the test problems used to check the validity of the model and to compare the computational competitiveness, numerical stability and accuracy of the proposed solution procedures.

The test problems used in this research can be classified into two categories. The first category is the variants of a model of the second floor of the Whittemore Hall of Virginia Tech. This model has been used to check the sensitivity of the model to various objective function parameters and the performance of the solution procedures in solving a real-life problem (single and multiple transmitter problems).

The second category of test problems is a set of standard test problems used in the literature to study the performance of various nonlinear search procedures. This tests the applicability of the present solution procedures on a variety of objective function structures, and provides us with information on the performance of these procedures when future research produces better (or different) propagation prediction techniques.

WHITTEMORE SECOND FLOOR MODEL:

An AutoCAD map of the building was available and the data from this map was used to create a database for the locations and descriptions of the walls on the second floor. A grid of pertinent receivers having defined grid density was constructed. A propagation prediction routine using the Rappaport - Seidel d^n [Seidel, S.Y., and Rappaport, T.S., 1991]

model was used to compute the path-losses. Given the location of a transmitter, the model begins by creating a grid of pertinent receivers and calls the propagation prediction routine for every receiver location. The propagation prediction procedure counts the number of walls blocking the line of sight between the transmitter and receiver, using the database containing the information regarding the walls. The model then uses the path-loss values at each of the receivers to evaluate the objective functions described earlier. The parameters describing the weights at each receiver location, the maximum allowable path-loss at each receiver location, the penalty term, and the convex combination weight used to combine the minimax and minisum objectives are all user defined.

The present model uses a two dimensional propagation prediction procedure, and hence, is a $2n$ dimensional problem.

STANDARD TEST PROBLEMS AND STARTING SOLUTIONS: The following functions [Sherali and Ullular(1990)] have been used to test the solution procedures.

1. Witte and Holst's Strait function: Starting solution (-1.2,1.0);

Final Objective Function Value: 0.

$$\left(x_2 - x_1^2\right)^2 + 100(1 - x_1)^2 \quad (5.1)$$

2. Witte and Holst's Cube function: Starting solution (-1.2,1.0)

Final Objective Function Value: 0.

$$100\left(x_2 - x_1^3\right)^2 + (1 - x_1)^2 \quad (5.2)$$

3. C. F. Wood's Function: Starting Solution (-3, -1, -3, -1)

Final Objective Function Value: 0.

$$100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 + 90\left(x_4 - x_3^2\right)^2 + (1 - x_3)^2 + 10.1(x_2 - 1)^2 +$$

$$10.1 (x_4 - 1)^2 + 19.8 (x_2 - 1) (x_4 - 1) \quad (5.3)$$

4. Powell's function: Starting solution (-3, -1, 0, 1)

Final Objective Function Value: 0.

$$(x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 \quad (5.4)$$

5. Witte and Holst's Shallow function: Starting solution (-3,-1.0)

Final Objective Function Value: 0.

$$(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (5.5)$$

6. Rosenbrock's function: Starting solution (-1.2, 1.0, -1.2, 1.0,...)

Final Objective Function Value: 0.

$$\sum_{i=2}^n 100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2 \quad (5.6)$$

7. Oren's Power Function: Starting solution (1,1,1,...,1)

Final Objective Function Value: 0.

$$(x^t Ax)^2, A = \text{diag}(1, 2, \dots, n) \quad (5.7)$$

Chapter 6

Implementation Guidelines and Results

This chapter documents the experimentation conducted using the solution procedures on the test problems described in the previous chapter. Additionally, as mentioned in the description of the solution methodologies, the conventional search techniques have been modified for solving the present problem. This chapter also discusses the implementation of these modifications made to the solution procedures, and compares the numerical stability, computational competitiveness and empirical convergence properties of the various solution procedures. The effect of various modelling parameters on the solution and their physical significance are also analyzed. This chapter also illustrates the implementation of the modelling procedure described in section 2.2. for solving multiple transmitter problems.

6.1 Effect of model objectives and parameters: The first set of results analyze the design of the objective function. As elaborated earlier, the objective function is a convex combination of a term that takes the worst covered receiver into account (minimax), and another term that measures the average coverage for all the receivers in the problem (minisum). Figure 6.1 illustrates the optimal location of a transmitter in the case when the minimax objective is used alone. The weightage given to the receivers along the top wall have been gradually increased (1.0, 1.5, 2.0) to test the response of the solution when a particular set of receivers demands more coverage. It should be noted that as the weights were increased, the worst-covered receiver, situated at one end of the building drew the transmitter very close to itself (represented by 1, 2, and 3 for weights of 1.0, 1.5, 2.0, respectively). Figure 6.2 performs the same experimentation using the minisum objective

function alone. In this case, the transmitter moves closer to the entire line of receivers that have been weighted. A small movement of the transmitter toward the worst-covered receiver can also be noticed because of the penalty term. These experiments show that the two terms in the objective function are performing exactly the purpose they were expected to perform. Figure 6.3 shows the effect of the convex combination solution on the above experiment.

The other modelling parameters that can be experimented with are the penalty term, the threshold path-loss value and the wall-attenuation factors in the propagation prediction function. Increasing (or decreasing) the penalty parameter and the threshold parameter exhibits the same effect on the solution as increasing the weight of the particular receiver (or lowering its weight). The model that has been constructed for this problem provides for a direct physical interpretation for the parameters used therein, thereby making the model more designer friendly. The increase in the values of wall-attenuation factors makes the discontinuities in the objective function due to sudden changes in the number of intervening walls more prominent, hence increasing the chance of stalling at a nonoptimal solution. This difficulty is somewhat mitigated by generating a good initial solution.

Experimentation with various wall-attenuation factors demonstrates the sensitivity of the model to the physical structure of the design space. Figure 6.4 illustrates the design space (location of walls) considered in the solution. Figure 6.5 demonstrates the effect of wall-attenuation factors on the solution. L_1 and $L_{2.5}$ represent the optimal transmitter location as the wall-attenuation factors (path-loss effect of intervening walls) are increased from 2.73 to 13.65 in five steps. It should be noted that only the minimax objective function has been used and as the wall-attenuation factors increase, the optimum is being drawn toward the receiver that is experiencing the highest path-loss. After a certain threshold value of the wall-attenuation factor, a further increase does not change the solution because the contribution of the wall-attenuation in comparison with the contribution because of distance becomes dominantly large. These results also exhibit the sensitivity of the modelling procedure and the solution methodology to the propagation prediction procedure used.

6.2 Implementations and comparisons: The solution procedures described in this thesis have been designed in the literature primarily as unconstrained search techniques. (Some modifications for handling constraints exist; see the Notes and References section, Chapter 8, Bazzara, Sherali & Shetty, 1993.) The gradient based techniques make the additional assumption that the gradient of the objective function can be computed. These search techniques usually make additional assumptions on the structure of the objective function for convergence arguments (elaborated earlier). In the chapter on solution methodologies, the theory of the techniques developed to relax the above assumptions has been described. This chapter describes in detail a step by step implementation of the solution procedures and their modifications. The issues of numerical stability, computational competitiveness and convergence are highlighted while presenting this description.

The present problem does not satisfy any of the theoretical convergence assumptions required by any of the proposed solution procedures. The global optimality of the solution obtained through the above procedures cannot be theoretically proven, but the following explanation is given to justify that the solution obtained by the proposed procedures is likely to be a near global optimal solution. The aspects of the solution procedure that promote achieving global optimality are 1) initial solution scheme, 2) grid size variation, and 3) termination criterion.

INITIAL SOLUTION: The initial solution takes into account the weights of the grid points and their locations, and solves a related location problem. Assuming that the design space is fairly symmetrical in the propagation properties (a reasonable assumption to make in the case of a planned building or a planned city), this solution is already in the close vicinity of an optimum. Hence, the subsequent search is conducted in the close vicinity of an optimum, and hence good convergence properties for the proposed algorithms can be expected.

GRID SIZE: The solution procedure starts with a sparse grid and refines the grid size as the solution proceeds. When the grid size is sparse, the design space is represented in lesser detail and hence the procedure solves a relaxation of the problem that is less complicated, and moreover, the variations in the objective function are better smoothed. This encourages the solution during initial iterations to move into the vicinity of an optimum. As the grid density is subsequently increased, the solution gets accordingly refined more accurately toward an optimum.

In the instance of the present problems, the dimensions of the Whittemore Second floor Hall are 247 X 100 ft. (rounded to the nearest ft.). In the implementation of the grid size variation, the starting grid density was 5 X 5 i.e., one receiver per 49.4 ft. along the length of the building and one receiver per 20 ft. along the width of the building. The grid density increases in three steps with every two direction generations to (10X5), (10X10) and finally to (20 X10) i.e. one receiver per 12.35 ft. along the length and one receiver per 10 ft. along the width.

TERMINATION: The termination criterion designed for this problem (in addition to first-order type optimality conditions) is based on successive improvements over the iterations. The advantage with this approach is that further exploration of the design space is aborted if the procedure fails to improve significantly. This is important because the procedure might crawl in the vicinity of the stationary solution, while achieving it only in the limit. The details of the criterion are elaborated in the following sections.

6.2.1 Line-search: A theoretical description of the exact and inexact quadratic interpolation line-searches and the prescribed step-length procedure using variable target values, have been described in Chapter 4. A justification for the use of inexact quadratic interpolation for solving this problem has also been provided.

A schematic representation of the implementation is given in Figure (6.11) and a detailed implementation has been given in Figure (6.12). The numerical stability issues of concern in the implementation are due to the division operators in the interpolation procedure, and the maximum allowable step-length. Care has been taken to avoid a division with a small number. In the case of such an occurrence, a logical alternative has been used (described in Figure (6.11)). (See Sherali and Pendyala et al.,1994 for a well documented “C” code)

6.2.2 Hooke and Jeeves: The Bazaraa, Sherali and Shetty version of this derivative free multidimensional search technique has been theoretically described in Chapter 4. Its theoretical convergence argument has also been provided. A detailed implementation has been provided in Figure (6.13).

This section details the termination criterion, additions made to the algorithm for computational competitiveness, and discusses the meshing of the direction generation and the line-search strategies. Computational experience with this algorithm using the test problems described in Chapter 5 is also provided below.

The implementation uses the positive and negative coordinate axes as the exploratory search directions. An exploratory direction is identified as an improving direction by checking the function value at a designated step-length (parameter) distance along that direction. The computational competitiveness of the algorithm is highly sensitive to this parameter. After an improving direction has been identified, a line-search is conducted along this direction and a new solution results. After all the exploratory directions are checked, a pattern search is conducted.

The parameter described above is the key to the design of a computationally competitive

algorithm along with a good termination criterion. (First-order optimality conditions are not used because of the absence of gradient evaluations.) Let us start with a vector of parameters, i.e., a step-length value along each direction that can be used to check if the direction is an improving direction. This vector has to be dynamically updated at each iteration so that it contains information about the structure of the function in the vicinity of the current solution. (This enhances computational competitiveness.) This has been achieved by using the optimal step-length of the previous iteration (result of the previous line-search) as the updated value along that direction in case an improvement was obtained. In the case of a failure, this parameter is decreased (because it is expected that step-lengths decrease as we approach the optimum) by a multiplicative parameter (0.5). The Euclidean norm of these parameters is used to identify termination (Termination tolerance 0.01). This enhances convergence because even if an iteration does not result in an improvement, the search continues with decreasing exploratory step-lengths until the step-lengths are too small (that is, the Euclidean norm of the prescribed step-length vector is below an acceptable termination tolerance). The algorithm has also been tried with the grid size variation scheme detailed in Chapter 4. Table 1 presents the computational results obtained using the test functions described in Chapter 5.

