

# Clustering sensors in wireless *ad hoc* networks using dynamic expected coverage model

by

**Dipeshkumar J Patel**

Department of Industrial Engineering

University at Buffalo

State University of New York

Thesis submitted to the faculty of the Graduate School  
of the University of Buffalo in partial fulfillment  
of the requirements for the degree of  
Master of Science  
Feb 2003

# Abstract

Multisensor Data Fusion deals with combining data from sensors spread over a geographically dispersed area. The sensors are mobile and change location with time. In order to collect data and process the data from these sensors, an equally flexible network of fusion beds (ClusterHeads) is required. Also the links between the sensors and clusterheads are prone to enemy attack and jamming or foliage effects may exist. In this work we develop a Mixed Integer Linear Programming model to determine the optimal clusterhead location strategy that maximizes the expected data covered over a period of time. The model is termed as Dynamic MEXCLP (Maximum Expected Covering Location Problem). Also the network of clusterheads can reorganise itself from time to time. Two greedy heuristics are proposed to solve the Dynamic MEXCLP and their worst case behavior is presented. Also a Column Generation technique is adopted to improve the heuristic solution for cases where solution quality is the prime concern. Computational experiments show that the Column Generation approach in conjunction with the heuristics performs much better than CPLEX in terms of solution time. Also the optimality gap is less than 5%.

# Dedication

To my parents

## Acknowledgment

I would like to express my sincere gratitude to my advisors, Dr. Rajan Batta and Dr. Rakesh Nagi for their constant support, guidance and patience. I am really thankful for their direction, devotion and encouragement which kept me focused on my work.

I would also like to thank Dr. James Llinas and the Center for Multisource Information Fusion for their extended support and cooperation.

I wish to thank my friends for making the lab and UB experience an enjoyable one. Avijit, Abhay, Hrishikesh and Mohan deserve a special thanks for their support at every stage of my thesis.

Finally, I would like to thank my brother, sister-in-law, sister and brother-in-law for their support and inspiration without which I would not have accomplished this.

# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>x</b>
<b>1 Introduction and Literature Review</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Literature review of Covering Models . . . . .	5
1.2.1 Deterministic Covering Models . . . . .	5
1.2.2 Stochastic Covering Models . . . . .	9
1.3 Literature Review of Dynamic Models . . . . .	12
1.3.1 Dynamic Plant Layout Problem . . . . .	13
1.3.2 Stochastic $P$ -Median Problem . . . . .	16
1.4 Literature review on Wireless <i>Ad hoc</i> Networks . . . . .	21
1.4.1 Network Topology . . . . .	21
1.4.2 Mobility or Location Management . . . . .	24
1.4.3 Routing Management . . . . .	24
1.4.4 Clustering Algorithms . . . . .	26
1.5 Thesis Objective . . . . .	29
1.6 Thesis Organization . . . . .	29
<b>2 Problem Description</b>	<b>31</b>
2.1 Introduction . . . . .	31
2.2 MEXCLP . . . . .	34

2.3	Problem Description and Formulation . . . . .	36
2.3.1	Dynamic MEXCLP with Relocation . . . . .	37
2.3.2	Dynamic MEXCLP with restriction on the maximum number of relocations for the entire time horizon . . . . .	41
2.3.3	Dynamic MEXCLP with location dependent relocation cost . . . . .	42
2.3.4	Dynamic MEXCLP for Clustering with Mobile Facilities . . . . .	43
2.4	Chapter Summary . . . . .	45
<b>3</b>	<b>Solution Methodology: A Heuristic Approach</b>	<b>46</b>
3.1	Relocation Heuristic . . . . .	47
3.2	No Relocation Heuristic . . . . .	50
3.3	Dynamic Programming Approach . . . . .	52
3.4	Chapter Summary . . . . .	54
<b>4</b>	<b>Solution Methodology: A Column Generation Approach</b>	<b>56</b>
4.1	Introduction . . . . .	56
4.2	Initial Basic Feasible Solution . . . . .	57
4.3	Column Generation Formulation-I (CG-I) . . . . .	58
4.3.1	Master Problem . . . . .	58
4.3.2	Sub-Problem . . . . .	59
4.4	Column Generation Formulation - II (CG-II) . . . . .	60
4.4.1	Master Problem . . . . .	60
4.4.2	Sub-Problem . . . . .	61
4.5	Strategies . . . . .	62
4.5.1	Strategy I . . . . .	62
4.5.2	Strategy II . . . . .	63
4.6	Solving Integer Master Problem . . . . .	64
4.7	Termination Criteria . . . . .	65
4.8	Chapter Summary . . . . .	66

<b>5</b>	<b>Computational Experiments and Detailed Numerical Analysis</b>	<b>67</b>
5.1	Input Parameters and Software Implementation . . . . .	68
5.1.1	Input Parameters . . . . .	68
5.1.2	Software Implementation . . . . .	69
5.2	Detailed Analysis of a scenario . . . . .	72
5.2.1	Effect of varying $p$ on the model behavior . . . . .	72
5.2.2	Effect of varying $C$ on the model behavior . . . . .	76
5.2.3	Effect of varying $n$ on the model behavior . . . . .	76
5.2.4	CPLEX solution within CG solution time . . . . .	77
5.2.5	Improvement in solution quality using CG over RH and NRH . . . . .	77
5.3	Analysis of Variance . . . . .	80
5.3.1	Screening Experiment: Fractional Factorial Design . . . . .	81
5.3.2	Parametric Experiment . . . . .	84
5.3.3	Discussion . . . . .	92
5.3.4	Computational Results for large size problem . . . . .	94
5.4	Regret Analysis . . . . .	95
5.5	Chapter Summary . . . . .	97
<b>6</b>	<b>Conclusion and Future Research</b>	<b>98</b>

# List of Figures

1.1	3-tier Hierarchical Network Architecture . . . . .	22
1.2	Flat Network Architecture . . . . .	23
2.1	Typical Scenario . . . . .	31
2.2	A snapshot of the network . . . . .	38
2.3	A snapshot of the network . . . . .	44
3.1	An example illustrating the worst case behaviour of RH . . . . .	48
3.2	An example illustrating the worst case behaviour of NRH . . . . .	51
3.3	State space diagram: Dynamic Programming . . . . .	54
4.1	Column Generation Flow . . . . .	57
4.2	Strategy I . . . . .	63
4.3	Strategy II . . . . .	65
4.4	State Space Diagram: Integer Master Problem . . . . .	66
5.1	Software Implementation Flow Chart . . . . .	71
5.2	Effect of varying $p$ on objective function value . . . . .	74
5.3	Effect of varying $p$ on solution time . . . . .	75
5.4	Effect of varying $C$ on objective function value . . . . .	77
5.5	Effect of varying $C$ on solution time for different values of $p$ . . . . .	78
5.6	Effect of varying $n$ on solution time . . . . .	79
5.7	CG vs CPLEX: Deviation . . . . .	85
5.8	CG vs CPLEX: Time . . . . .	86



5.9	CG Main Effect Plot for Deviation . . . . .	87
5.10	CG Interaction Plot for Deviation . . . . .	88
5.11	CG Main Effect Plot for Solution Time . . . . .	89
5.12	CG Interaction Plot for Solution Time . . . . .	90
5.13	CPLEX Main Effect Plot for Deviation . . . . .	91
5.14	CPLEX Interaction Plot for Deviation . . . . .	92
5.15	CPLEX Main Effect Plot for Solution Time . . . . .	93
5.16	CPLEX Interaction Plot for Solution Time . . . . .	93

# List of Tables

5.1	Range of parameters . . . . .	73
5.2	Effect of varying $p$ on objective function value . . . . .	73
5.3	Effect of varying $p$ on solution time . . . . .	75
5.4	CPLEX objective function value within CG solution time . . . . .	79
5.5	Comparison between RH, NRH and CG solution . . . . .	80
5.6	Fractional Factorial Design: Parameter Value . . . . .	82
5.7	Fractional Factorial Design: Data Matrix (Randomized) . . . . .	83
5.8	Parametric Experiment: Factor Levels and Value . . . . .	85
5.9	Significant Factors and Interactions: CG Deviation . . . . .	87
5.10	Significant Factors and Interactions: CG Time . . . . .	88
5.11	Significant Factors and Interactions: CPLEX Deviation . . . . .	90
5.12	Significant Factors and Interactions: CPLEX Time . . . . .	91
5.13	Computational Results . . . . .	94
5.14	Range of parameters for regret analysis . . . . .	96
5.15	Objective function table . . . . .	96
5.16	Regret table . . . . .	96

# Chapter 1

## Introduction and Literature Review

### 1.1 Introduction

Location of a facility is a critical strategic planning decision that plays a key role in determining the success of an industry or a service provider. It is also a crucial aspect for efficient utilization of resources. Some common examples of facility location problems encountered in practice are a retail chain siting a new outlet, a manufacturer locating a new warehouse and a city planner determining locations for schools, emergency facilities like firestations, ambulance bases, etc. An efficient location of a facility immensely helps a service provider to gain advantage over its competitors.

Location problems discussed in the literature are classified according to various objectives as follows:

1. Center Problem
2. Median Problem
3. Covering Problem

The objective of the Center Problem is to open a given number of facilities and assign each demand node to exactly one of them such that the maximum distance (weighted or unweighted) from any open facility to any of the demand nodes assigned

to it is minimized. Locating a fire extinguisher in a manufacturing facility constitutes a facility location problem with center objective.

The objective of the Median Problem is to locate a given number of facilities such that the sum of the weighted distances (or travel times) between demand nodes and the closest facility is minimized.

In some location problems selecting locations which minimize the average distance travelled or maximum distance travelled may not be appropriate. The critical nature of the demands in such problems require the service to be available within a prespecified time or to be within a threshold distance. To locate such facilities, the primary issue is “coverage”. A demand node is said to be covered if it lies within a threshold distance (or time) of at least one facility. Location of emergency facilities like fire engines or ambulances is typically modeled as a covering problem.

Covering location problems are classified as:

1. Set Covering Location Problem (SCLP): The objective of the SCLP is to minimize the total number of facilities to be located such that all the demand nodes are within a threshold distance of the closest facility. Toregas et al. [62] was the first to introduce the SCLP.
2. Maximal Covering Location Problem (MCLP): The objective of the MCLP is to maximize the number of demand nodes covered with a limited number of resources (facilities). Church and Reville [26] introduced the MCLP.

All covering problems encountered in the literature fall in the genre of either SCLP or MCLP. All models cited in the literature are basically variant of the SCLP or the MCLP. SCLP and MCLP have also been classified according to the deterministic and stochastic nature of the problem. Toregas [62] and Church and Reville [26] deal with the deterministic version of the SCLP and MCLP respectively. Chapman and White [20] introduced the stochastic SCLP considering the unavailability of facilities at certain times. Daskin [29] considered the probability of facility being unable to serve a demand node at all times and modeled the Maximum Expected Covering

Location Problem (MEXCLP) as a variant of the MCLP to maximize the expected demand covered.