Table 1: Hooke & Jeeves Results

Pr #	n	Final Soln.	# fn eval	# iter	Init obj fn	final obj fn	Best Sol
1	2	(3108.92, 2191.52)	37200	10	69.93	64.43	64.23
2	2	(3077.91, 2070.70)	7276	10	69.93	64.41	64.23
3	2	(3202.66, 2228.96.)	34600	10	77.66	71.03	68.54
4	2	(3328.64, 2672.79.)	5876	10	77.66	73.77	68.54
5	2	(3265.16, 2177.81)	37200	10	82.82	74.38	74.27
6	2	(3327.66, 2469.40)	5051	10	82.82	75.83	74.27
7	2	(0.99726, 0.995)	280	10	484.19	.00074	0
8	2	(0.3869, .0589)	265	10	749.03	.37	0

Table 1: Hooke & Jeeves Results

Pr #	n	Final Soln.	# fn eval	# iter	Init obj fn	final obj fn	Best Sol
9	4	(1.5946,2.5476,.1738,-0.0234)	444	10	19192	4.7	0
10	4	(0.5068,-0.0234,0.2441,0.4967)	608	10	2735	0.46	0
11	2	(0.9572,0.9381)	312	10	116	0.0022	0
12	4	(0,-0.0631,0.0207,-0.00708)	405	10	532.4	3.52	0
13	4	(0,-0.0085,-0.00708,-0.00708)	325	10	10	.00049	0

ANALYSIS OF RESULTS: The first column of the above table is the test problem number, n is the number of variables in the test problem, the third column is the final solution obtained, the total number of propagation predictions is recorded in the fourth column for examples 1 - 6 and the total number of functional evaluations is recorded in the fourth column for examples 7- 13, the number of iterations is given in the fifth column, (here, an iteration has been defined as a set of n exploratory searches and one pattern search), and the sixth and seventh columns represent the initial and final objective function values, respectively. The eighth column of the above table has the best known objective function values. The best known objective function value has been computed using an exhaustive (brute force) randomized search. The “best solution” is known to be an optimum value for the test problems 7- 13.

The first, third and fifth test functions in the above table are three single transmitter location examples described earlier with different convex combination parameter values.(0.2,0.5 and 0.7 respectively). The second, fourth and sixth examples are the same problems as the first, third and fifth, but executed with the grid variation scheme described earlier. The initial solution for the above four test problems has been computed using the procedure described in Chapter 4. The highest grid density used during the solution procedure has been used in computing the initial and final solutions. The results clearly indicate the computational superiority of the grid variation procedure. The first six examples also

indicate a consistency in the convergence of the algorithm (compare the final objective function values with the best known objective function values). Also, the results suggest that the designed termination criterion is effective.

The rest of the test functions are the standard functions described in Chapter 5, listed in the same order. Their final and best known objective function values and their solutions also reinforce the convergence properties of the algorithm for the different test problems. The number of function evaluations and the number of iterations recorded in Table 1 will be used later to compare the computational competitiveness of this method versus the other proposed procedures. All the above problems were solved using the same parameter settings with an acceptable termination tolerance of 0.01. Since a high accuracy tolerance was used, as a safeguard, an alternative termination criterion based on a maximum number of iterations (10) was also employed. It should be noted that increasing the maximum allowable number of iterations from 10 to 15 improved some of the solutions very marginally. The greatest improvement has been noticed in example 9, the final objective function value was 1.88.

6.2.2 Conjugate gradient methods: The theoretical motivation behind these gradient deflection algorithms has been discussed in Chapter 4. Figure (6.14) provides a flow chart describing the implementation details for the variant of this algorithm that is designed for the present problem. This section describes some particular details of the various steps, along with the alterations made for improving the computational competitiveness and numerical stability, and provides computational results.

PROJECTION: As described in Chapter 4, the algorithm has to be modified to accommodate the box constraints. This is done by projecting any direction that is infeasible to these box constraints back onto the feasible region.

GRADIENT EVALUATION: This algorithm is a gradient based search technique. The overall convergence and the computational competitiveness of the algorithm is very highly influenced by the accuracy of the gradient evaluation procedure. The theory behind the finite differencing and first-order model building procedures has been described in Chapter 4. A forward differencing technique, a central differencing technique and a first-order model building technique have been investigated for this problem. While the latter two techniques give a more accurate estimate of the gradient at a point, the former uses half as many evaluations as does the central differencing technique. The number of functional evaluations explode with the number of variables in the problem in the model building technique. A more accurate gradient evaluation can be expected to result in better search directions and hence reduce the number of iterations. Our computational experience with the present problem indicates that the accuracy of the gradient evaluation for all the above methods is comparable. Hence the forward differencing technique has been used with the proposed algorithm.

While solving unstructured problems like the present one, the step-length δ that is used for computing the forward differences (Equation (4.9)) is a very important parameter. A relatively large δ contains information regarding a considerable local neighborhood and may not be a good estimate of the gradient at that point. Directions generated using this gradient take the iterations to the vicinity of an optimum faster, but experience convergence difficulty near the optimum. A relatively smaller δ provides a good local estimate of the gradient and hence, is good for the final convergence purpose. In the instance of the present problem, we have good quality initial solutions, and therefore, relatively small values of δ can be used. A further reduction in these values as the iterations proceed improve the asymptotic convergence properties. The initial value for δ is 0.5, and the consecutive reductions used halve δ for every two new directions generated, until it is reduced to 0.125.

DIRECTION GENERATION: As described in Chapter 4, a new direction is generated in this method by deflecting the anti-gradient at the present iteration using an additive term in the spirit of conjugate gradient methods. Equations (4.11, 4.12 and 4.13) are used to implement this idea. A three term conjugate gradient method (Dixon, Ducksbury and Singh, 1977) has also been tried. In this method, the deflection proceeds as follows,

$$d_{k+1} = -\nabla f(x_{k+1}) + (\beta_k) d_k + \gamma_k d_1 \quad (6.1)$$

where d_{k+1} is the direction vector at iteration $k+1$, β_k is computed using Equations (4.12 or 4.13), and $\gamma_k = \frac{(\nabla f(x_{k+1}))^T q_k \vee (k \neq 1)}{(d_1)^T q_1}$ and zero otherwise. This method is designed to retain information that has been accumulated over previous iterations when restarting with a direction d_1 . Since the present problem has a good initial solution, we have noticed that Equations 4.12. and 4.13 provide us with the best computational results i.e., in the initial iterations, when the grid is sparse, equation 4.12 has been used and as the algorithm proceeds to denser grid, Equation 4.13 has been used. It can be seen from Equation 4.13 that a numerically insignificant step-length at an iteration, or a numerically insignificant denominator in the computation of β_k at any iteration, can cause numerical stability problems. In such a case we have restarted the algorithm as described at the end of this section.

TERMINATION: The standard KKT conditions described below

$$(\nabla f(x))_i \geq 0 \forall x_i = h_i$$

$$(\nabla f(x))_i \leq 0 \forall x_i = 0$$

$$(\nabla f(x))_i = 0 \quad \text{otherwise.}$$

have been used to identify optimality (the value of 0.0001 was used instead of 0 in the implementation). In addition, if the sum of the optimum step lengths over the previous three line-searches is below an acceptable tolerance (0.01), the procedure is terminated. This termination criterion is a valuable safeguard, because the KKT conditions may not be satisfied in the case of unstructured problems, even in the very close vicinity of an optimum. (As mentioned previously, an alternate criterion based on a maximum iteration limit (7) has also be employed.) It should also be noted that the improvement - based termina-

tion parameter can be used to strike a balance between computational competitiveness and accuracy.

RESTARTING AND SCALING: A restart of the algorithm, using the present iterate as the initial solution and its negative gradient direction as the initial search direction, is done each time n directions are generated or when numerical instability is detected in the computation of a new direction. This has been a proven technique to improve computational competitiveness while solving non-quadratic problems.

Also, it has been noticed that the direction vectors generated are numerically insignificant when compared with the objective function values and the step-lengths obtained during the line-search. This might result in a loss of accuracy due to round-off, and hence result in a poor performance of the overall algorithm. The direction vector is hence scaled by multiplying it by a factor of 1000.

Table 2: Conjugate gradient Results

Pr #	n	Final Solution	# fn eval	# iter	Init obj fn	Final obj fn	Best Sol
1	2	(2807.10,2050.87.)	53400	3	69.93	65.60	64.23
2	2	(3187.41,2672.42)	1450	1	69.93	67.25	64.23
3	2	(3269.29,2184.82.)	43800	5	77.66	71.51	68.54
4	2	(3126.24,2196.35)	37300	7	77.66	70.82	68.54
5	2	(3307.83,2175.22)	23800	1	82.82	74.90	74.27
6		(3098.41,2361.30)	24125	5	82.82	76.69	74.27
7	2	(0.9790, 1.002)	185	2	484.19	0.04	0
8	2	(0.279,0.0129)	181	2	749.03	0.52	0
9	4	(-.6691,.417,.2059,0.0756)	168	1	19192	26.32	0

Table 2: Conjugate gradient Results

Pr #	n	Final Solution	# fn eval	# iter	Init obj fn	Final obj fn	Best Sol
10	4	(0.32,-0.08,-0.23,0.06)	459	3	2735	0.73	0
11	2	(1.017,1.09)	653	6	116	0.003	0
12	4	(-0/12,-0.02,-0.03,-0.051)	323	1	532.40	4.01	0
13	4	(-0.13,-0.1,-0.01,-0.05)	309	2	10	0.05	0

ANALYSIS OF RESULTS: Table 2 records the computational experience with the Conjugate gradient Method using the usual format (same as Table 1). Here an iteration has been defined as n direction generations. The results tabulated above can be used to assess the convergence properties of the proposed solution procedure and the computational and convergence advantages of grid size variation. These results are used to compare this algorithm with the other direction generation procedures.

Test problems 1-6 demonstrate the convergence properties of the solution procedure. It should be noted that the number of iterations are very few for this algorithm. It has been noticed that for most termination parameters, the grid size variation scheme performs better, but it should be noted that the results of Example 2 and 6 indicate a higher final objective function. This is because of the availability of a good initial solution. The use of grid size variations is still preferred because of its tendency to produce more accurate solutions in the case of a poor starting solution. For example, test problems 3 and 4 were run with an arbitrary starting solution of (2000, 2000), being similar in every aspect except that problem 4 was solved using the proposed grid size variation while problem 3 used the final refined grid throughout. The final solution obtained for problem 3 was (3370.75, 2322.53), with the initial and final objective function values being 78.788, and 75.31 respectively. (This is not competitive with the solutions obtained using a good starting solution.) On the other hand, the final solution obtained for problem 4 using the same arbitrary starting solu-

tion was (3168.588, 2215.591) of final objective function value 70.928825. This solution is indeed comparable to the solution obtained using a good initial start.

6.2.4 Quasi-Newton Method: The theoretical motivation behind these gradient deflection algorithms has been discussed in Chapter 4. These methods typically require more memory than the conjugate gradient methods. The implementation of these methods is exactly identical to the implementation of the conjugate gradient methods except for the direction generation procedure. The direction generation procedure involves updating a deflection matrix (approximation to the inverse Hessian), ensuring positive definiteness of this matrix, and deflecting the present anti-gradient by multiplying with this matrix.

Computational experience with this algorithm and a comparison with the earlier solution procedures is provided below.

DIRECTION GENERATION: The deflection matrix D is initialized as an identity matrix. This makes the initial direction the steepest descent direction. Also, the deflection matrix is reinitialized as an identity every time the algorithm is restarted. D_k is sequentially updated by adding a correction term given by the BFGS update according to

$$D_{k+1} = D_k + C_k \quad (6.2)$$

$$C_k = \left[\begin{array}{c} \left(\frac{\rho_k \rho_k'}{\rho_k' \sigma_k} \right) \left\{ 1 + \frac{\sigma_k' D_{k-1} \sigma_k}{\rho_k' \sigma_k} \right\} - \frac{(D_{k-1} \sigma_k \rho_k' + \rho_k \sigma_k' D_{k-1})}{\rho_k' \sigma_k} \end{array} \right] \quad (6.3)$$

where C_k is the additive correction matrix, $\rho_k = (x_{k+1} - x_k)$ and σ_k is the difference of the gradients at x_{k+1} and x_k . This BFGS update is computationally known to perform very well with inexact line-searches. The positive definiteness of D_{k+1} is ensured using the super diagonalization technique. The procedure is restarted if numerical instability results in the updating of the deflection matrix.

The rest of the implementation is consistent with the implementation of conjugate gradient methods. The same parameters were also used in this implementation, except the maximum number of iterations(5).

Table 3: Quasi-Newton Results

Pr #	n	Final Solution	# fn eval	# iter	Init obj fn	Final obj fn	Best Sol
1	2	(3106.51,2118.60)	25400	3	69.93	64.33	64.23
2	2	(3184.74,2696.87)	11900	13	69.93	68.07	64.23
3	2	(3037.26,2067.68.)	31600	2	77.66	72.26	68.54
4	2	(3444.74,2211.31.)	27300	6	77.66	72.01	68.54
5	2	(3301.20,2679.11)	22800	2	82.82	81.24	74.27
6		(3181.74,2390.08)	3650	2	82.82	76.06	74.27
7	2	(0.9722, 0.9993)	111	1	484.19	0.079	0
8	2	(0.1705,0.0091)	107	1	749.03	0.6897	0
9	4	(.3144,0.1002,-0.7359,0.5461)	794	1	19192	21.82	0
10	4	(-3.2252,.0127,-1.073,-3.29)	978	1	2735	57.13	0
11	2	(.7234,.5129)	318	3	116	0.07	0
12	4	(-.2138,-.108,-.1403-.352)	260	1	532.40	22.50	0
13	4	(-0.0859,-0.1117,-0.1,-0.1031)	376	1	10	0.1	0

ANALYSIS OF RESULTS: Table 3 records the computational experience with quasi-Newton Method using the usual format (same as Table 1). Here an iteration has been defined as n direction generations.