In this work, a critical scenario encountered in a military application is modeled as a variant of the Maximal covering location problem. The application deals with gathering intelligence. One of the key techniques to achieve intelligence is Data Fusion. Data Fusion deals with combining information from various sources to determine the situation or threat in the system.

Data Fusion can be formally defined as “a multilevel, multifaceted process dealing with detection, association, correlation, estimation and combination of data and information from multiple sources to achieve refined state and identity estimation, and complete and timely assessments of situation and threat” [1].

The combination of data from multiple sensors is termed as MultiSensor Data Fusion Process. The fusion of data from multiple sensors facilitates a better judgment of the scenario since we get multiple input to verify the authenticity of the information. Now these sensors (sources of information) can lie in different geographical regions or at a distance from each other. In order to fuse data (information) from all these sensors, the data has to be sent to the location where it is going to be fused.

In a non-wartime situation with all the sensors being static and the fusion location fixed, the data transfer can take place through wired networks. But during wartime or for battlefield surveillance, all the sensors may be fixed or mobile and the fusion bed (consists of fusion processors, which in most cases is an Airborne Warning and Control System (AWACS) that performs the fusion operation) is also mobile. Thus the network has to rely on wireless transfer of information. Since there exists no fixed infrastructure or predetermined connectivity between sensors and the AWACS, the network could be considered as a wireless *ad hoc* network. *Ad hoc* networks are described in detail in Section 1.4.

In this work, we are basically interested in the logistics of fusion process: How the data transfer for the fusion process will be carried out? And where will the fusion take place? Our work is to facilitate the fusion process by providing a robust

architecture for performing fusion. The decision variables are the locations of the fusion processors, such as AWACS surveillance aircrafts etc. (We will address them as Cluster Heads.) In a situation like this, we know that all the entities or sources of data (radars, sonars, soldiers, etc.) are mobile and hence we need an equally flexible architecture in order to collect data from all these sources. The position or location of all the entities is assumed to be known at all times.

The obvious medium of data transfer is a wireless communication network. Wireless communication is characterized by transmission range, available bandwidth, etc. Since the sensors are mobile they may move out of each others range or the bandwidth restrictions might disrupt or disconnect the communication link between them. In addition to this a communication link is also prone to enemy attack. Hence a communication link has an associated probability of jamming, failure, or there might exist a foliage/terrain effect. The challenge here is to ensure that no data is lost and it becomes absolutely necessary to “multiply cover” the sensors.

Typically sources of information such as radars, soldiers, etc. lie on the ground, or are under water in the case of sonars. The facilities with processing power such as AWACS or surveillance aircrafts are at a higher altitude. AWACS also act as sources of information.

The objective of this work is to develop an optimal strategy and solution for locating a given number of AWACS (Cluster Head) such that maximum information can be gathered and processed from the sensors under hostile conditions.

Given the above motivation, we model the problem under consideration as a covering problem due to its inherent structure, which maximizes the expected demand covered with a given number of cluster heads. A distinctive characteristic of this problem is that all the entities are mobile. Additionally the links between the facilities (Cluster heads) and “demand” nodes (sensors) are prone to failure. We incorporate these features in our model and hence this work can be viewed as a variant of the work of Daskin [29] due to the following reasons:

1. Daskin considers facility failure as opposed to link failure.

2. Daskin considers a static scenario as opposed to a dynamic scenario, i.e., the demand nodes were considered to be stationary.

In the following section, a comprehensive review of the covering location problem, wireless *ad hoc* networks and dynamic and stochastic models encountered in other applications like plant layout and  $p$ -median problem is provided.

## 1.2 Literature review of Covering Models

Covering location problem is a genre of location problem based on the notion of “coverage”. A node is said to be covered, if it lies within an acceptable distance of at least one facility or it can be served within a prespecified time. The quality of the service required or the criticality of the service governs the threshold distance (or time). The covering location problem addressed in the literature are mainly based on the following basic models: (1) Set Covering Location Problem (SCLP) and (2) Maximal Covering Location Problem (MCLP). In SCLP, the objective is to cover all demand with least number of facilities. Whereas in MCLP, the objective is to cover maximum demand with a restricted number of facilities. In majority of the literature both demand nodes and potential facility locations are a discrete set of points.

We can broadly classify the Covering literature into the following two categories:

1. Deterministic Covering Models
2. Stochastic Covering Models

### 1.2.1 Deterministic Covering Models

Covering models considering a deterministic scenario where all the input parameters are known comprise this section.

The earliest work in the area of covering is dealt by Toregas et al. [62]. They model the location of emergency service facilities as a set covering location problem with equal costs in the objective function. Here the authors consider the maximum

distance or time that separates the demand node from its closest service facility as a crucial parameter. Location of fire stations in a city is such an example. Other examples are location of ordinary service facilities such as schools, libraries. The objective is to locate minimum number of facilities such that the maximum response time for attending any demand node is less than a specified threshold and every demand node will be attended by at least one facility. The solution to this problem indicates both, the number and the location of the facilities that provide the desired service. In their paper, to achieve a better problem structure the authors assume that the demand nodes and potential facility locations are finite points in a plane and facilities can only be placed at demand node locations. The authors also assume that the distance and minimum response time between every pair of points in the plane are known. A standard linear programming code has been used to solve the problem with the addition of cuts in case of fractional results.

Later in 1974, Church and Reville [26] look at the covering model from a different perspective where there is a restriction on the number of facilities to be located. The objective of the model proposed is to cover maximum number of demand nodes with limited resources (fixed number of facilities) such that the covered node is within the desired service distance ( $S$ ) of its closest facility. The authors designate the problem as Maximal Covering Location Problem (MCLP). They also consider a small variant of the MCLP by introducing the mandatory closeness constraint to additionally ensure that no demand node is farther than a distance  $T$  ( $T > S$ ) to its closest facility. The lesser the difference between  $T$  and  $S$ , the fairer the solution is to the demands not covered within  $S$ . In this paper the authors adopt linear programming solution techniques with Branch and Bound to get optimal integer solutions. Two heuristics called Greedy Adding (GA) Algorithm and Greedy Adding with Substitution (GAS) Algorithm are also employed. The GA first picks a facility which covers the maximum demand, then picks the one which covers the most demand not covered by the previous facilities and continues until all the facilities are located. This algorithm does not guarantee an optimal solution, since once the facility is selected it cannot be replaced.



The GAS builds on GA with the only exception that GAS at each step tries to improve the solution by replacing each facility, one at a time, with a facility at another free site. If an improvement is possible, the pair of facility sites which provides the greatest improvement in the objective function is chosen.

In order to overcome the uncertainty of a facility being operative at all times, the concept of multiple coverage comes into the picture. One of the early works in this area is cited in Hogan and Reville in [35]. In this paper, the authors introduce the concept of multiple coverage in the context of classic covering models: the set covering location problem and the maximal covering location problem. The maximal backup coverage problem is modeled as a multi objective formulation in this work. The authors define backup coverage as the coverage of demand points by two or more facilities and tradeoff the coverage as defined in SCLP and MCLP against backup coverage. Two Models, viz. BACOP1 and BACOP2 have been described by the authors. In BACOP1 first coverage is mandatory for all nodes, but the focus is on first redundant coverage of a node. BACOP2 trades off first coverage against backup coverage. Their main result is the fact that without substantial loss of first coverage, significant levels of backup coverage within a system can be achieved.

Earlier work in covering literature assumed that the demand points are covered if a single unit of service is available within the desired service distance or time of its location. In the paper [11], Batta et al. reconsiders the SCLP and the MCLP with multiple units required by the demand nodes. Their work can be viewed as a generalization of the concept of backup coverage by Hogan and Reville [35]. Here the authors model separate formulations for SCLP and MCLP. Also an important criterion considered in modeling was that the demand which requires greater number of units to respond is more critical and hence the model requires the closest service facility to be closer than the one which requires less units. The authors first transform the problem into a 0-1 integer program and then apply implicit enumeration branch and bound algorithm to solve the problem.

The MCLP has wide use in selecting locations for facilities, especially for problems

involving the public sector services, such as fire, police, ambulance and educational services. An implicit assumption of the MCLP is that, the demand nodes not covered by a facility are not served. This is an impractical assumption especially in emergency service sectors. Church et al. [25] overcome this assumption and also consider the distance traveled to cover these demand nodes in their formulation. The authors formulate the problem as a two-objective location covering problem which directly considers the travel distance that uncovered demand must traverse to reach its nearest facility. A Lagrangian relaxation method was used to solve the problem.

Moon and Chaudhry in [46] introduce an additional aspect to the set covering location problem of providing a secondary coverage to the facilities providing primary coverage to the demand nodes. They term the problem as a conditional covering problem (CCP). The CCP requires facilities to be located such that all demand nodes are covered within an acceptable distance or time of the facility. Also any two facilities are no farther away than a specified distance. The authors model the problem as an Integer Program and solve the relaxed linear program with cut constraints. Later in 1987, Chaudhry et al. [22] proposed seven greedy heuristics for solving the CCP.

Revelle et al. in [56] extend the notion of CCP [46]. They introduce three models in the paper: Maximal Conditional Covering Problem (MCCP I and MCCP II) and Multiobjective Conditional Covering Problem (MOCCP). The objective of MCCP is to locate a given number of facilities to maximize the facilities which are themselves covered by another facility within a prespecified distance. In MCCP the supporting coverage is not mandatory unlike in CCP, but the demand nodes do require primary coverage. The MCCP I *prevents* the supporting facility to be located at the same location, whereas MCCP II permits the supporting facility to be stationed at the same node. The MOCCP relaxes the constraint that all nodes should have a primary coverage. MOCCP is a two-objective problem which sees the tradeoff of primary coverage to the demand nodes against the secondary coverage of the facilities. They model both MCCP and MOCCP as a Linear Integer Program and solve the relaxed LP using a Branch and Bound procedure.

Berman and Krass [14] relax the assumption made in traditional MCLP that coverage of a demand node is binary (i.e., either fully covered or not covered at all). The authors introduce a generalized version of MCLP (GMCLP) with partial coverage of demand nodes. They assume that for each demand node, a multiple set of coverage levels, with corresponding coverage radii are specified. They also assume that the coverage level is a decreasing step function of the distance to the closest demand node. The GMCLP is shown to be equivalent to the Uncapacitated Facility Location Problem (UFLP). Several Integer Programming formulations are developed by the authors capitalizing on the special structure of the GMCLP. Computational results show that small instances of the problem are the toughest to solve. However solvability improves with increase in the number of demand nodes.

Brotcorne et al. [18] develop five heuristics for a special case of large scale covering location problems, where the set of demand point is discrete and the set of potential locations for the facilities is continuous.

### 1.2.2 Stochastic Covering Models

Most of the location theory papers like the ones described above deal with deterministic conditions. Stochastic Covering Problems takes into account the uncertainty of a facility working condition and models the covering problem such that a more realistic and reliable system is developed.

Chapman and White [20] were the first to consider the unavailability of the facilities covering the region. They came up with the probabilistic version of SCLP. The model ensure that the probability of a demand node being served by at least one facility is greater than or equal to a specified reliability level  $\alpha$ .