Test problems 1-6 demonstrate the convergence properties of the solution procedure. A comparison of the results (objective function values and number of function evaluations) of Examples 1, 3 and 5 with 2, 4 and 6, respectively, illustrate the advantage of the grid

size variation procedure.

While the above results clearly demonstrate the computational convergence of the quasi-Newton method, the inconsistency in its behavior when compared to the conjugate gradient procedure can be explained by the fact that this procedure approaches the steepest descent method in the case of too many restarts. These restarts are triggered whenever poor Hessian estimates cause the generated directions to be non-improving. Also, the frequent loss of positive-definiteness ultimately results in the Hessian becoming diagonally dominant after sufficiently large constants are added to the diagonal, and the generated directions then degenerate to steepest descent directions.

Comparison: As far as the relative computational competitiveness of the three search direction strategies are concerned, we observe that on the first six problems, the relative algorithmic performance is somewhat balanced. with the algorithm that produces better quality solutions before terminating consuming more functional evaluations as well. On the remaining test problems, Hooke & Jeves algorithm appears to be most robust with respect to the quality of the solution produced, although it requires the most number of functional evaluations. The Conjugate gradient method appears to be somewhat more robust than the quasi-Newton method.

Next, let us consider some multiple transmitter problems and analyze their results.

6.2.5 Multiple Transmitter Problem: The results presented in this section are intended to illustrate the modelling procedure described in Section 2.2 and the solution procedure described in Section 4.3.

Four examples have been created to illustrate the methodology for the multiple transmitter

problem. Each of these four examples mentioned above solves the Whittemore Second Floor problem using five transmitters. The first example uses equal weights (1.0) for all the spatial design grid points. The second example uses a relatively heavier weightage (2.0) for the receivers at the center of the design space. The third example uses a relatively heavier weightage (2.0) for all the receivers on the right edge of the design space. The fourth example solves the problem by using relatively heavier weights ascribed to two sets of receivers, one being located at the right corner of the design space, and the other being located along the middle section of the lower edge of the design space. The solutions for the above scenarios have been pictorially depicted and analyzed below.

Figure 6.6 depicts the initial location (*) and allocation (partitions) corresponding to the starting solution for example 1. (The transmitters have been numbered for convenience.) The problem is then solved using the model in Section 2.2 and the methodology in Section 4.3. The final solution is depicted in Figure 6.7. The changes in the allocations made for the final solution are consistent with the physical structure of the design space (Figure 6.4).

Example 2 gives a higher priority to all the receivers placed at the center of the design space. Figure 6.8 depicts the dotted rectangle at the center of the design space representing this region of high priority. The initial solution is given by "*" and the final solution by "&". It should be noted that the allocation of receivers lying within the high priority region has been split among four transmitters, depending on the partition of the walls and the presence of a hallway in that region. The final allocations of the solution are depicted in Figure 6.8 in the form of partitions. The rest of the transmitters are re-located to service their respective receivers.

Example 3 places a higher priority on the receivers located along the right-hand edge of the design space. The situation is represented pictorially in Figure 6.9. The initial and final

solutions have been represented by “*” and “&”, respectively. It should be noted that transmitters 1 and 3 have moved closer to the right-hand edge and that their area of coverage has increased along the edge. The location of transmitter 3 and the allocation made to it represent the presence of a hallway in the design space. The final allocations are represented by the partitions in Figure 6.9.

Example 4 is designed to check the sensitivity of the solution if different areas are given a higher priority. The dotted lines in Figure 6.10 represent two such areas of a relatively higher priority (weights of 1.5). The initial and final solutions are represented by “*” and “&”, respectively. The partitions represent the final allocations. The movement and the change in allocations for transmitters 1 and 4 represent the response of the solution to the high priorities given to the receivers along the dotted lines. The location and receiver allocations for transmitter 2 is being governed by the presence of a hallway in the design space.

Now, let us use the above built examples to experiment with the performance of the modified Conjugate gradient algorithm

Table 4: Multiple Transmitter Problem Results

#	Initial Solution	Init Obj fn	Final Solution	Final Obj fn	#fn eval (# prop pred)
1	2082.66, 1980.04 2823.84, 1980.04 3935.62, 1980.04 2453.25, 2583.12 3935.62, 2583.12	6517.29	2839.88, 2944.97 3265.17, 2869.02 3935.62, 1996.41 2453.25, 2583.12 4750.93, 2898.28	5948.23	258 (51600)
2	2082.66, 1980.04 2823.84, 1980.04 3935.62, 1980.04 2453.25, 2583.12 3935.62, 2583.12	6517.29	2082.66, 1980.04 1786.18, 2994.97 3935.62, 1980.04 2453.25, 2583.12 3935.62, 2583.12	5346.17	78 (390)

Table 4: Multiple Transmitter Problem Results

#	Initial Solution	Init Obj fn	Final Solution	Final Obj fn	#fn eval (# prop pred)
3	3195.44, 2281.58 3204.44, 2286.58 3214.44, 2291.58 3224.44, 2296.58 3234.44, 2301.58	8545.99	4750.93, 2385.60 4033.74, 2994.97 3424.98, 2695.08 3824.48, 2994.97 3581.40, 2994.97	5715.31	340 (25500)
4	2000, 2000 2500, 2000 3000, 2000 3500, 2000 4000, 2000	7587.55	2262.60, 2944.97 4435.09, 2944.97 4750.93, 2465.67 4750.93, 2776.04 4637.64, 2626.37	5270.688	420 (67850)
5	2082.66, 1980.04 2852.53, 2015.06 3891.70, 1993.44 2532.40, 2564.73 3872.09, 2570.20	11598.07	1786.18, 1985.64 2262.01, 1738.81 3354.52, 1781.55 3038.05, 2944.97 3799.57, 2634.66	8455.34	337 (67400)
6	2082.66, 1980.04 2852.53, 2015.06 3891.70, 1993.44 2532.40, 2564.73 3872.09, 2570.20	11598.07	2083.75, 1980.04 2963.42, 1876.93 3891.70, 1993.44 2532.40, 2564.73 3872.09, 2570.20	7750.07	147 (11050)
7	3186.20, 2288.28 3196.20, 2293.28 3206.20, 2298.28 3216.20, 2303.28 3226.20, 2308.28	9400.36	3000, 2013.42 3196.20, 2166.28 3206.20, 2198.62 3216.20, 2228.90 3226.20, 2219.02	9299.71	135 (10150)
8	2000, 2000 2500, 2000 3000, 2000 3500, 2000 4000, 2000	8513.43	2000, 2038.28 2500, 2058.5 3031.39, 2211.03 3651.45, 2038.91 4375.46, 2023.13	8485.19	135 (10150)

Table 4: Multiple Transmitter Problem Results

#	Initial Solution	Init Obj fn	Final Solution	Final Obj fn	#fn eval (# prop pred)
9	2082.66, 1939.83 2823.84, 1939.83 3935.62, 1939.83 2453.25, 2583.12 3935.62, 2583.12	12754.17	2082.66, 1939.83 2823.84, 1939.83 3935.62, 1939.83 2453.25, 2583.12 3935.62, 2583.12	12754.17	102 (20400)
10	2082.66, 1939.83 2823.84, 1939.83 3935.62, 1939.83 2453.25, 2583.12 3935.62, 2583.12	12754.17	1858.17, 1831.89 3140.80, 1921.43 4307.41, 2323.90 2453.55, 2559.58 3939.73, 2941.27	12542.79	252 (34650)
11	3194.44, 2232.24 3204.44, 2237.24 3214.44, 2241.24 3224.44, 2246.24 3234.44, 2251.24	12001.60	2217.39, 2308.11 3041.71, 2107.88 3464.39, 2182.76 4750.93, 2100.17 3777.37, 1738.81	11442.22	289 (41750)
12	2000, 2000 2500, 2000 3000, 2000 3500, 2000 4000, 2000	12136.78	1907.56, 2009.34 2730.22, 2252.82 2926.06, 1896.79 3413.58, 1869.40 4097.88, 1954.18	11099.22	135(10150)
13	2082.66, 1958.11 2823.84, 1980.04 3935.62, 1980.04 2453.25, 2561.19 3935.62, 2583.12	8271.23	2082.66, 1958.11 2823.84, 1980.04 3935.62, 1980.04 2453.25, 2561.19 3935.62, 2583.12	8271.23	108 (21600)
14	2082.66, 1958.11 2823.84, 1980.04 3935.62, 1980.04 2453.25, 2561.19 3935.62, 2583.12	8271.23	1786.18, 2018.83 2856.91, 2062.25 3935.62, 1980.04 2391.44, 2540.24 3935.62, 2583.12	7892.89	204(23800)

Table 4: Multiple Transmitter Problem Results

#	Initial Solution	Init Obj fn	Final Solution	Final Obj fn	#fn eval (# prop pred)
15	3163.18, 2276.5 3173.18, 2281.5 3183.18, 2286.5 3193.18, 2291.5 3203.18, 2296.5	10109.19	2831.75, 2008.59 2993.30, 2286.57 3194.42, 2311.80 3250.11, 2419.40 3491.40, 2944.97	8205.5	137(9150)
16	2000, 2000 2500, 2000 3000, 2000 3500, 2000 4000, 2000	11432.02	1840.03, 1875.64 2390.37, 2001.40 2855.72, 2102.35 3495.31, 1971.72 4033.16, 2003.95	8230.12	207(24150)

Table 4 summarizes the results of the experiments. Column 1 assigns an index number to each experiment. Columns 2 and 3 present the initial solution and the corresponding objective function value. Columns 4 and 5 give the final solution obtained and the corresponding objective function value. Column 6 records the number of function evaluations and in parenthesis, the number of propagation predictions. Rows 1 - 4 tabulate the results of experimentation with Example 1. The first row solves the example without grid size variation and uses the initial solution proposed in Section 4.3. The second row solves the example with grid size variation and the same starting solution proposed in Section 4.3. The third row solves the example with grid size variation, but the initial location of the transmitters is clustered around the center of gravity of the entire design region. The fourth row solves the example with grid size variation and a manually prescribed random initial solution. Rows 5 - 16 tabulate the corresponding results for examples 2 - 4 in the same order. The above built models have been run with other intelligently selected starting solutions and other termination parameters, but better solutions didn't result.

The results summarized in Table 4 highlight the importance of the grid variation scheme in the convergence of the algorithm. Although the algorithm was started with a good initial solution (proposed in Section 4.3), it stalled at a relatively poor solution when run

without the grid variation scheme in examples 3 and 4. When the grid size variation was used, all the examples were solved to a better solution. The grid size variation resulted in a good solution even when the algorithm was started with poor initial solutions. As a whole, the above tabulated results demonstrate that the multiple transmitter problem can be solved using the proposed model and solution procedure quite effectively.

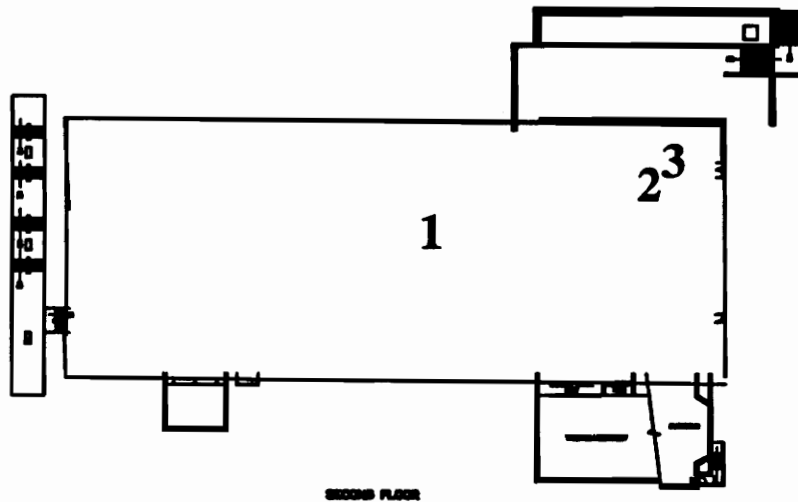


Figure 6.1; Effect of Weights Using the Minimax Objective Function.