Later Daskin [29] introduced a variant of the MCLP that considers the possibility that facilities may be unable to respond to demand at all times. In both SCLP and MCLP, it was assumed that the facilities will be able to provide service to all the demand nodes such that they are within the desired distance  $S$ . This paper relaxes the above assumption and associates a probability of a facility being operative. In

MEXCLP (Maximum Expected Covering Location Problem), every demand node will have potentially more than one facility to cover itself. In other words, not all facilities will be able to respond to demands at all times, i.e., we need to consider the probability of a region being covered. Here the author assumes that the probability  $p$  of a facility working is independent of other facilities and  $p$  is same for all facilities. A single node substitution heuristic algorithm was proposed for solving the MEXCLP.

Later in 1989 Batta et al. [10] relaxes three of the assumptions made in the MEXCLP model viz. facilities operate independently, facilities have the same working probabilities and the probabilities are invariant with respect to their locations. This paper discusses three techniques of improving upon the solutions proposed by Daskin: 1) use the post-IP analysis procedure on Daskin's result; 2) use the adjusted MEXCLP (AMEXCLP) objective function in Daskin's Procedure; and 3) use the hypercube optimization procedure.

The covering literature which considers the unavailability of certain facilities for service due to facility failing or facility serving some other customer is termed as Reliability Covering Problem (RCP). Ball et al. [5] consider the RCP in which the given routes service various stops (e.g., in a transportation system). They address reliability with respect to possible route failures in a covering context. The objective of their work is to find the probability that all stops will be covered by an operating route, given that the routes are subjected to failure. The authors show that the RCP is  $NP$ -hard on both directed and undirected networks. Some polynomially solvable cases are developed by the authors when some additional structure is imposed on the routes of the tree.

Ball and Lin [4] models the emergency service vehicle location problem as a 0-1 integer programming formulation. The model explicitly captures the stochastic nature of the problem by trying to optimize the reliability of the system, where the vehicle will not be available to respond to all demand calls within the specified (acceptable) time. The authors propose valid inequalities as a preprocessing technique to augment the IP. A Branch and Bound procedure is used to solve the IP. The computational

results show that the preprocessing techniques are highly effective.

Marianov and Reville in [40] develop a queuing model for the probabilistic set covering location problem considering the unavailability of the facilities to respond to demand (when they are serving some other customer) at certain times. The objective of the model is to minimize the total number of facilities required to cover all demand with a minimum reliability  $\alpha$ . The paper explicitly considers the dependence of the probabilities of facilities being busy, when the facilities are in the same region. For each region the author models the behaviour as an  $M/M/s$ -loss queuing system ( a Poisson arrival, exponentially distributed service time,  $s$  servers, loss system). They solve the relaxed LP using a branch and bound procedure.

Marianov and Serra [41] model the probabilistic MCLP with a restriction on the maximum waiting time for any demand or the maximum queue length. In the paper, two models are presented:

1. Queuing Maximal Covering Location-Allocation Model (QM-CLAM): The objective is to locate  $p$  centers to maximize the demand node covered, where each covered node is within a prespecified distance (or time) and no demand from the covered node

(a) has to wait more than the maximum waiting time  $\tau$ , or

(b) is accompanied by more than  $b$  demand in the queue,

with a probability of at least  $\alpha$ . Here the number of facilities per center is restricted to one.

2. QM-CLAM with co-location of  $m$  servers per center: Here each center consists of  $m$  servers.

They model the problem as a 0-1 integer program and heuristic solutions are also presented.

Melachrinoudis and Helander in [42] considers the problem of locating a single facility on an undirected tree with  $n$  nodes in the presence of unreliable edges. The edges fail with a probability of  $(1 - p_e)$ . The authors assume that the probability of failure of the edges are independent of each other and the nodes are perfectly reliable. The objective of this work is to find a network location that maximizes the expected number of nodes reachable by operational paths from a given service facility. A decomposition formula is developed by the authors. They show that for a linear graph with equal edge probabilities, the median node will be the optimal facility location. The authors also present two polynomial time algorithms for this problem. The first algorithm is a label correcting procedure, based on modification to the Floyd-Warshall all pairs shortest path algorithm. It has a running time of  $O(n^3)$ . The second algorithm is of  $O(n^2)$  that is based on a depth-first node traversal and takes advantage of the tree structure. The polynomial algorithms are possible because of the uniqueness of paths in trees and also because the problem is restricted to locating a single facility. They also provide sensitivity analysis ranges and analytically derive marginal values for  $p_e$ . The authors conclude that for multiple facility locations in a cyclic network the problem becomes very complex.

A detailed review on covering literature can be found in [48, 59, 65].

### 1.3 Literature Review of Dynamic Models

In this review, we will cite some of the work conducted in other location siting problems, where dynamic or stochastic aspects are considered to incorporate the real world scenario. Dynamic models incorporate the deterministic changes that are going to occur over a period of time. Whereas stochastic models capture the uncertainty involved in one or more parameters on which the models depend. Here a detailed review of some of the dynamic and stochastic models in the following areas is carried out.

1. The Dynamic Plant Layout Problem and

## 2. The Stochastic $P$ -Median Problem

### 1.3.1 Dynamic Plant Layout Problem

Before introducing the dynamic plant layout problem, let us explain the static plant layout problem.

The Static Plant Layout Problem (SPLP) considers that the material flow among given entities is same for all periods and attempts to minimize the total material handling cost by assigning a given number of facilities to a set of locations. The SPLP is often formulated as a Quadratic Assignment Problem.

The SPLP has been formulated in Rosenblatt [57] as follows:

$$\begin{aligned} \text{Minimize } z_t &= \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n a_{ijkl} X_{ij} X_{kl} \\ \text{Subject to} \end{aligned}$$

$$\sum_{i=1}^n X_{ij} = 1 \quad j = 1, \dots, n, \quad (1.1)$$

$$\sum_{j=1}^n X_{ij} = 1 \quad i = 1, \dots, n, \quad (1.2)$$

$$X_{ij} \in \{0, 1\} \quad \forall i, j, \quad (1.3)$$

where

$$X_{ij} = \begin{cases} 1 & \text{if department } i \text{ is assigned to location } j \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ikjl} = \begin{cases} f_{ik} d_{jl} & \text{if } i \neq k \text{ or } j \neq l \\ f_{ii} d_{jj} + c_{ij} & \text{if } i = k \text{ and } j = l \end{cases}$$

$$c_{ij} = \text{cost per unit time associated directly with assigning department } i \text{ to location } j,$$

$$d_{jl} = \text{distance from location } j \text{ to location } l \text{ (travel cost between locations); where } d_{jj} = 0, \text{ and}$$

$$f_{ik} = \text{work flow from department } i \text{ to department } k.$$

Constraint (1.1) ensures that only one department is assigned to location  $j$ . Constraint (1.2) ensures that the facility is assigned to only one location. Rosenblatt [57] introduced the Dynamic Plant Layout Problem (DPLP), where the material flow is

different for each period. The author considered a deterministic environment where the material flow is known for each period of time. Given that the locations of facilities is allowed to change, the DPLP model considers tradeoff between the material handling cost and the relocation cost. A dynamic programming formulation is developed to determine the optimal layout for each period of the time horizon.

Let  $Z_{t,k}$  be the material handling costs for layout  $A_k$  in time period  $t$ . Here the author finds the best  $R_t$  solutions for each time period  $t$  such that  $K = Z^{UB} - Z^{inf}$  and  $R_t$  is given by  $Z_{t,R_t} - Z_{t,1} \leq K$  and  $Z_{t,R_t+1} - Z_{t,1} > K$  where  $Z^{UB}$  is upper bound and  $Z^{inf}$  is the lower bound determined by summing the costs obtained by solving the SPLP in each period without considering rearrangement cost. The paper discusses techniques to determine simple upper bounds by:

1. Keeping the same solution throughout the time horizon, or
2. Determining the solution obtained by finding the optimal layout for each time period using SPLP and adding the cost of rearrangement from one layout to another.

The author finds the best  $R_t$  solutions by introducing an additional constraint (1.4) to the previous best solution.

$$\sum_{(i,j) \in A_k} X_{ij} - \sum_{(i,j) \in B_k} X_{ij} \leq n - 2 \quad (1.4)$$

where  $A_k = \{(i, j) : X_{ij} = 1\}$  and  $B_k = \{(i, j) : X_{ij} = 0\}$

Constraint (1.4) forces the new layout to differ from the previous layout by at least changing the assignment of two facilities.

$Z_{t,2}$  is found by adding constraint (1.4) to the SPLP formulation with solution set of  $Z_{t,1}$ ,  $Z_{t,3}$  is found by adding constraint (1.4) to the SPLP formulation with solution set of  $Z_{t,2}$  and so on.

Each time period ' $t$ ' forms a 'stage' having  $R_t$  states, each state corresponding to a



different layout. Both optimal and heuristic solution procedures are discussed using dynamic programming. The issue of rolling horizon is also addressed.

Batta [9] establishes a class of best possible upper bounds for the DPLP described in [57]. The author shows that if the same layout is used for all periods, then the problem reduces to solving a single SPLP where the flows between any two facilities for each periods are added up to obtain the resultant (corresponding) flow.

Balakrishnan et al. [3] considers the dynamic plant layout problem (DPLP) with the additional constraint on the total budget for shifting facilities. The authors term the problem as Constrained Dynamic Plant Layout Problem (CDPLP). They formulate this problem as a Quadratic Assignment Problem similar to the SPLP. Also the authors model the CDPLP as a single Constrained Shortest Path Problem (CSP) and compare the CSP algorithm with the Dynamic Programming (DP) approach used in previous works in this area. The results in the paper shows that in most cases the CSP algorithm performs better than the DP algorithm.

Montreuil and Laforge in [45] incorporate the probabilistic nature of the future requirements in the Dynamic Layout Design model. The model requires the user to input the scenario tree of the probable future. For each future, the designer has to specify the shape requirements of the cells, the material flow between the cells and the cost of relocating the facilities from the current future to the probable future. The authors formulate the problem as a linear program.

Kouvelis et al. [39] aims at developing robust layouts for single and multiple period layout planning. The authors propose algorithms for generating robust layout designs for both single and multiple periods taking uncertainty in product mix as well as production volumes into consideration. For solving the single period layout problem a simple modification to the branch and bound procedure is suggested, whereas for large multi-period dynamic layout design problem, a heuristic approach is proposed.

Palekar et al. [49] deals with the dynamic plant layout problem (DPLP) by explicitly capturing the stochastic nature of the demand in the future periods. They model the Stochastic Dynamic Plant Layout Problem (SDPLP) where the Inter-

departmental Flow Matrix (IDFM) at any time depends on the previous states. The model minimizes the expected material handling cost and the relocation cost of the facilities for the time horizon. The authors also propose an exact and heuristic solution procedure for solving the SDPLP.

In [27], Conway and Venkataramanan extend the work of Rosenblatt [57] and Balakrishnan [3] by introducing the budget constraint for each period in the multi-period dynamic plant layout problem. The authors propose a genetic algorithm based solution technique for the multi-constrained DPLP.

Yang and Peters [66] model the DPLP considering a rolling horizon planning time window to obtain a robust layout design in a flexible manufacturing system. An efficient heuristic solution procedure is also proposed.

Kochhar and Heragu [38] provide a genetic algorithm based heuristic for the dynamic plant layout problem for two consecutive planning periods. The algorithm, Dynamic Heuristically Operated Placement Evolution (DHOPE) attempts to find the layout for the next period, given the current layout with the objective of minimizing the rearrangement cost and the total material handling cost.

Baykasoglu and Gindy [12] solves the DPLP problem using a simulated annealing based procedure.

### 1.3.2 Stochastic $P$ -Median Problem

In this section, some of the stochastic models applied to the classical  $p$ -median location problem are explored.

The  $p$ -median problem is a class of location problems where the objective is to locate  $p$ -facilities (medians) such that the sum of weighted distance (or travel time) between demand nodes and the closest facility is minimized. The  $p$ -median problem was introduced by Hakimi [33, 34]. Hakimi [34] proved an important property that one of the optimal solution of the  $p$ -median problem consist of locating the facilities only on the nodes of the network.

Mirchandani and Odoni [43] extended the  $p$ -median problem for networks where

link traversal time are assumed to be a random variable with a known discrete probability distribution over a finite set of values including infinity representing link failure. Since there is a finite number of identifiable potential sites for facility location, the authors formulate the stochastic median location problem as an integer linear program. They show the existence of an optimal solution under a reasonable set of assumptions: and thereby corroborates Hakimi's [33] result. The authors further determine that if the utility function for travel time is convex and non-increasing, at least one set of expected optimal  $k$ -medians exists on the nodes of a network (oriented and non-oriented).

Weaver and Church [64] formulate the stochastic  $p$ -median location problem as follows:

$$\begin{aligned} & \text{Minimize } \sum_{k=1}^s \sum_{i=1}^n \sum_{j=1}^n Q_k a_{ik} d_{ijk} x_{ijk} \\ & \text{Subject to} \end{aligned}$$

$$\sum_{j=1}^n x_{ijk} = 1 \quad \forall i, k, \quad (1.5)$$

$$\sum_{j=1}^n y_j = p, \quad (1.6)$$

$$x_{ijk} \leq y_j \quad \forall i, j, k, \quad (1.7)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j, k, \quad (1.8)$$

$$y_j \in \{0, 1\} \quad \forall j, \quad (1.9)$$

where

$$\begin{aligned} a_{ik} &= \text{the demand at node } i \text{ in state } k, \\ Q_k &= \text{the probability of being in state } k, \\ s &= \text{the number of states of the network}, \\ n &= \text{the number of demand nodes}, \\ d_{ijk} &= \text{the distance (travel time) from node } i \text{ to node } j \text{ in state } k, \\ x_{ijk} &= \begin{cases} 1 & \text{if demand node } i \text{ is assigned to a facility at node } j \\ & \text{in system state } k \\ 0 & \text{otherwise,} \end{cases} \\ y_j &= \begin{cases} 1 & \text{if there is a facility located at node } j \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The objective of the above formulation is to maximize the sum of the expected weighted distance from facility to the demand nodes. Constraint (1.5) ensures for a given state, the demand node  $i$  is assigned to only one facility. The restriction on the number of facilities is given by constraint (1.6). Constraint (1.7) prevents demand nodes to be assigned to locations where no facility is placed.

The authors [64] provide computational procedures for solving the stochastic  $p$ -median problem. One of the solution procedure to generate an approximate solution is a generalization of the exchange heuristic developed by Teitz and Bart [60]. The exchange heuristic starts with an initial solution and exchanges a pair of nodes, one in the current solution and the other not in the current solution, which yields maximum improvement in the objective function value. This procedure is repeated until no such pair of nodes is found which improves the objective function value (solution). The authors also propose a bounding method based on Lagrangian relaxation with subgradient optimization. In case the sub-gradient procedure fails to terminate with an optimal solution for the primal, a branch and bound algorithm can be used with Lagrangian relaxation for obtaining lower bounds instead of the LP relaxation.

Berman and Odoni [16] relaxed the assumption that the facilities have to be located permanently at a given location and allowed the facilities to be relocated at a cost with change in the network states. The authors presents a more realistic version of the problem of locating facilities on a stochastic network where the travel time on a link is a random variable and relocation of facilities is allowed on a network in a reaction to the changes in the state of the network. Transition among the states of the network are assumed to be Markovian. Some of the assumptions made in the paper are:

1. Network is connected for all states and traversal time of all the links of the network takes a finite value for all the states.
2. Time intervals between any change in the state of the network are much longer than the service time on the network.

3. All facilities are available for service whenever a demand occurs.
4. Same set of locations are chosen every time the network is in a particular state  $s$ , independent of the previous states of the network. Later in the paper, the authors provide a counter example to this assumption.

The authors show that for a case of locating a single facility on a tree, the problem reduces to finding the median of the tree for any one state of the network. A heuristic algorithm is proposed for finding the location of a facility on a general network. The authors also develop simple bounds on the optimal value of the objective function for the multifacility and multistate problem.

Later Berman and LeBlanc [15] developed a heuristic for solving the stochastic  $p$ -median problem with relocation of facilities allowed with change in the state of the network. The heuristic, which runs in polynomial time, determines the optimal location of facilities for the multi-facility, multi-state problem. The authors compare the heuristic with linear programming formulation of the problem for a series of small test problems. For larger test problem, the solution provided by the heuristic is compared to several bounds.

Later Berman and Rahn timer [17] extend the work of Berman and Odoni [16] with the only modification that the location of the  $p$  facilities depends on the previous state unlike assumption 4 (mentioned previously) of [16]. The authors present an algorithm where complexity is reduced by taking advantage of the special structure of the problem.

Berman in [13] models the location of a single facility with the objective of minimizing the expected response time with the following uncertainty involved:

1. Travel times on the links of the network are discrete random variables with a known probability distribution.
2. Demands occurring in the network follow a homogeneous Poisson process.

In the paper, two models have been developed: 1) Stochastic Expected Loss Median

problem (SELM) and 2) Stochastic Expected Queue Median problem (SEQM). In SELM, the demands that arrive when the server is busy are lost at a positive cost. Whereas in SEQM, the demands that occur when the server is busy are queued and are served on an FCFS basis.

Carson and Batta [19] show for a case of locating an ambulance in a University that a significant amount of savings is achievable, when relocation of ambulances is allowed based on the temporal variation of demand. The performance measure adopted is to minimize the system-wide average response time to a call.

Vairaktarakis and Kouvelis [63] model the 1-median location problem on a tree network considering the dynamic aspects and/or uncertainty involved in the demands of the node and the length of the links'. The node demand and links length are either dynamic, i.e., a linear function of time or uncertain given by a finite number of scenarios. Several models have been presented by taking various combinations of: 1) Linear demand, 2) Stochastic demand with a) Linear link length or b) Stochastic link length. The authors model all the problems with the objective of minimax regret criteria (minimizing the maximum loss). Polynomial time algorithms for each case are also presented.

Averbakh and Berman [2] consider the 1-median problem with the uncertainty involved in the weights (demand) of the nodes of the network. The weight of each node is a random variable with an unknown probability distribution. An interval is known in which the weight of each node lies. The objective of the minmax regret model presented is to develop a solution which minimizes the worst case loss in the objective function for such a scenario. The paper also shows that the minmax regret model is polynomially solvable.

Current et al. [28] approaches the dynamic facility location problem with uncertainty involved in the total number of facilities to be located. In the paper, two different objectives are considered for modeling the  $p$ -median problem with NOFUN (Number Of Facilities Uncertain). They are: 1) Minimization of expected opportunity loss (EOL), and 2) Minimax regret, i.e., minimize the maximum loss.

The authors model the EOL problem as a 0-1 integer programming problem. The minimax regret criterion for the objective function value is also considered for the case where the probabilities of the various states are not known.

## 1.4 Literature review on Wireless *Ad hoc* Networks

In recent years, *Ad hoc* networking has gained enormous importance in the field of mobile computing. An *ad hoc* network is a self-organizing multi-hop wireless network, which relies neither on fixed infrastructure nor on predetermined connectivity. All the entities in an *ad hoc* network can be mobile. The communication between network components is carried over a wireless medium and the network topology changes depending on the node mobility. The main advantage of such networks are that they can be rapidly deployed and therefore applications of these are in situations which either lack fixed infrastructure or are at high risk; e.g. in military communications, disaster management, law enforcement, etc.

The issues in *ad hoc* networks [58] can be classified as follows:

1. Network Topology
2. Location Management
3. Routing Management

### 1.4.1 Network Topology

In a highly dynamic environment of an *ad hoc* network, proper design of network architecture is crucial to many related issues such as routing or location management. Typically, the topology in an *ad hoc* network is either flat or hierarchical.

#### **Hierarchical Architecture**

In a hierarchical topology, nodes are partitioned into groups called **clusters**. Within each cluster, a node is chosen to perform the function of a **cluster head**. Depending

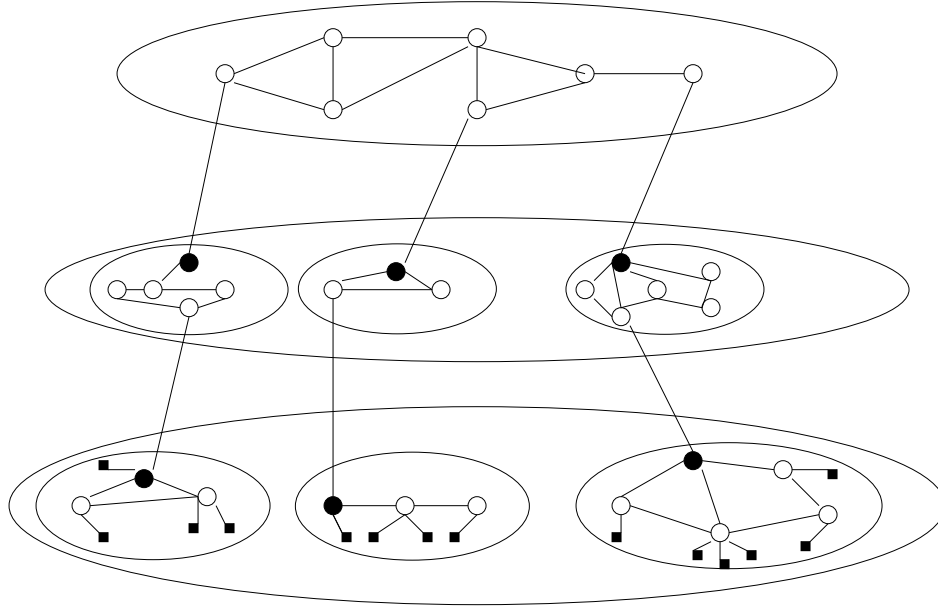


Figure 1.1: 3-tier Hierarchical Network Architecture

on the number of hierarchies, such networks can have one or more *tier* or *level*. Cluster head is responsible for keeping track of locations (also called *location management*) in its cluster. Also, routing between two nodes in different clusters are always through their respective cluster heads. A cluster head, on getting the message from the source node, passes it to a node in the higher level, which, in turn, sends it to its cluster head (if the message need to go upward) or to a neighboring node in the same cluster (if the message remains in the same level or go downward). Ramanathan and Steenstrup [55] presents Multimedia support for Mobile Wireless Networks (MMWN) based on hierarchical structure. MMWN is a modular system of distributed, autonomously adaptive algorithms that cooperate to support distributed, real-time multimedia applications in large, multihop mobile wireless networks. A typical hierarchical network is shown in Fig. 1.1.