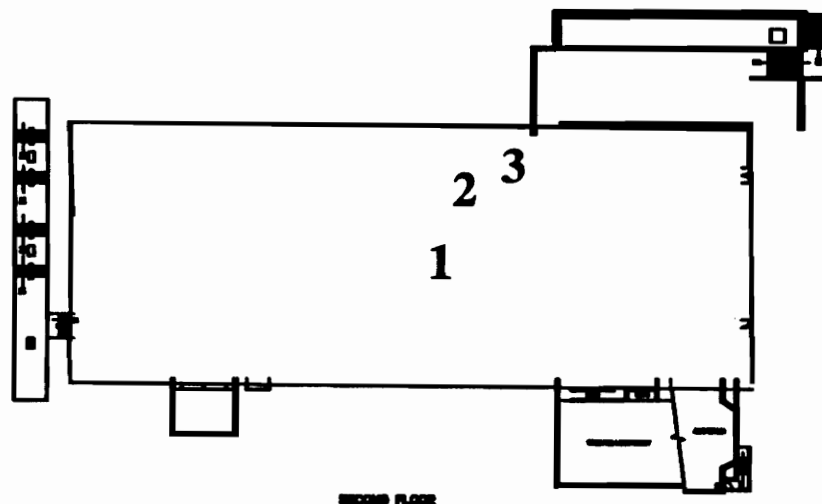


Figure 6.2; Effect of Weights Using the Minisum Objective Function

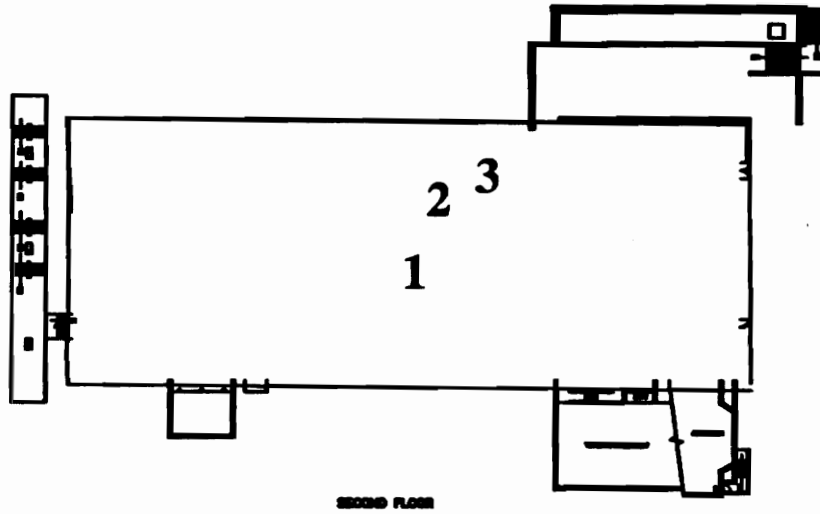


Figure 6.3; Effect of Weights Using the Convex Combination (0.5) Objective Function.

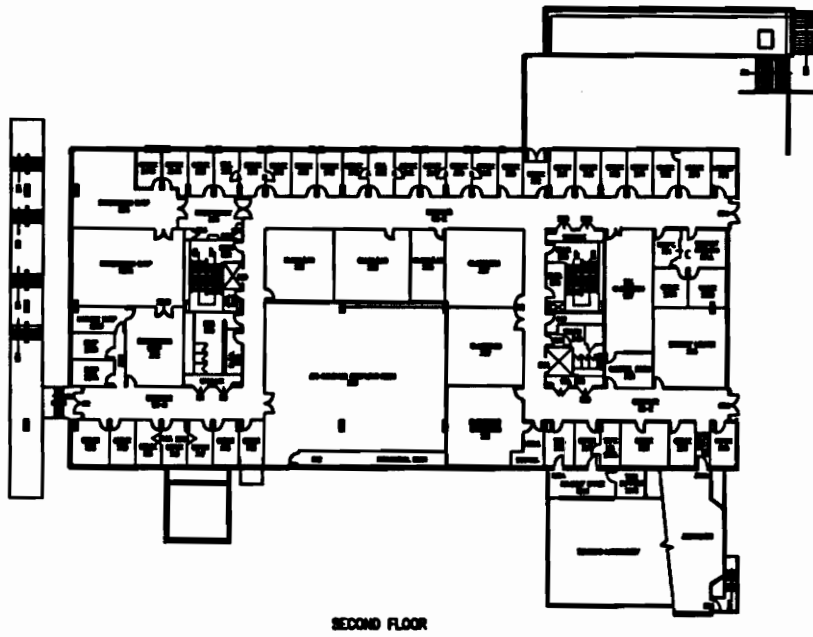


Figure 6.4; Whittemore Hall Second Floor.

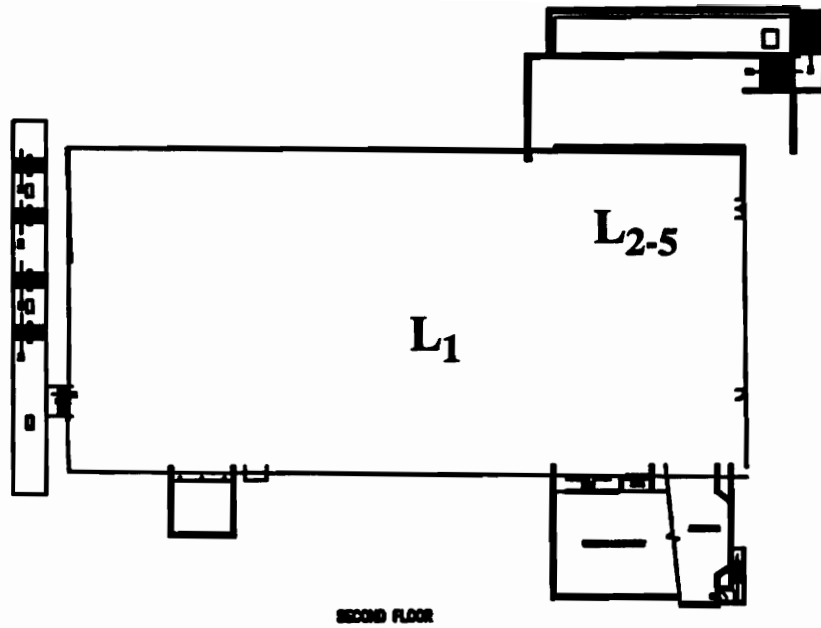


Figure 6.5; Effect of Varying Wall Attenuation Factors.

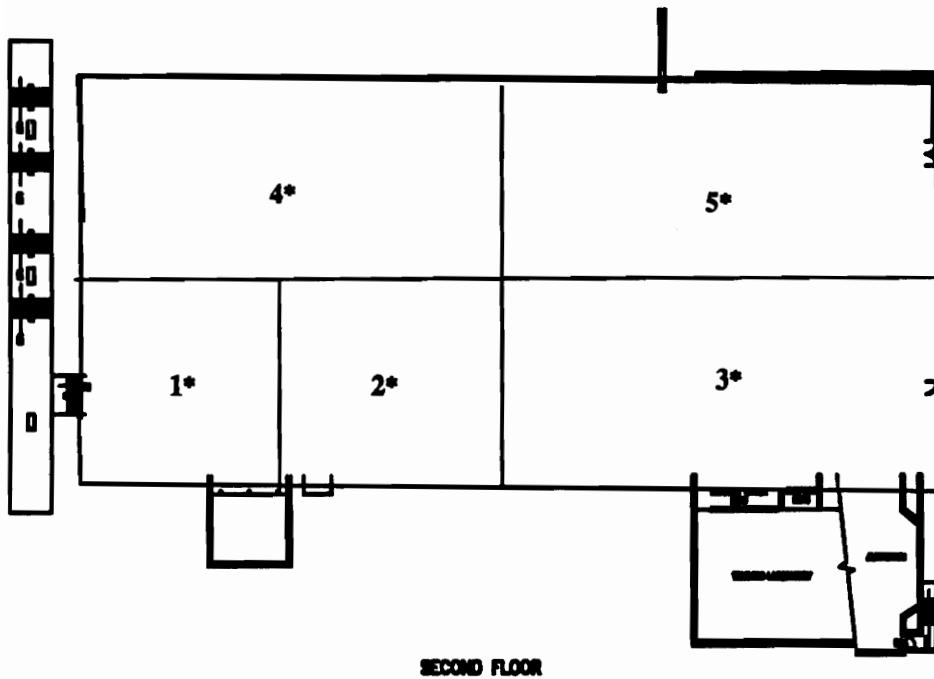


Figure 6.6 Initial Solution for Multiple Transmitter Problem.

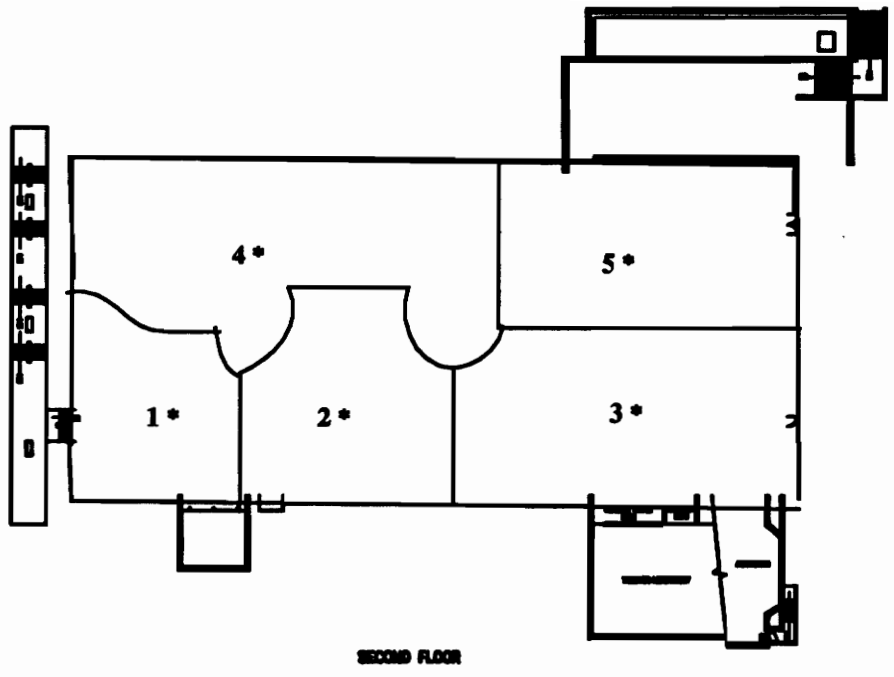


Figure 6.7; Final Solution with Equal Weights.

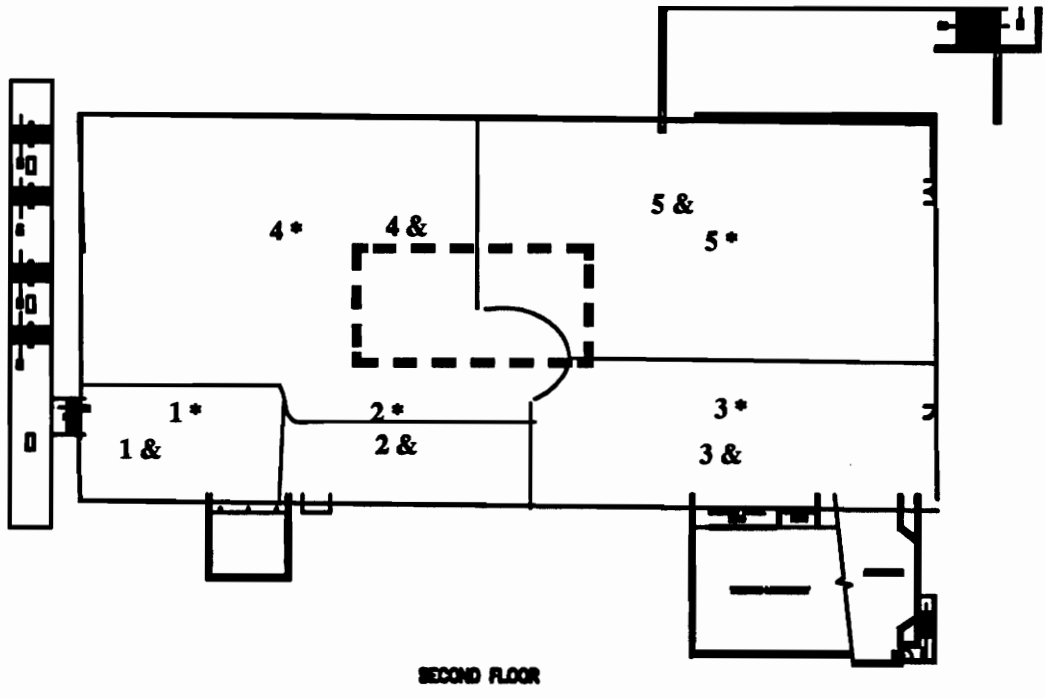


Figure 6.8. Initial (*) and Final(&) Solutions for Weights (1.5) at the Center.