The paper discusses the various processes involved in the MMWN such as cell formation, hierarchical clustering and cluster dynamics. A detailed review of the existing clustering algorithms is presented in Section 1.4.4



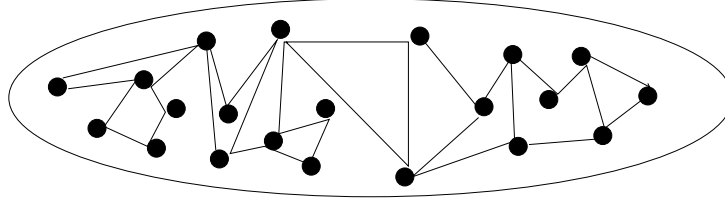


Figure 1.2: Flat Network Architecture

### Flat Architecture

In a flat architecture, all the nodes are equal and each of them act as a router. Connections are established between nodes which are in close proximity and routing is constrained only by connectivity conditions and possibly by security limitations. Fig. 1.2 shows a diagram of a flat network.

Haas and Tabrizi [32] presents an comparison between hierarchical and flat networks. The authors favores the flat architecture because of the following advantages:

1. In a flat network, the routing is optimal, and often more reliable since usually more than one path exist between source and destination nodes. In hierarchical networks, however, cluster heads are single points of failure and therefore these routes are more susceptible to attack.
2. Nodes in flat network transmit at a significantly lower power than the transmission power of cluster heads in hierarchical networks. This results in more network capacity, less expense in power and larger degree of **Low Probability of Interception/Low Probability of Detection**.
3. Also, in a flat network, no overhead is associated in dynamic addressing, cluster creation and maintenance and mobile location management.

On the other hand, flat networks are not scalable. Mobility management is simpler in a hierarchical network since the cluster head keeps the database containing locations of all nodes in its cluster.

### 1.4.2 Mobility or Location Management

Mobility or location management deals with keeping track of locations of all the mobile nodes within the network. To maintain current locations of each node in a network, it is necessary to maintain a large database with periodic or continuous updating. Location Management can be classified on the basis of this update into **Static** and **Dynamic** strategies. In static strategies, the location is updated at predetermined set of locations, whereas in dynamic strategies, the nodes (end-users) determine when an update should be generated based on its movement. Kasera and Ramanathan [37] considers the location management problem in a *hierarchically organized multi-hop wireless network* where all nodes (switches and end-users) are mobile. Cluster heads work as location managers for each cluster to maintain the association database and perform other functions. Their location management mainly deals with *location updating*, *location finding* and *switch mobility*. Location updates occur when one of the following events happen [37, 55]:

1. End-user reaffiliates to some other switch
2. The switch to which the end-user is affiliated moves to some other cluster
3. Cluster reformation such as cluster splitting or merging.

In a flat network, however, location management seems difficult since there is nothing which is equivalent to cluster heads. However, in [32], Haas and Tabrizi have made a case for flat networks with zone routing as described in the next section.

### 1.4.3 Routing Management

In *ad hoc* networks the routing has to be determined dynamically and the literature for routing protocols is divided into **Proactive** or **Table Driven Routing Protocol**, **Reactive** or **On-Demand Routing Protocol** and **Hybrid Protocol**, the last being a combination of the first two.

### **Proactive protocol**

In a proactive protocol, the route between each pair of nodes is continuously maintained in a tabular format. This table is updated on a continuous basis, recording the changes in the network topology. Thus the delay in determining the route is minimal, but maintaining the table is a costly affair which is wasteful for both time and bandwidth. Some of the protocols cited in the literature are:

1. Destination-sequenced Distance-Vector Routing (DSDV) [53]
2. Clusterhead Gateway Switch Routing (CSGR) [24]
3. Wireless Routing Protocol (WRP) [47]

### **Reactive protocol**

In a reactive protocol, routes are determined on a need to basis. That is, when a packet is to be delivered from source node to the destination node, a route discovery procedure is initiated and a route is determined through a global search procedure. This would result in large delays and are therefore unacceptable in applications where long routing delays cannot be allowed. Some of the protocols cited in the literature are:

1. Ad Hoc On-Demand Distance Vector Routing (AODV) [54]
2. Dynamic Source Routing (DSR) [36]
3. Associativity Based Routing (ABR) [61]
4. Temporally Ordered Routing Algorithm (TORA) [52]

### **Hybrid protocol**

Hybrid protocols combines the advantages of both reactive and proactive protocols. Haas [31] proposed Zone Routing Protocol (ZRP), a hybrid protocol based on the notion of routing zone. In this protocol, each node pro-actively determines the routes

between itself and nodes within its routing zone. Thus whenever a demand arises with little effort the route can be determined among the node not in the routing zone. Some of the advantages of ZRP are:

1. It requires only a relatively small number of query messages, as these messages are routed only to peripheral nodes, omitting all other nodes within the routing zones.
2. As the zone radius is significantly smaller than the network radius, the cost of updating the routing information for the zone topologies is small compared to a global proactive mechanism.
3. ZRP is much faster than a global reactive route discovery mechanism, as the number of nodes queried is very small compared to the global flooding process.
4. It discovers multiple routes to the destination.
5. The path determined by ZRP needs less number of hops and therefore is more stable.

#### **1.4.4 Clustering Algorithms**

Clustering of nodes in *ad hoc* networks are done in order to use the wireless resources efficiently by reducing congestion and for proper location and routing management. Various algorithms has been proposed in the literature for clustering in *ad hoc* networks. Most common features considered by the clustering algorithms are:

1. Stability: The node mobility should not cause frequent changes in clusterhead assignment and clusterhead should be comparatively less mobile.
2. Load Balancing: The cluster should neither be densely populated nor scarcely populated.
3. Battery Power: Cluster head consumes more power than other nodes. Thus the solution should not cause excessive drainage of some nodes than others.

4. **Transmission Range and Signal Strength:** Clusterhead should have sufficient transmission range and signal power to reach all the nodes in its cluster.

The clustering algorithms can be broadly classified as Graph based and Geographical based clustering. Graph based heuristics view the network as a graph. Whereas Geographical based heuristics uses Global Positioning System (GPS) to accurately determine the location and velocity of the nodes in the network. In general, the message cost of maintaining a cluster is better for graph-based clustering, whereas the number of nodes without a clusterhead is smaller for geographical clustering. Graphical clustering, however, is more suitable when nodes have widely varying transmission ranges.

### **Graph based Clustering**

Since the clusterhead selection is an NP-hard problem [7], all existing solutions available are based on heuristic approaches.

- **Highest Degree Heuristic:** Degree of a node is defined as the number of nodes within its transmission range. The degree based heuristic is a modified version of [51]. The node with highest degree is selected as clusterhead and with all the nodes within its transmission range forms the cluster. The process continues until all nodes are assigned to a clusterhead. Load balancing is poor but the stability of the cluster is good.
- **Lowest ID Heuristic:** Each node has a unique identifier. In this heuristic, the node with lowest id is selected as clusterhead [30] and all nodes within its transmission range forms the cluster. The process continues until all nodes has a designated clusterhead. Performance is better than highest-degree heuristic [24]. Excessive drainage of lower id nodes is found.
- **Node Weight Heuristic:** In this heuristic Basagni [6] assigns a weight of  $-v$  to a node with a speed of  $v$  units. The selection criteria is same as the highest

degree heuristic. A stable solution is obtained, i.e. number of clusterhead reassignment is small. No other features are captured in the heuristic.

- **Weight Based Clustering Algorithm:** Chatterjee et al. [21] propose a weight-based clustering algorithm where the weight of each node is updated periodically. Here the authors assigns weight to each of the feature such as Load balancing, Battery Power, Signal Strength and Mobility. The solution obtained from this algorithm has less reassignment of clusterheads and flexibility of changing weight to give more weightage to certain features than others.
- **MOBIC Clustering Algorithm:** Basu et. al. [8] propose a distributed clustering algorithm called *MOBIC* for mobile ad hoc networks (MANET). This is based on relative mobility metric for clusterhead selection. All nodes send a “Hello” message to all its neighbors and each of the node finds its relative mobility metric by a formula using the signal strength of the received message. Each node shares its mobility metric with other nodes in its transmission range and the node with least mobility metric assumes the position of clusterhead. The results in the paper shows that the algorithm has 33% lower rate of clusterhead change.

### Geographical based Clustering

This algorithm uses GPS (*Global Positioning Systems*) to find the latitude, longitude and velocity of the nodes and uses these information to form clusters based on the spatial density. The clusters look like rectangular boxes (defined by grids) unlike traditional clustering. The clusterhead is elected among the centrally located members of each cluster. The cluster reformation is inevitable (necessary) due to node mobility.

The clustering algorithm works using a 2-stage process [23], where the first stage, **Central Periodic Clustering Algorithm** is a periodic procedure to form clusters based on spatial density for the entire network. This stage is carried out by a central

global manager. This stage is divided into

1. Box Generation: Divides the region into small boxes.
2. Box Size Refinement: Checks the size of box and restrict the ratio of length to breadth of the boxes to 1.5.
3. Node Density Adjustment: Removes boxes with no nodes in it and merges boxes with less nodes and splits boxes with high node density.

The second stage, **Cluster Maintenance** is a maintenance algorithm executed locally within each cluster between two executions of the periodic protocol. The outcomes of the algorithm includes changes in cluster membership, clusterhead responsibility, merging and splitting based on spatial density, etc.

## 1.5 Thesis Objective

The focus of the thesis is to design a flexible network architecture that can reorganize itself to capture the dynamics of sensor (demand node) movement for efficient data fusion. The objectives of the thesis is:

1. To propose a mathematical model for finding the optimal location strategies for the clusterheads over the entire time horizon.
2. To capture the mobility of sensors and the reliability issues (probability of link failure) using a Mixed Integer Linear Programming (MILP) model.
3. To develop solution methods that can perform well in terms of both solution quality and time.

## 1.6 Thesis Organization

Chapter 2 is comprised of detailed description of the problem followed by a Mixed Integer Linear Program (MILP) formulation. Several variants of the model have

been developed to capture varying characteristics of the system. In Chapter 3, two heuristics are presented. Also a Dynamic Programming based approach is explored. A Column Generation approach is developed to solve the Dynamic MEXCLP in Chapter 4. In Chapter 5 the software implementation for the solution methodology is discussed and the results are presented. Conclusion and directions for future work is discussed in Chapter 6.



# Chapter 2

## Problem Description

### 2.1 Introduction

At the very outset, let us describe a battlefield scenario which forms the basis of this work. In a battlefield the entities on the ground are radars, soldiers, tanks, etc. Entities under water include sonars installed on ships, submarines etc. which are used by them to locate targets. Unmanned Aerial Vehicles (UAVs) are examples of airborne entities that are used for the purpose of surveillance. All these entities are sources of information that could be of crucial importance in a wartime scenario. These entities are represented as black dots located on the lower plane in Fig 2.1.

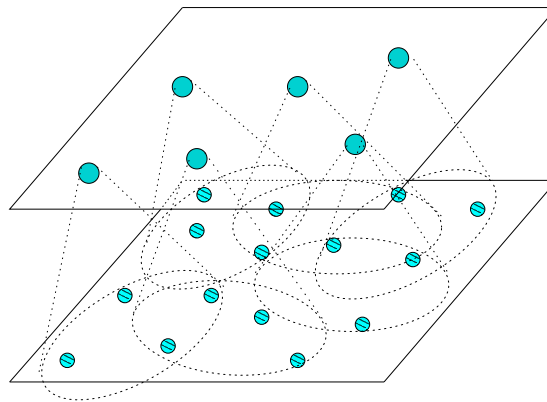


Figure 2.1: Typical Scenario

Let us define these entities as *sensors* for future reference. The information generated by all these sensors is useful to recognize the threats to the system and to aid in

the purpose of gathering military intelligence. The gathering of information from the sensors is achieved by the process of Data Fusion. Typically there are two strategies that are used for data fusion:

1. Centralized data fusion
2. Decentralized data fusion

In Centralized data fusion, data from all the sensors are fused at a single processor and then the information is shared by all the entities of the system. Whereas, in decentralized data fusion, data from a group of sensors called a cluster are processed together at one Fusion Processor Airborne Warning and Control System (AWACS) called **clusterhead** and the fused data (information) is transmitted to other clusterheads. It should be noted here that the sensors are not necessarily in close proximity of one another. Hence a pertinent question is: where should the data fusion take place? This question serves as the motivation of this work. Also in a battlefield scenario the sensors could be possibly mobile. An implication of this fact is that the data communication takes place through wireless rather than wired networks. This gives rise to additional challenges like network reliability in the system.

The fusion is typically carried out by Airborne Warning and Control System (AWACS). AWACS are typically aircrafts that possess fusion processors to carry out the data fusion process. These AWACS are represented by the black dots on the higher plane in Fig. 2.1. It should be noted that though the sensors and AWACS are located on the lower and upper plane respectively in Fig. 2.1, this is not necessary. It is just for the purpose of visualization of the problem scenario.

The objective of the work is to locate these AWACS such that maximum data is gathered from the sensors. In this work it has been assumed that the AWACS are located at discrete points. Each AWACS covers a set of sensors depending on its location and transmission range. The sensors covered by an AWACS can be considered to form a *cluster* with the AWACS as its *clusterhead*. Henceforth we use the term clusterhead (CH) or facility instead of AWACS.

The data transfer takes place through wireless medium of communication. Typical issues related to wireless communication are transmission range, available bandwidth, etc. Since the sensors are mobile they may move out of the facilities' (CH) range or the bandwidth restrictions might disrupt or disconnect the communication link between them. In addition to this, a communication link is prone to enemy attack. Hence a communication link has an associated probability of jamming, failure, or there might exist a foilage effect. Thus it is extremely desirable that each sensor should have multiple coverage, i.e. each sensor should be covered by more than one clusterhead. This ensures maximum network reliability in the event of breakdown of the communication link between a sensor and its clusterhead due to hostile jamming, weather conditions, etc. In case of a catastrophic failure of a communication link, the sensors could retain the capability to switch to a backup clusterhead.

The basic objective of the work is to locate AWACS such that the maximum data is collected from the sensors. Additionally all the sensors are capable of moving and they can change their position with time in order to perform the task assigned to them. Since the sensors are mobile, relocation of AWACS is necessary to achieve maximum coverage of sensors data. However to ensure a stable location strategy over the entire "time horizon", we either restrict the maximum number of relocations for the entire time horizon or associate a cost for every relocation of AWACS (termed as relocation cost). Thus we consider a tradeoff between data coverage and relocation cost. Our objective is to achieve this tradeoff over a fixed time period, addressed previously as time horizon. The time horizon is split up into discrete time periods of equal length. Relocation of AWACS is permitted only at the beginning of these time periods.

In order to achieve the tradeoff between coverage and relocation cost over the entire time horizon and to ensure network reliability, we model the problem as a covering location problem with the objective of maximizing the expected demand covered by locating a given number of facilities (CH).

In later sections, we formulate our problem as an integer program, since our prob-

lem structure closely resembles the Maximum Expected Covering Location Problem (MEXCLP) addressed widely in the OR literature. Our work evolves from the traditional MEXCLP model. In the following section, we present a detailed description and analysis of the MEXCLP.

## 2.2 MEXCLP

The traditional Maximal Covering Location Problem (MCLP) assumed that the facilities are perfectly reliable and are able to serve demands at all times. But this assumption is not practical. Daskin [29] modeled the Maximum Expected Covering Location Problem (MEXCLP) considering the unavailability of servers to serve all demands at all times. In MEXCLP, Daskin associates each facility with a probability  $p$  of being inoperative. The model assumes that the probabilities of the facilities not working are independent of each other and are same for all facilities.

The objective of MEXCLP is to maximize the expected demand covered by locating a given number of facilities. In MEXCLP the number of facilities working follows a binomial distribution and the probability of a node  $k$  being covered is given by

$$\begin{aligned} &= 1 - \text{Prob}(\text{node } k \text{ not being covered}) \\ &= 1 - \text{Prob}(m \text{ facilities are not working}) \\ &= 1 - p^m \end{aligned}$$

where  $m$  is the number of facilities covering node  $k$ .

Let  $H_{k,m}$  be the random variable denoting the number of demands at node  $k$  covered by a working facility, given that  $m$  facilities are capable of covering node  $k$ .

Hence we have:

$$H_{k,m} = \begin{cases} d_k & \text{with probability } 1 - p^m \\ 0 & \text{with probability } p^m, \end{cases}$$

$$\text{and } E(H_{k,m}) = d_k(1 - p^m) \quad \forall \quad k, m.$$

If the number of facilities covering node  $k$  increases from  $m - 1$  to  $m$ , the corresponding increase in the expected coverage of node  $k$  is given by

$$\begin{aligned} \Delta E(H_{k,m}) &= E(H_{k,m}) - E(H_{k,m-1}) \\ &= d_k p^{m-1}(1 - p) \quad \forall \quad m = 1, 2, \dots, n \end{aligned}$$

We now define the variables and their corresponding indices, utilized in the MEX-

CLP formulation of [29], as follows:

$$\begin{aligned}
i &= \text{index for potential facility locations,} & i = 1, \dots, N, \\
k &= \text{index for demand nodes,} & k = 1, \dots, N, \\
N &= \text{number of demand nodes,} \\
n &= \text{number of facilities to be located,} \\
D &= \text{the distance beyond which a demand node is considered} \\
&\quad \text{“uncovered”,} \\
D_{ik} &= \text{distance between potential facility location } i \text{ and demand} \\
&\quad \text{node } k, \\
d_k &= \text{demand of node } k, \\
p &= \text{probability of a facility failure } (0 < p < 1), \\
r_{ik} &= \begin{cases} 1, & \text{if } D_{ik} < D \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

The decision variables of the problem are:

$$\begin{aligned}
x_i &= \text{number of facilities placed at location } i, \\
y_{jk} &= \begin{cases} 1, & \text{if demand node } k \text{ is covered by at least } j \text{ facilities} \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

The MEXCLP [29] is formulated as follows:

$$\text{Maximize} \quad \sum_{k=1}^N \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jk}$$

subject to

$$\sum_{j=1}^n y_{jk} - \sum_{i=1}^N r_{ik} x_i \leq 0 \quad \forall \quad k = 1, \dots, N, \tag{2.1}$$

$$\sum_{i=1}^N x_i \leq n, \tag{2.2}$$

$$x_i \in Z^+ \quad \forall \quad i = 1, \dots, N, \tag{2.3}$$

$$y_{jk} \in \{0, 1\} \quad \forall \quad j = 1, \dots, n, k = 1, \dots, N. \tag{2.4}$$

The objective function maximizes the total expected coverage. The inner summation in the objective function represents the number of demands that are covered by at least  $j$  facilities in which the term  $(1-p)p^{j-1}$  represents the weight associated with the number of demands covered by at least  $j$  facilities for any demand node  $k$ .

The objective function is concave in  $j$  for each  $k$ . If node  $k$  is covered by  $m$  facilities, constraint (2.1) assigns each of the variables  $y_{1k}, y_{2k}, \dots, y_{mk}$  a value of 1 since the objective function is a maximization function containing the term  $y_{jk}$ . Constraint (2.2) restricts the maximum number of facilities to be located. Constraint (2.3) is an integer constraint for the number of facilities allowed to be located at location  $i$ . Constraint (2.4) is a binary constraint.

## 2.3 Problem Description and Formulation

We consider the MEXCLP from the viewpoint of communication networks. In communication networks, facilities and the links between facilities and the demand nodes are prone to failure. However in this work, we assume that the facilities are perfectly reliable. Hence we consider the coverage problem with a probability of link failure. Alike traditional MCLP, MEXCLP also assumes that the demand nodes are static and the locations of all demand nodes are known. Here we allow the demand nodes to be mobile and assume that the velocity vector of all demand nodes is known and hence we have *a priori* knowledge about the exact location of demand nodes at any given time. From this aspect our work can be viewed as a dynamic variant of the MEXCLP model proposed by Daskin. The motivation for this assumption arises due to situations encountered in a military scenario as described in Sections 1.1 and 2.1.

A thorough analysis revealed that under a given set of assumptions, the objective function structure for link failure is similar to the objective function structure of the MEXCLP model where facility failure is considered. Here follows an argument to support the claim.

Let us assume that a facility and a demand node are connected by at most a single link. Let  $p$  be the probability of failure of a link. We assume that the probabilities of link failure are independent of each other and are same for every link. The expected demand coverage of a node  $k$ , given that there exists  $m$  links is given by  $d_k(1 - p^m)$ . Further if we consider the facility failure probability  $q$ , given  $m$  facilities that cover

demand node  $k$ , the expected demand coverage is  $d_k(1 - q^m)$ . Thus we observe that the structure of the objective function is similar under the assumption that all failure probabilities (link or facility) are independent of each other and same for all links (or facilities).

We propose several models incorporating the dynamic nature into the MEXCLP. We term this model as the dynamic MEXCLP. The basic objective of the model is to maximize the expected demand covered by locating a given number of facilities over a specific time horizon allowing relocation of facilities from one time period to next and minimizing relocation costs.

The following assumptions are made in our model:

1. Location of the sensors are known at any given instance.
2. Potential clusterhead locations constitutes a set of discrete points.
3. Relocation of facilities takes place at discrete time periods.
4. The clusterheads are perfectly reliable.
5. The probability of failure of all links is the same.