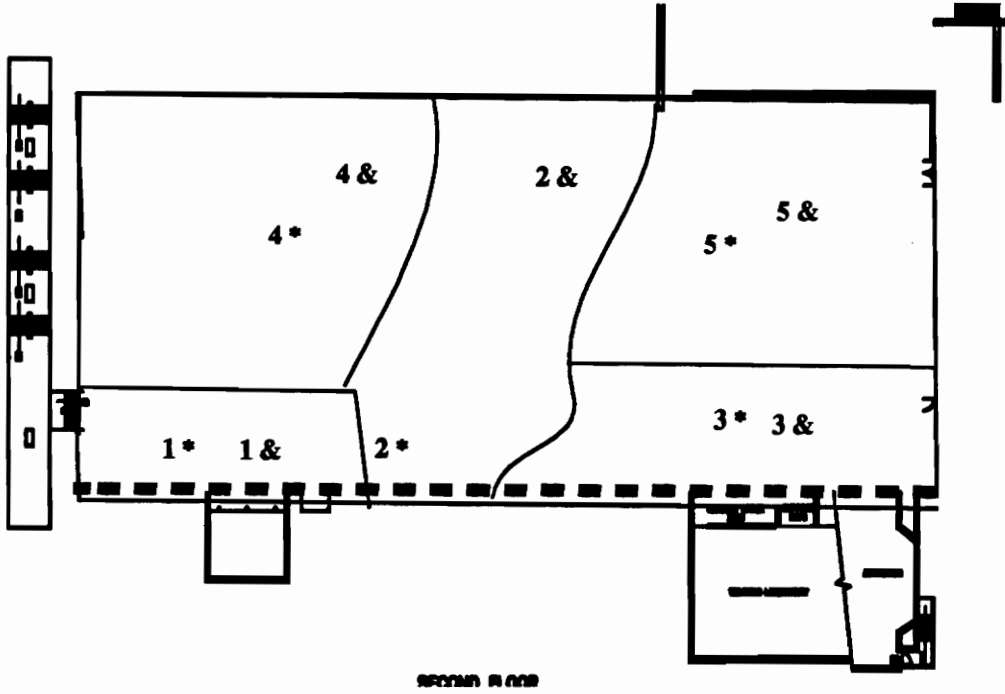


Figure 6.9; Initial (*) and Final(&) Solutions for Weights(1.5) at an Edge.

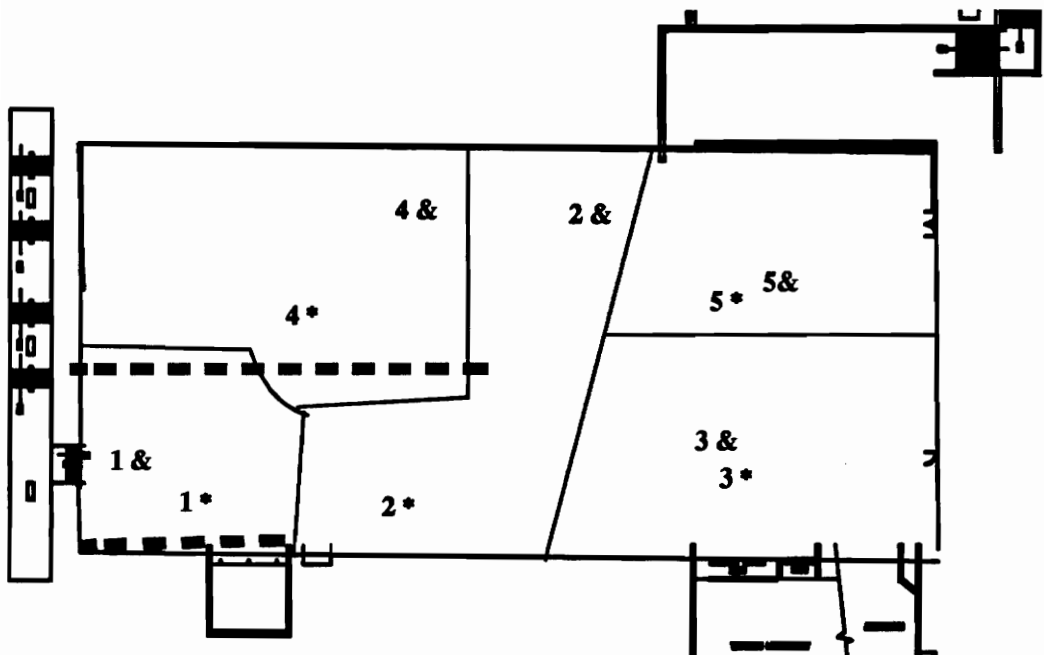


Figure 6.10 ; Initial (*) and Final(&) Solutions for Weights (2.0) at two Regions.

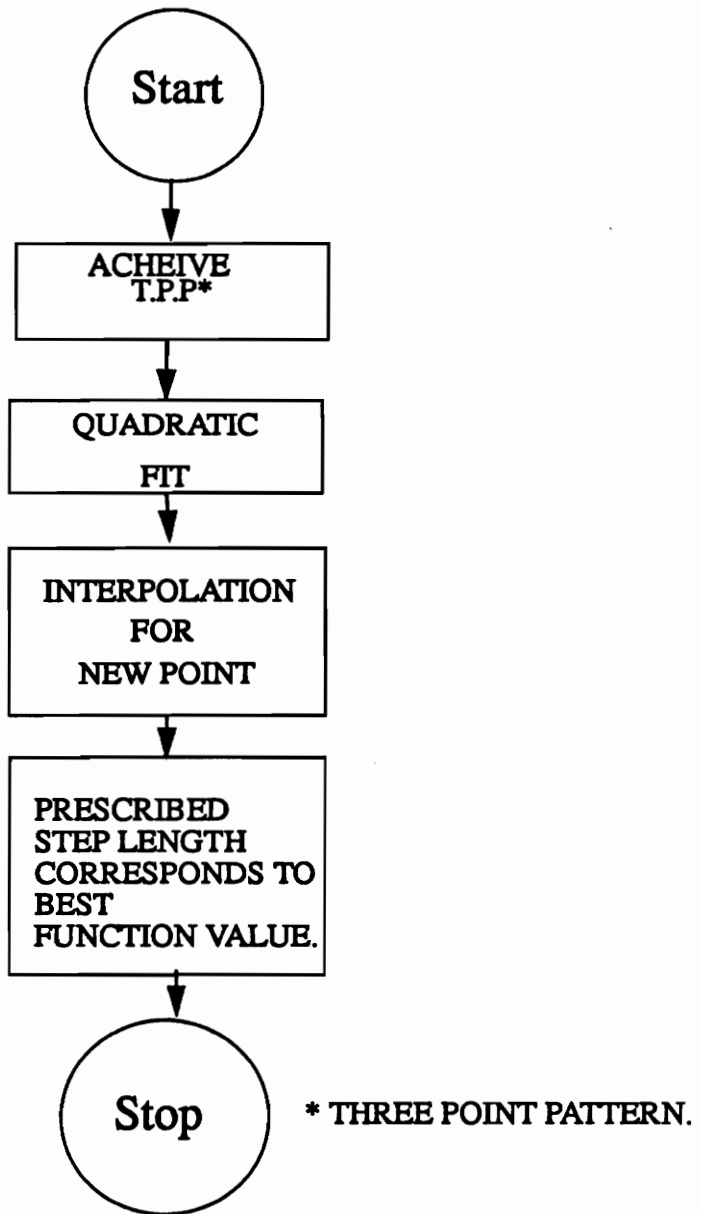


Figure 6.11; Inexact Quadratic Iterpolation.

M=Max step length.
K=Max # of iterations.
D=Direction vectors.
X=Point.
C=Box Constraints.
 $\alpha > 1$ (2.0)
 $0 < \beta < 1$ (0.5)

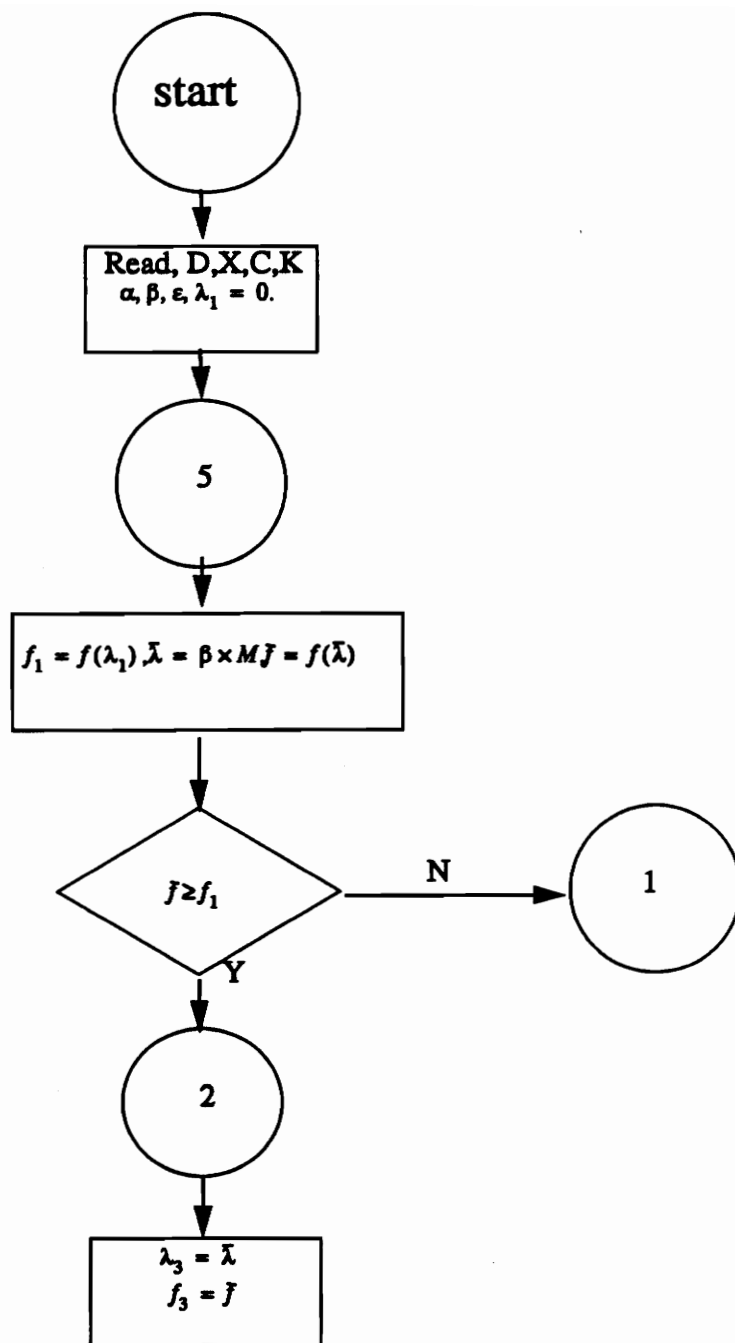


Figure 6.12 Line-search (continued..)