### **2.3.1 Dynamic MEXCLP with Relocation**

We note that the optimal location of facilities that maximizes the expected coverage in one time period may not necessarily be optimal in the next time period. The relocation of facilities increases the expected coverage that would otherwise be obtained without considering relocation of facilities. Hence in this model we allow the facilities to change location with time at an additional cost. Here a tradeoff between relocation cost and the increase in coverage achieved by relocating the facilities is considered.

We now formulate the problem of determining the optimal locations of facilities with relocation allowed at discrete time periods. Let us define the parameters and their corresponding indices as follows:

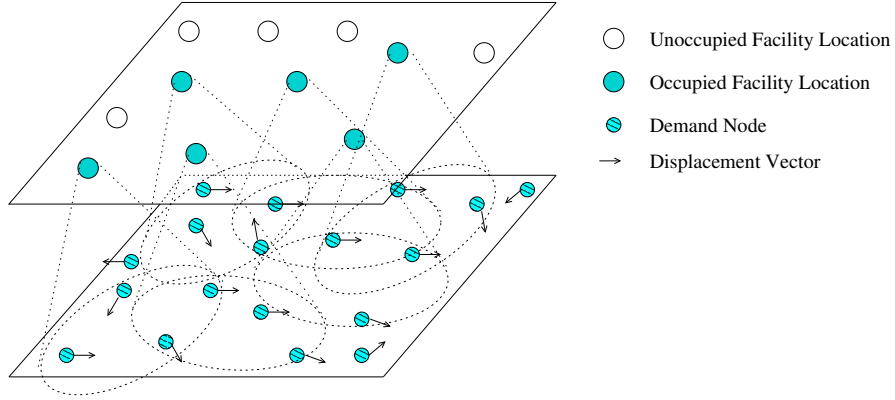


Figure 2.2: A snapshot of the network

$\Delta$  = set of potential facility locations.

$\Theta$  = set of demand nodes.

$n$  = maximum number of facilities to be located.

$T$  = maximum number of time periods in the horizon under consideration.

$U$  = the distance beyond which a demand node is considered “uncovered”.

$D_{ikt}$  = distance between potential facility location  $i$  and demand node  $k$  at time  $t$ .

$d_k$  = demand per period of node  $k$ .

$p$  = probability of a link failure per period (between any facility and demand node). ( $0 < p < 1$ )

$r_{ikt} = \begin{cases} 1, & \text{if } D_{ikt} < U. \\ 0 & \text{otherwise.} \end{cases}$

$C$  = cost per unit change in the number of facilities at any location  $i$  (one-half of relocation cost).

The decision variables of the problem are:



$$x_{it} = \begin{cases} 1, & \text{if a facility is placed at location } i \text{ at time } t \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{jkt} = \begin{cases} 1, & \text{if demand node } k \text{ is covered by at least } j \text{ facilities at time } t \\ 0 & \text{otherwise.} \end{cases}$$

$$w_{it} = \text{positive difference in the number of facilities at location } i \text{ between time } t-1 \text{ and time } t$$

The dynamic MEXCLP with relocation is formulated as follows:

$$\text{Maximize} \quad \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} - \sum_{t=1}^T \sum_{i \in \Delta} C w_{it},$$

subject to:

$$\sum_{j=1}^n y_{jkt} - \sum_{i \in \Delta} r_{ikt} x_{it} \leq 0 \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T, \quad (2.5)$$

$$\sum_{i \in \Delta} x_{it} \leq n \quad \forall \quad t = 0, \dots, T, \quad (2.6)$$

$$w_{it} \geq x_{it-1} - x_{it} \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T, \quad (2.7)$$

$$w_{it} \geq x_{it} - x_{it-1} \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T, \quad (2.8)$$

$$x_{it} \in \{0, 1\} \quad \forall \quad i = 1, \dots, |\Delta|, t = 0, \dots, T, \quad (2.9)$$

$$w_{it} \geq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T, \quad (2.10)$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \dots, n, k = 1, \dots, |\Theta|, t = 0, \dots, T. \quad (2.11)$$

The objective function maximizes the expected demand covered allowing relocation of facilities with time. If node  $k$  is covered by  $m$  facilities at time  $t$ , constraint (2.5) assigns each of the variables  $y_{1kt}, y_{2kt}, \dots, y_{mkt}$  a value of 1 since the objective function is a maximization function containing the term  $y_{jkt}$ . Constraint (2.6) restricts the maximum number of facilities to be located to  $n$  for any time  $t$ . Constraints (2.7) and (2.8) determine the positive difference between the number of facilities located at location  $i$  between time  $t-1$  and  $t$ . Constraint (2.9) is a binary constraint determining whether a facility is located at location  $i$ . Constraint (2.10) is a non-negativity constraint. Constraint (2.11) restricts the variable  $y_{jkt}$  to take a value between 0 and 1. We now present the following observations:

**Observation 1:**

In the formulation variables  $x_{it}$  are restricted to take only binary values. Constraints (2.7) and (2.8) together represent  $w_{it} = |x_{it} - x_{it-1}|$ . Since  $x_{it}$  and  $x_{it-1}$  are both integers  $w_{it}$  assumes an integer value at optimality.

**Observation 2:**

In the formulation, variables  $x_{it}$  assume only binary values. Hence the term  $\sum_{i \in \Delta} r_{ikt} x_{it}$  is also an integer since  $r_{ikt}$  is a constant which takes a value of 0 or 1 as defined earlier. Thus we have:

$$\sum_{j=1}^n y_{jkt} \leq \sum_{i \in \Delta} r_{ikt} x_{it} \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T \quad (2.12)$$

$$\text{or, } \sum_{j=1}^n y_{jkt} \leq m \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T \quad (2.13)$$

where  $m$  is an integer for any particular  $k$  and  $t$ .

For a given value of  $x_{it}$ , the problem turns out to be

$$\text{Maximize } \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n a_j d_k y_{jkt},$$

subject to:

$$\sum_{j=1}^n y_{jkt} \leq m \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T \quad (2.14)$$

where  $a_j = (1-p)p^{j-1} \quad \forall \quad j = 1, 2, \dots, n$ .

For a particular values of  $k$  and  $t$ , the above problem is separable into knapsack problems of the following form:

$$\text{Maximize } a_1 y_{1kt} + a_2 y_{2kt} + a_3 y_{3kt} + a_4 y_{4kt} + \dots + a_n y_{nkt}$$

subject to

$$y_{1kt} + y_{2kt} + y_{3kt} + y_{4kt} + \dots + y_{nkt} \leq m. \quad (2.15)$$

Since  $0 < p < 1$  its obvious that  $a_1 > a_2 > a_3 > a_4 > \dots > a_n$  and hence at optimality the first  $m$  variables ( $y_{1kt}, y_{2kt}, \dots, y_{mkt}$ ) will assume the value of 1 and the remaining  $n - m$  variables ( $y_{m+1kt}, \dots, y_{nkt}$ ) are set to 0. Thus even though the variables  $y_{jkt}$  are continuous variables between 0 and 1, at optimality  $y_{jkt}$  assumes binary values.

### 2.3.2 Dynamic MEXCLP with restriction on the maximum number of relocations for the entire time horizon

In this variant of the Dynamic MEXCLP, instead of having a relocation cost, we consider a restriction on the maximum number of facility relocations allowed for the entire time horizon. This is a practical assumption, since many applications may require a stable solution which provides a good coverage for a longer period of time. We now formulate the dynamic MEXCLP with restrictions on the maximum number of relocations as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} \\ \text{subject to} \quad & \end{aligned}$$

$$\sum_{j=1}^n y_{jkt} - \sum_{i \in \Delta} r_{ikt} x_{it} \leq 0 \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T, \quad (2.16)$$

$$\sum_{i \in \Delta} x_{it} \leq n \quad \forall \quad t = 0, \dots, T, \quad (2.17)$$

$$w_{it} \geq x_{it-1} - x_{it} \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T, \quad (2.18)$$

$$w_{it} \geq x_{it} - x_{it-1} \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T, \quad (2.19)$$

$$\sum_{t=1}^T \sum_{i \in \Delta} w_{it} \leq 2H, \quad (2.20)$$

$$x_{it} \in \{0, 1\} \quad \forall \quad i = 1, \dots, |\Delta|, t = 0, \dots, T, \quad (2.21)$$

$$w_{it} \geq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T, \quad (2.22)$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \dots, n, k = 1, \dots, |\Theta|, t = 0, \dots, T. \quad (2.23)$$

Constraint (2.20) restricts the maximum allowable number of relocations for the entire time horizon to  $H$ , where

$H$  = Total number of facilities allowed to relocate for the entire time horizon.

The factor 2 arises because each relocation causes a change in the number of facilities at 2 different locations in consecutive time periods.

### 2.3.3 Dynamic MEXCLP with location dependent relocation cost

Here, we consider an even more realistic situation where the relocation cost varies with the pair of locations between which the given facility changes its position. In this model, we have different indices for each of the facilities and we exactly know which facility is located at which location. Here we can have facilities of different characteristics.

We now define some additional parameters used in this model, and their corresponding indices are as follows:

$$\begin{aligned} i, m &= \text{index for potential facility locations.} & i, m &= 1, \dots, |\Delta|. \\ l &= \text{index for facility.} & l &= 1, \dots, n. \\ C_{lim} &= \text{cost of relocating facility } l \text{ from location } i \text{ to location } m. (C_{lil} = 0) \end{aligned}$$

The decision variables of the problem are:

$$\begin{aligned} x_{lit} &= \begin{cases} 1 & \text{if facility } l \text{ is located at location } i \text{ at time } t \\ 0 & \text{otherwise.} \end{cases} \\ y_{jkt} &= \begin{cases} 1 & \text{if demand node } k \text{ is covered by at least } j \text{ facilities at time } t \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

$w_{limt}$  is a continuous variable but it assumes only integer values (refer to Section 2.3.1) and is defined as follows:

$$w_{limt} = \begin{cases} 2 & \text{if facility } l \text{ is relocated from location } i \text{ to location } m \text{ at time } t \\ 1 & \text{otherwise.} \end{cases}$$

The Dynamic MEXCLP with location dependent relocation cost is formulated as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} - \sum_{t=1}^T \sum_{l=1}^n \sum_{i \in \Delta} \sum_{m \in \Delta} C_{lim} (w_{limt} - 1) \\ \text{subject to} \quad & \end{aligned}$$

$$\sum_{j=1}^n y_{jkt} - \sum_{i \in \Delta} \sum_{l=1}^n r_{ikt} x_{lit} \leq 0 \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T, \quad (2.24)$$

$$\sum_{i \in \Delta} \sum_{l=1}^n x_{lit} \leq n \quad \forall \quad t = 0, \dots, T, \quad (2.25)$$

$$\sum_{i \in \Delta} x_{lit} = 1 \quad \forall \quad l = 1, \dots, n, t = 0, \dots, T, \quad (2.26)$$

$$\sum_{l=1}^n x_{lit} \leq 1 \quad \forall \quad i = 1, \dots, |\Delta|, t = 0, \dots, T, \quad (2.27)$$

$$w_{limt} \geq x_{lit-1} + x_{lmt} \quad \forall \quad l, i, m, t = 1, \dots, T, \quad (2.28)$$

$$x_{lit} \in \{0, 1\} \quad \forall \quad l, i, t = 0, \dots, T, \quad (2.29)$$

$$w_{limt} \geq 1 \quad \forall \quad l, i, m, t = 1, \dots, T, \quad (2.30)$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \dots, n, k = 1, \dots, |\Theta|, t = 0, \dots, T \quad (2.31)$$