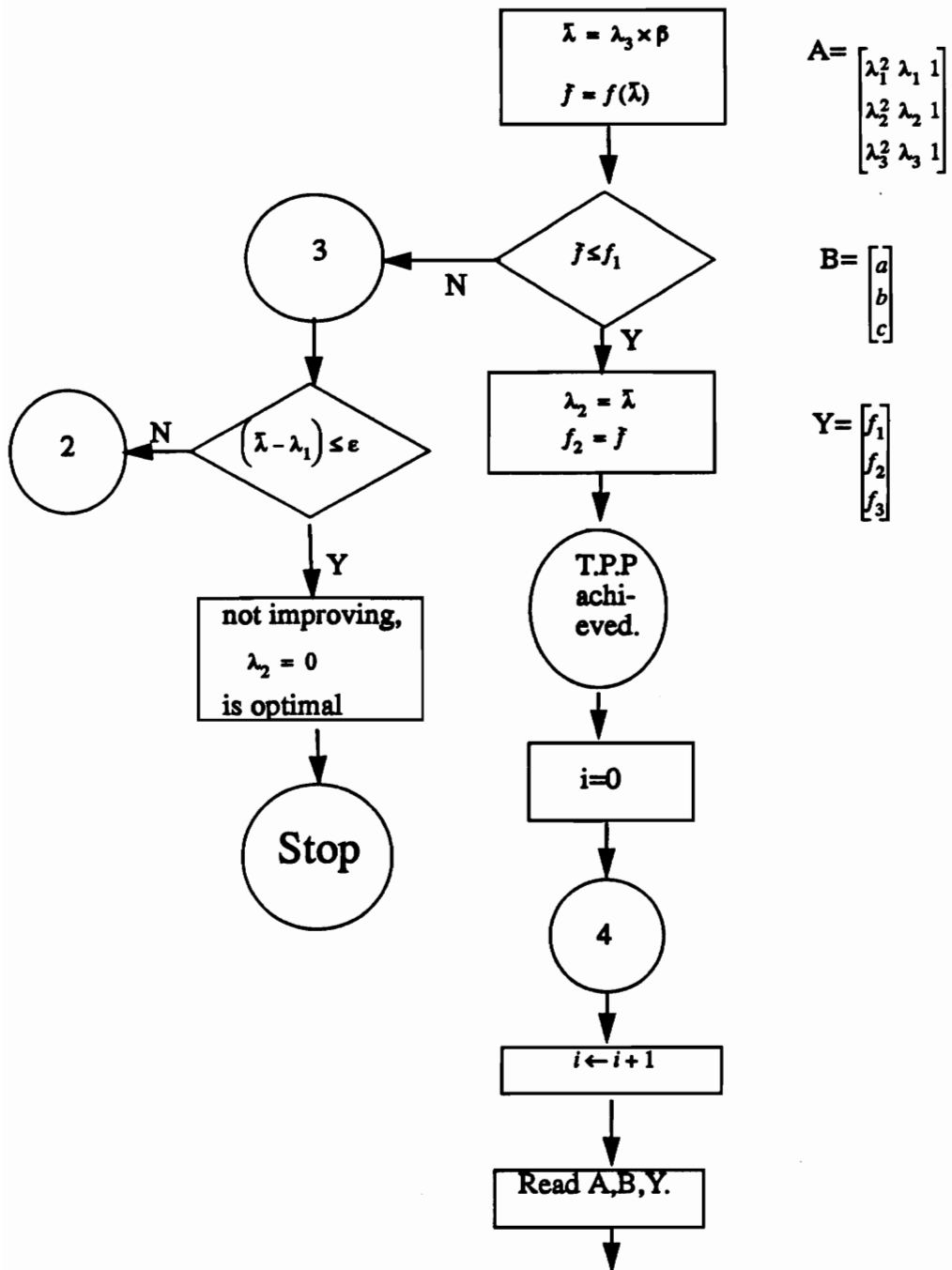


Figure 6.12 Line Search (continued..)

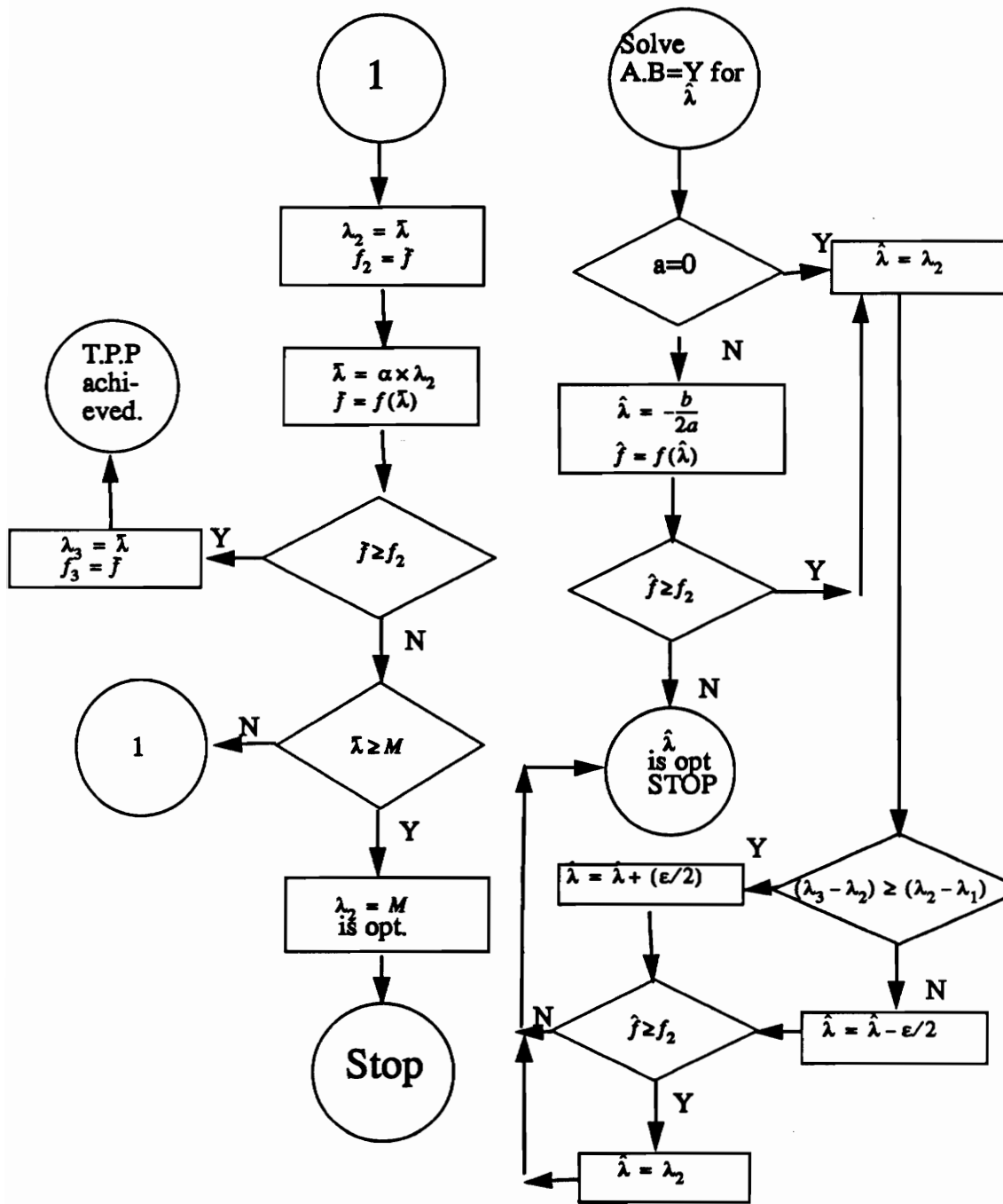


Figure 6.12 Line Search (continued..)

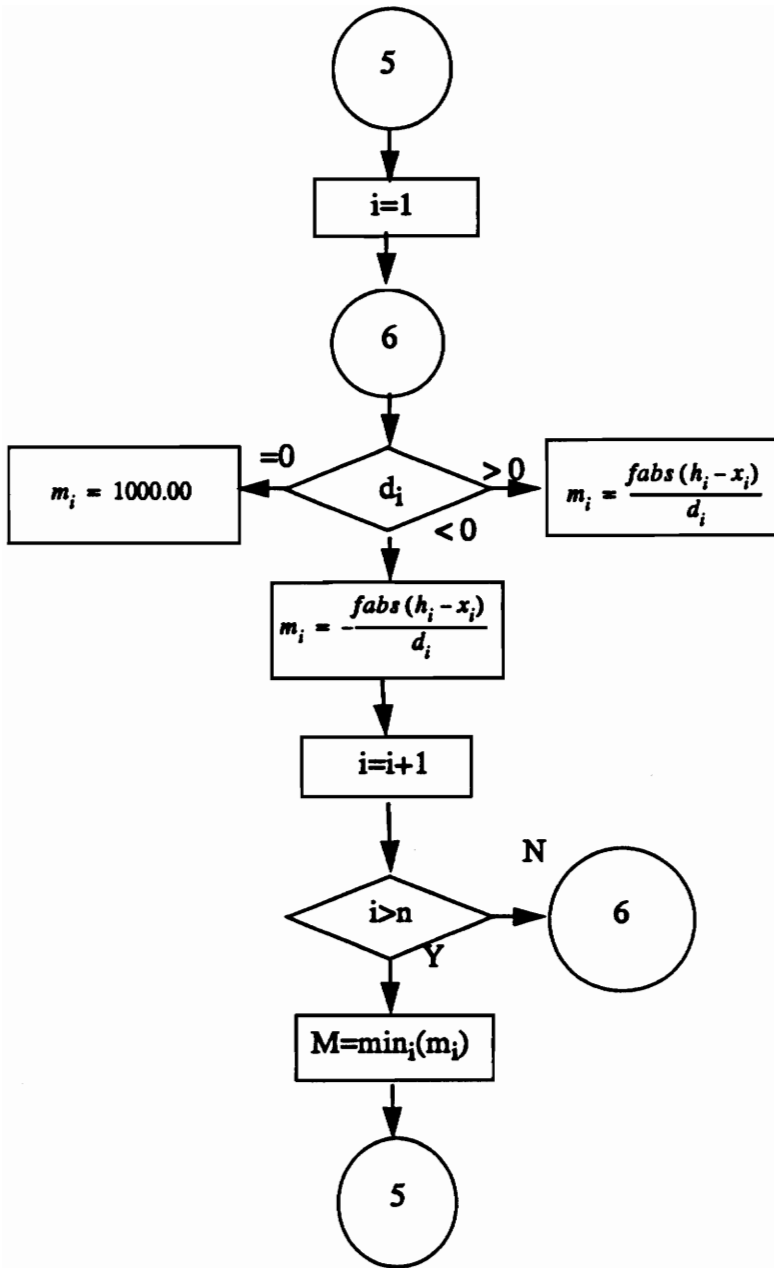


Figure 6.12 Line Search (continued..)

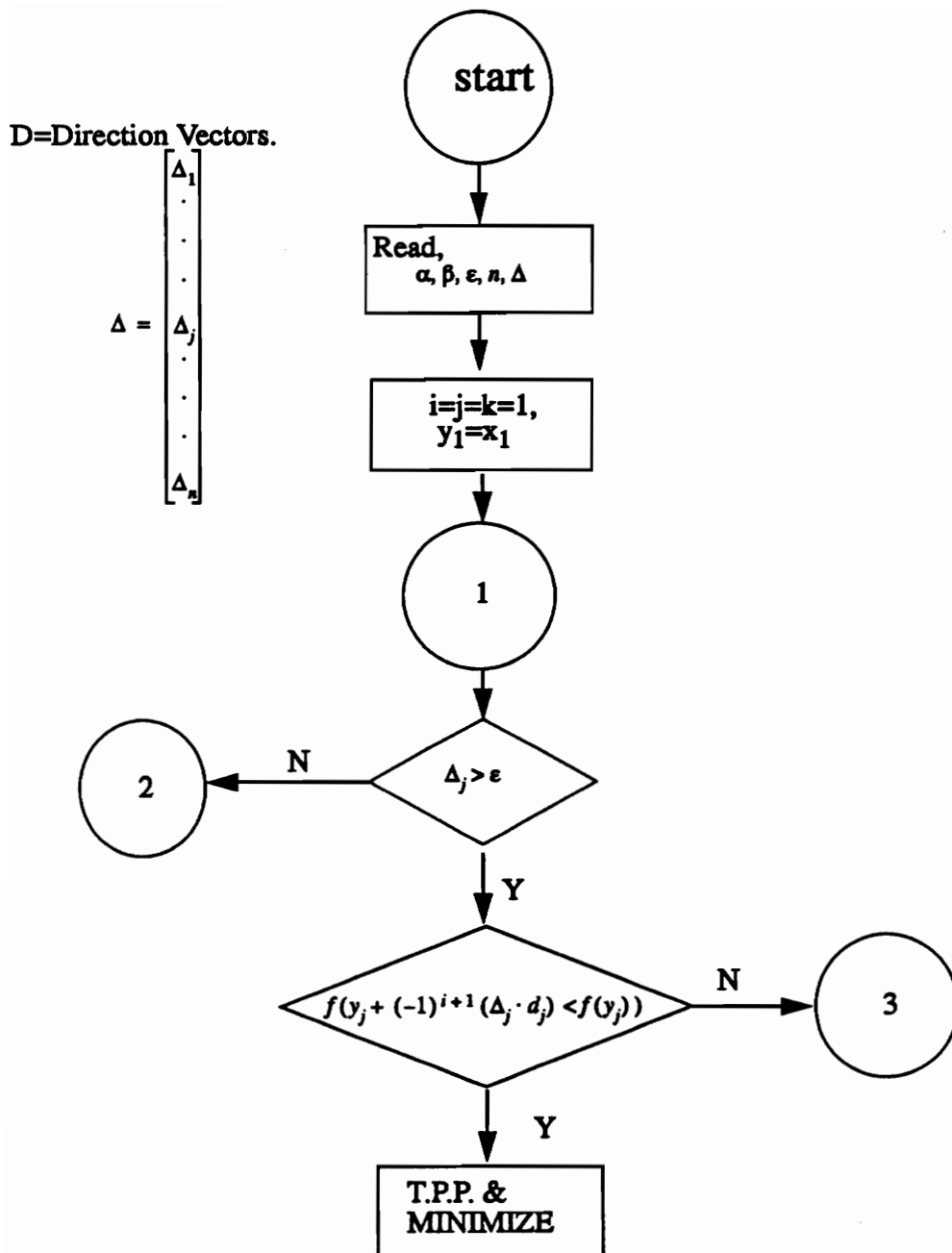


Figure 6.13 Hooke & Jeeves' Method. (continued..)

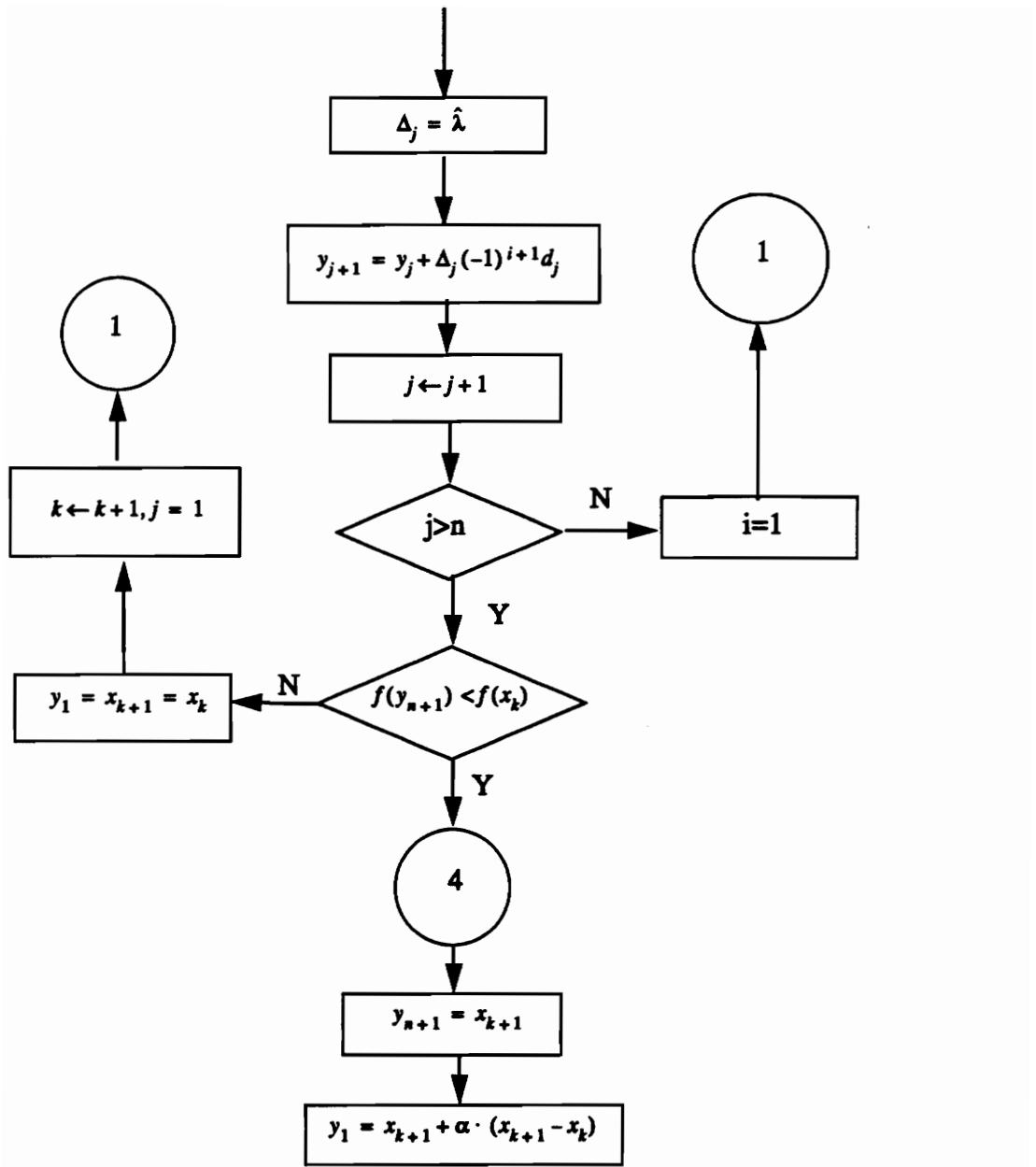


Figure 6.13 Hooke & Jeeves (continued..)

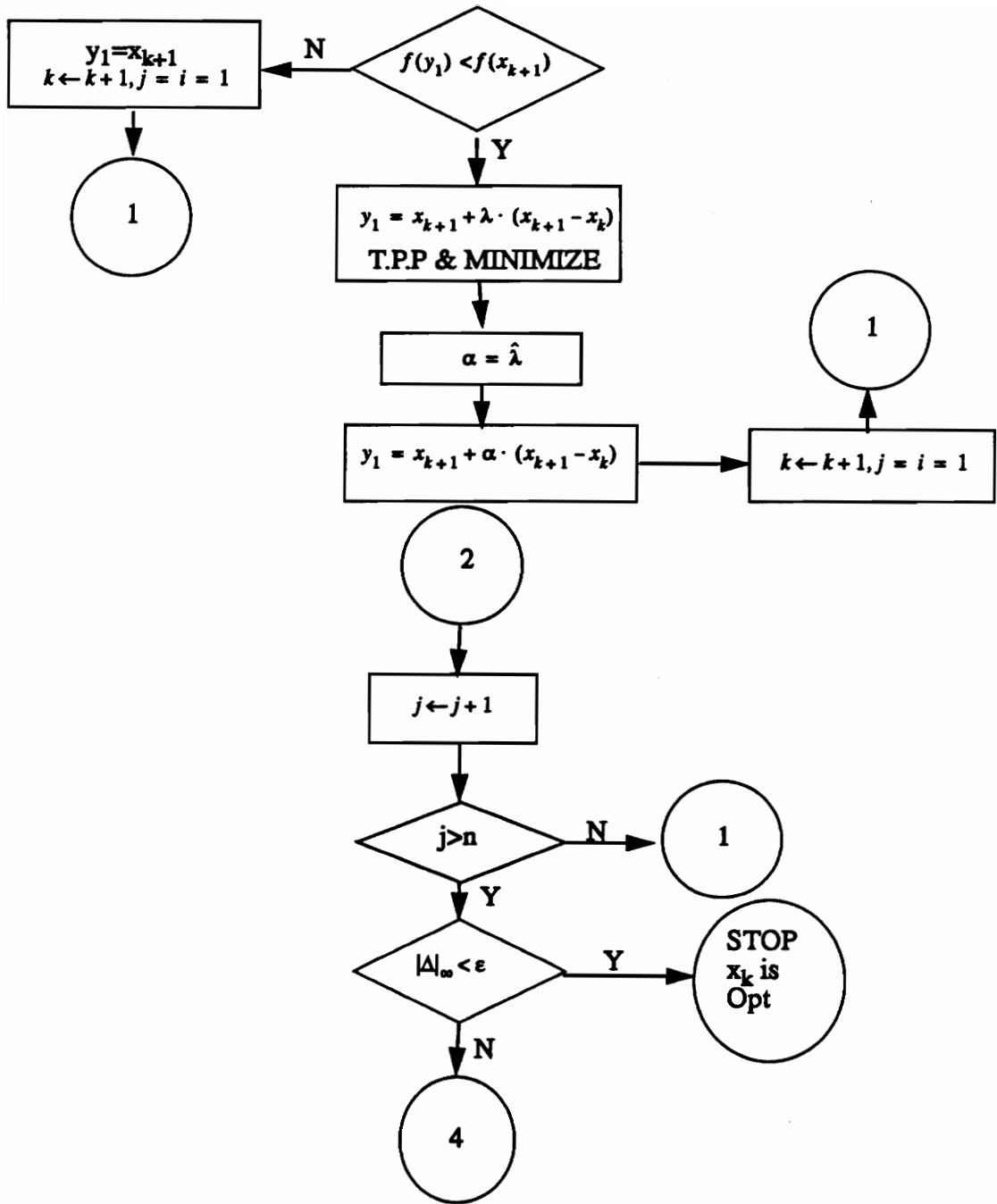


Figure 6.13 Hooke & Jeeves (continued..)

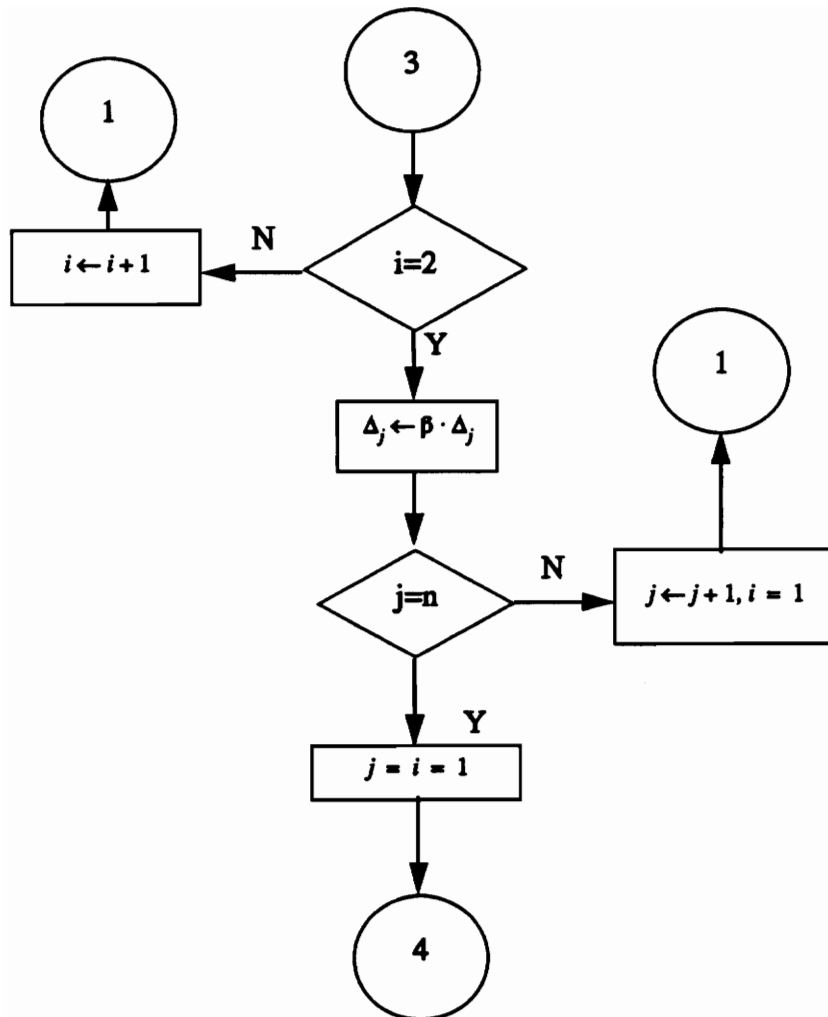


Figure 6.13 Hooke & Jeeves (continued..)

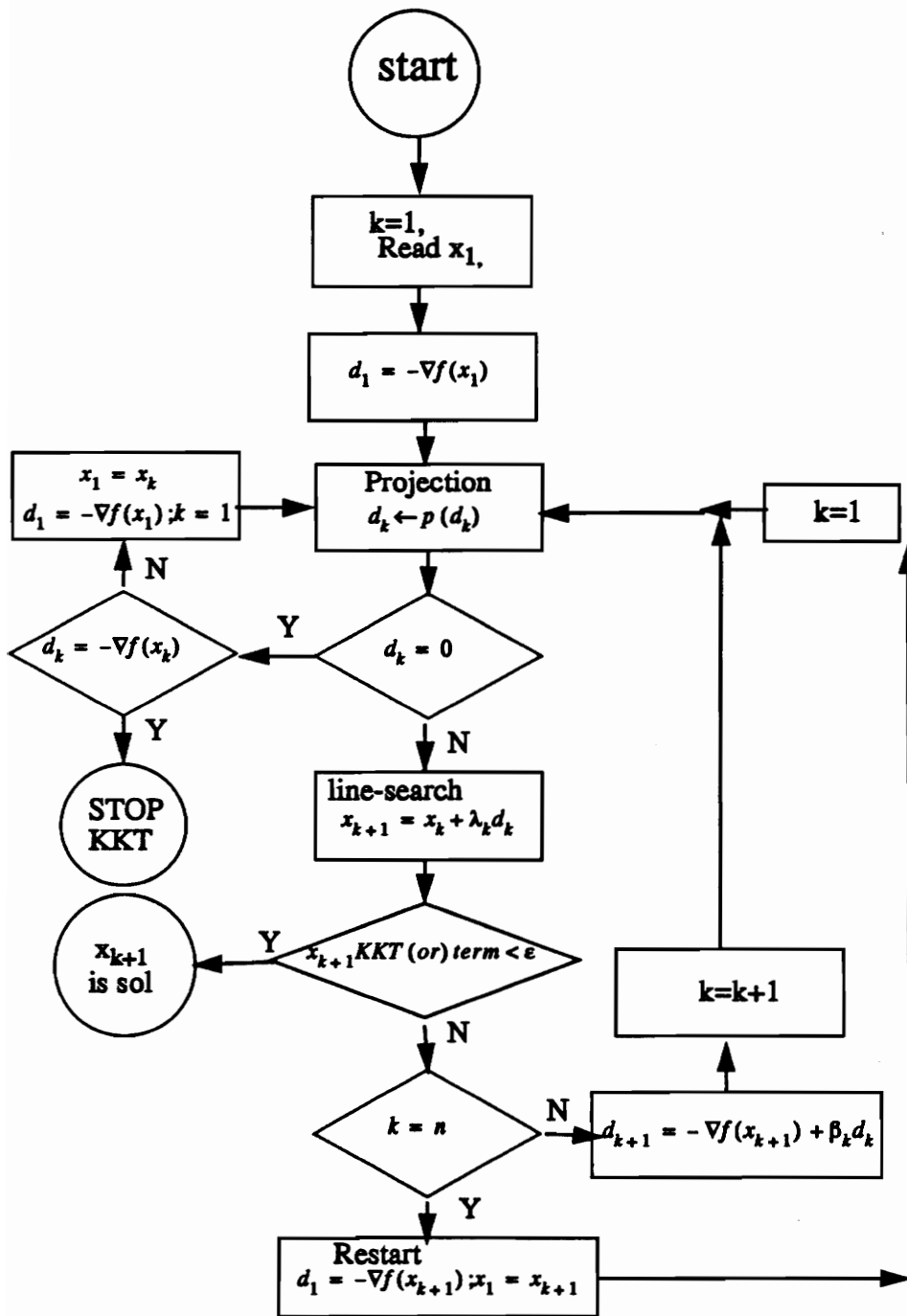


Figure (6.14); Conjugate-gradient method.

Chapter 7

Conclusions and Future Research

The research work reported here involves two distinct phases: 1) Analysis and modelling of the problem and 2) Development of effective solution procedures. Models have been developed for both the single and multiple transmitter location problems. These models are based on a propagation prediction procedure to determine the path-loss at each of the receiver locations.

A set of suitable solution procedures has also been designed for the proposed problems. The structure of the modelled problem was used to modify various competitive, standard existing solution procedures, in order to develop effective variants for our purpose.

A real-life problem was used as a part of the test in the verification phase. An AutoCAD map of Whittemore Hall at Virginia Tech was used to create a database containing the information needed for propagation prediction. A software tool based on a d^n [Seidel, S.Y., Rappaport, T.S., (1991)] propagation prediction technique was developed. This test problem structure was then used to investigate various modelling techniques and the effect of different modelling parameters.

It can be concluded from the research in this thesis that we now have an adequate model and effective solution procedures for the problem at hand. The model adopts a grid to create a description of the design space to the accuracy needed. This feature is used in the solution procedure to enhance its convergence and computational competitiveness. The

solution procedure starts by solving the problem with a sparse grid and refines the solution using denser grids. The weights in the model can be used to incorporate desired priorities and densities of receiver distribution across the design space. The penalty and threshold parameters can be used to incorporate the constraints that define the coverage properties of the design. The model employs a combination of two objectives to be achieved during the design of the system, minimax and minisum. A convex combination of these two objectives is used to trade-off between their effects. The location of the transmitter can be restricted to a desirable feasible region within the design space.

Also, we now have a numerically stable, computationally competitive and accurate solution procedure that can use the foregoing model to solve the engineering problem at hand. These solution procedures have been tested on a variety of test problems to ensure that they are robust with respect to the changes in propagation prediction procedures. The results indicate that overall, the starting solution scheme that has been prescribed along with the grid-size variation scheme used in concert with the proposed Hooke & Jeeves or Conjugate gradient search direction strategies appear to compose a viable algorithm procedure for the problem at hand. The In-Exact Quadratic interpolation line-search is the best line-search procedure in combination with the proposed solution procedures. The use of grid size variation is help full for computational competitiveness and essential for good convergence properties in the absence of a good starting solution. The variable target value method can be revisited for comparison when problems of higher dimensions are being solved. The RSM procedure with its latest developments can be used effectively in the case of a stochastic propagation prediction procedure.

Further research extensions, beyond this thesis, can focus on the following topics.

1. Incorporation of other design efficiency describers such as delay spread, and other quality of coverage describers into the objective function.

2. The cell division procedure is dependent on the resource allocation procedures (frequencies or codes) used. The proposed model can be extended to be sensitive to these resource allocation procedures. A more descriptive model of this kind will be helpful in designing a reliable system.

3. In the proposed model, the demand has been treated as being static. When the number of users are large, and mobile, a “probable distribution of receivers at any given time” approach of the type used here is valid, but when the number of users is small, varying loads on the cells, and varying loads at different places within a cell, cannot be accurately modelled as a static demand. The present model can be extended to model a system having very few users with dynamically varying priorities (if the use of a CAD system to design such a small system can be justified).

4. More complex propagation prediction procedures using ray-tracing algorithms and considering diffraction, effect of building materials, reflections, etc., can be used in the model instead of the simple d^n model. Also, after the work on interfacing the AutoCAD input with the propagation prediction software is available, more complex outdoor and indoor situations can be used to test this tool.

5. Further research extensions of this model to deal with complex feasible regions.

6. The Response Surface Methodology that has been described in this thesis can be enhanced, tested and fine tuned for accommodating stochastic propagation prediction procedures (when they are developed).

7. The models and the solution procedure can be experimentally verified using measure-

ments.

8. The procedures and the evaluation of the objective function mentioned in this thesis can be easily parallelized for computational superiority. Software work can also concentrate on interfacing this tool with the propagation prediction tools available.

9. The present model solves an engineering design problem as an optimization problem. The model developed for this problem, and the solution methodologies designed, can find applications in other engineering design problems in the fields of Aerospace, Civil, and Mechanical Engineering. Such extensions can also be explored.

References

- [1] Bazaraa, M.S., Sherali, H.D., and Shetty, C.M., **Nonlinear Programming: Theory and Algorithms**, John Wiley & Sons, Inc., New York, NY, Second edition., 1993.
- [2] Box, G.E.P., Draper, N.R., **Empirical Model-Building and Response Surfaces**, John Wiley & Sons, Inc., 1987.
- [3] Camerini, P.M., L. Fratta, and F.Maffioli, "*On Improving Relaxation Methods by Modified Gradient Techniques*", **Mathematical Programming Study** 3, pp 26-34,1975.
- [4] Dixon, L.C.W., Ducksbury, P.G., and Singh, P., "*A New Three-Term Conjugate gradient Method*", **Journal of Optimization Theory and Applications**: Vol. 47, No. 3, Nov' 1985. 285-300
- [5] Francis, R.L., McGinnis, L.F., Jr., and White, J.A., **Facility Layout and Location: An Analytical Approach**, 2nd edition, Printice Hall, Englewood Cliffs, New Jersey, 1992.
- [6] Handler, G.Y., and Mirchandani, P.B., "*Location on Networks*", The MIT Press, Cambridge, Massachusetts, 1979.
- [7] Joshi, S., Sherali, H.D., Tew, J.D., "*An Enhanced Response Surface Methodology Algorithm Using Gradient-Deflection and Second-Order Search Strategies*", **Research Notes, Department of Industrial and Systems Engineering, V.P.I & S.U**, 1993
- [8] Khuri, A.I., Cornell, J.A., **Response Surfaces Designs and Analysis**, Marcel Dekker, Inc., 270 Madision Avenue, New York, New York 10016, 1987.
- [9] Luenberger, D.G., **Introduction to Linear and Nonlinear Programming**, 2nd Edition, Addison - Wesley, Amsterdam, 1984.
- [10] Nordai, F.L., "*Balanced, Capacitated, Location-Allocation Problems on Networks*

- with a Continuum of Demand*". Doctoral Dissertation, V.P.I & S.U, 1985.
- [11] Powell, M.J.D., "*Restart procedures for the Conjugate gradient method*", **Mathematical Programming**, 12 (1977) 241-254, North-Holland Publishing Company.
- [12] Raymond H. M., **Response Surface Methodology**, Allyn and Bacon, Inc., Boston, 1976.
- [13] Seidel, S.Y., and Rappaport, T.S., "*914 MHz path-loss Prediction Models for Indoor Wireless Communications in Multi-floored Buildings*", **IEEE Transactions on Antennas and Propagation**, May 1991.
- [14] Shanno. D.F., "*Conjugate gradient Methods with Inexact Searches*", **Mathematics of Operations Research**, Vol3, No.3, August 1978 244-256.
- [15] Sheldon H. J., Lee W. S., "*Techniques for Simulation Response Optimization*", **Operations Research Letters**, 8 (1989) 1-9, North-Holland.
- [16] Sherali, H.D., and Pendyala C.M., "*Non differentiable Optimization (software)*", **Research Notes, Department of Industrial and Systems Engineering, V.P.I & S.U**, 1994.
- [17] Choi, G., and Sherali, H.D., "*Non differentiable Optimization*", **Research Notes, Department of Industrial and Systems Engineering, V.P.I & S.U**, 1992.
- [18] Sherali, H.D., and Osman, U., "*Conjugate gradient Methods Using quasi-Newton Updates with Inexact line-searches*", **Journal of Mathematical Analysis and Applications**, 150(2), 359-377, 1990.
- [19] Sherali, H.D., and Ullular, O., "*A Primal-Dual Conjugate subgradient Algorithm for Specially Structured Linear and Convex Programming Problems*", **Applied Mathematics and Optimization**, 20: 359-377, 1989.
- [20] Sherali, H.D., and Tuncbilek, C. H., "*A Squared Euclidean Distance Location-Allocation Problem*", **Naval Research Logistics**, 1992, Vol 39, pp. 447-469.
- [21] Tansel, B.C., Francis, R.L., and Lowe, T.J., "*Location on Networks: A Survey Part I: The p-center and p-median Problems*", **Management Science**, Vol. 29, pp. 482-497, 1983,a.

- [22] Tansel, B.C., Francis, R.L., and Lowe, T.J., "*Location on Networks: A Survey Part II: Exploiting Tree Network Structure*", *Management Science*, Vol. 29, pp. 497-511, 1983,b.
- [23] Tuncbileck, C.H., "*A Squared Euclidean Distance Location-Allocation Problem*", Master's Thesis, VPI & S.U, 1990.
- [24] William, E.B., H.Tamer.O., "*Optimization of Simulation Responses in the Multi-Computing Environment*", *Proceedings of the 1987 Winter Simulation Conference*, 1987, pp. 402-408.

Vita

Mr. Chandra M. Pendyala was born on sixth August 1970 in India and acquired his bachelor's degree in Mechanical Engineering from the "Jawaharlal Nehru Technological University" (J.N.T.U) - Hyderabad, India in 1991. He worked with Dr. Hanif D. Sherali from the Department of Industrial and Systems Engineering and the Mobile and Portable Radio Research Group from the Department of Electrical Engineering at Virginia Polytechnic Institute and State University for a Master's in Industrial and Systems Engineering. Mr. Pendyala is going to work at Ryder Dedicated Logistics starting July 1994.



Chandra M. Pendyala