The objective function maximizes the expected demand covered, allowing relocation of facilities with time, where the relocation cost depends on the pair of locations between which a facility moves. If node  $k$  is covered by  $m$  facilities at time  $t$ , constraint (2.24) assigns each of the variables  $y_{1kt}, y_{2kt}, \dots, y_{mkt}$  a value of 1 since the objective function is a maximization function containing the term  $y_{jkt}$ . Constraint (2.25) restricts the maximum number of facilities to be located to  $n$  for any time  $t$ . Constraint (2.26) restricts a facility to be located at only one location at any given period  $t$ . Constraint (2.27) restricts the maximum number of facilities to be located at any particular location to 1. If facility  $l$  is located at location  $i$  at time  $t - 1$ ,  $x_{lit-1} = 1$ . Similarly if facility  $l$  is located at location  $m$  at time  $t$ ,  $x_{lmt} = 1$ . Based on the above, if facility  $l$  is relocated at time  $t$  from location  $i$  to location  $m$ , constraint (2.28) ensures that  $w_{limt} \geq 2$ . The location dependent relocation cost is incurred only in this case, as is evident from the second summation term in the objective function of this model. Constraint (2.29) is a binary constraint.

### 2.3.4 Dynamic MEXCLP for Clustering with Mobile Facilities

In this variant of the Dynamic MEXCLP, we relax the assumption that the potential location of clusterheads (facilities) are different from the location of the sensors. In this model, instead of locating facilities at potential locations, we elect sensors which act as the clusterhead. Since the sensors are mobile, the elected clusterhead is itself a sensor. Thus clusterheads are also mobile and follow the same trajectory as they

followed when they were sensors.

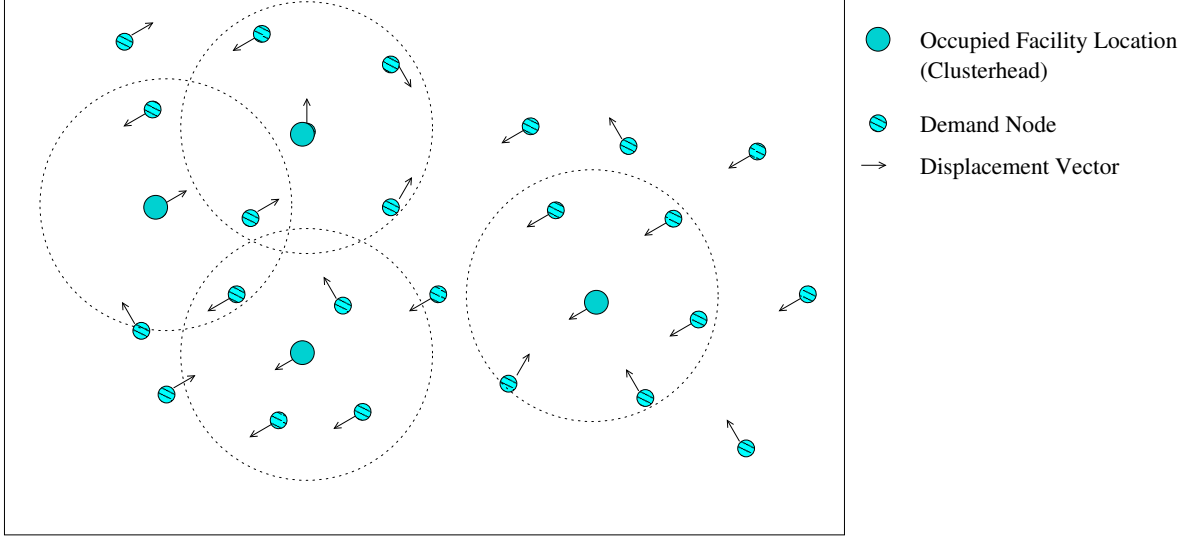


Figure 2.3: A snapshot of the network

We now redefine some of the parameters as follows:

$\Theta$  = set of potential facility locations.

$\Theta$  = set of demand nodes.

$D_{ikt}$  = distance between demand node  $i$  and demand node  $k$  at time  $t$ .

$C$  = cost per change in the status from CH to sensor or viceversa

We also redefine the decision variables of the problem as follows:

$x_{it} = \begin{cases} 1, & \text{if demand node } i \text{ acts as a clusterhead at time } t \\ 0 & \text{otherwise.} \end{cases}$

$y_{jkt} = \begin{cases} 1, & \text{if demand node } k \text{ is covered by atleast } j \text{ facilities (CHs) at time } t \\ 0 & \text{otherwise.} \end{cases}$

$w_{it} = \begin{cases} 1, & \text{if a demand node } i \text{ changes its status from clusterhead to sensor or viceversa.} \\ 0 & \text{otherwise.} \end{cases}$

The dynamic MEXCLP for clustering with mobile facilities is formulated as follows:

$$\text{Maximize} \quad \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} - \sum_{t=1}^T \sum_{i \in \Theta} C w_{it}$$

subject to

$$\sum_{j=1}^n y_{jkt} - \sum_{i \in \Theta} r_{ikt} x_{it} \leq 0 \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T, \quad (2.32)$$

$$\sum_{i \in \Theta} x_{it} \leq n \quad \forall \quad t = 0, \dots, T, \quad (2.33)$$

$$w_{it} \geq x_{it-1} - x_{it} \quad \forall \quad i = 1, \dots, |\Theta|, t = 1, \dots, T, \quad (2.34)$$

$$w_{it} \geq x_{it} - x_{it-1} \quad \forall \quad i = 1, \dots, |\Theta|, t = 1, \dots, T, \quad (2.35)$$

$$x_{it} \in \{0, 1\} \quad \forall \quad i = 1, \dots, |\Theta|, t = 0, \dots, T, \quad (2.36)$$

$$w_{it} \geq 0 \quad \forall \quad i = 1, \dots, |\Theta|, t = 1, \dots, T, \quad (2.37)$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \dots, n, k = 1, \dots, |\Theta|, t = 0, \dots, T, \quad (2.38)$$

The formulation is similar to the one described in Section 2.3.1 with the only difference that the set of potential facility location is same as that of the set of sensors. Also the objective is to elect clusterheads among the sensors such that the expected demand of the sensors covered by each clusterhead is maximized. This formulation can be used as a clustering tool in ad hoc networks ensuring network reliability. Clustering is done for efficient management of the network and also to ease location management and routing.

## 2.4 Chapter Summary

In this chapter, we presented a detailed description of our problem scenario. We studied the MEXCLP and its formulation by Daskin [29] in details. Then we model the problem at hand as a variant of MEXCLP termed as Dynamic MEXCLP. This is followed by an indepth study of the various variants of the Dynamic MEXCLP, along with their respective problem formulations. We also establish that the structure of the formulation remains unaltered if link failure is considered instead of facility failure. In chapter 3, we present a solution methodology developed for our problem.

## Chapter 3

# Solution Methodology: A Heuristic Approach

In the previous chapter, several variants of the dynamic MEXCLP were presented. The given models can be solved using a standard Mixed Integer Linear Programming (MILP) solver like CPLEX. But with increasing problem size, it becomes more and more difficult to solve the MILP using CPLEX. Large problems cannot be even read in to the CPLEX environment and hence solving such problems is not possible in today's context. We know from the literature that covering location problems including MEXCLP are  $NP - hard$  problems. Since Dynamic MEXCLP is a variant (or a time extension) of the traditional MEXCLP, it is also an  $NP - hard$  problem. We know that  $NP - hard$  problems are generally computationally intensive and tough to solve. Thus we need an alternative approach to solve the Dynamic MEXCLP efficiently, in order to determine the optimal facility locations. Since we are considering a military scenario, developing a solution methodology with emphasis on solution time is of crucial importance. However in a peacetime scenario, the quality of the solution (objective function value) is more important compared to the solution time. Hence, depending on the scenario, it would be necessary to consider a tradeoff between solution time and solution quality. And it would be a desirable property of the same solution method to operate under either situation.

In this chapter we develop two greedy heuristics for solving the Dynamic MEXCLP and analyze their performances. They are:



1. Relocation Heuristic (RH), and
2. No-Relocation Heuristic (NRH).

### 3.1 Relocation Heuristic

Relocation Heuristic (RH) is a greedy heuristic, which picks the best  $n$  locations, one at a time, for each time period. The heuristic takes into consideration the link failure probabilities as well as the relocation cost incurred in switching facilities locations in two successive time periods. The detailed procedure of the RH is presented as follows:

1. Calculate the demand covered by each potential location.
2. Select the potential location with maximum coverage and place a facility at this location. This potential location is now precluded from future placement of a facility for this particular time period.
3. Since there exists a probability of failure of the link ( $p$ ) between the facility placed and the demand covered, a fraction of the demand is actually covered. Thus we update the demand of the demand node covered by this facility by multiplying it by  $p$ . We use the updated demand for future calculation of coverage for only this time slot.
4. Repeat the above three steps until all the facilities are placed.
5. Repeat this procedure for all time slots. In order to take care of the relocation cost, we add  $2C$  (since a change in number of facility at two locations constitutes a relocation of a facility) to the total coverage of the potential location for time  $t$ , if that location was selected in the previous time period  $t - 1$ .

The heuristic pretty much captures the link failure probabilities by updating the demand of the sensor nodes after picking each location for placing a facility. The heuristic also gives weightage equivalent to relocation cost to locations selected in previous time periods in order to avoid too many relocations. RH is found to perform

considerably well for most problem instances, as is evident from the results discussed in detail in chapter 5.

One way of investigating how good the heuristic performs as far as solution quality is concerned is to find how poorly the heuristic can perform. Thus we carry out a worst case analysis for the heuristic. In order to evaluate the heuristic, we analyze its worst case error bound for which we require the concept of an  $\epsilon$ -approximate algorithm. This is formally defined as follows in Papadimitriou and Steiglitz [50]:

*Let  $A$  be an optimization (minimization or maximization) problem with positive integral cost function  $c$ , and let  $B$  be an algorithm which, given an instance  $I$  of  $A$ , returns a feasible solution  $f_B(I)$ ; denote the optimal solution of  $I$  by  $\hat{f}(I)$ . Then  $B$  is called an  $\epsilon$ -approximate algorithm for  $A$  for some  $\epsilon \geq 0$  if and only if*

$$\frac{|c(f_B(I)) - c(\hat{f}(I))|}{c(\hat{f}(I))} \leq \epsilon$$

*for all instances  $I$ .*

Let us consider the following example:

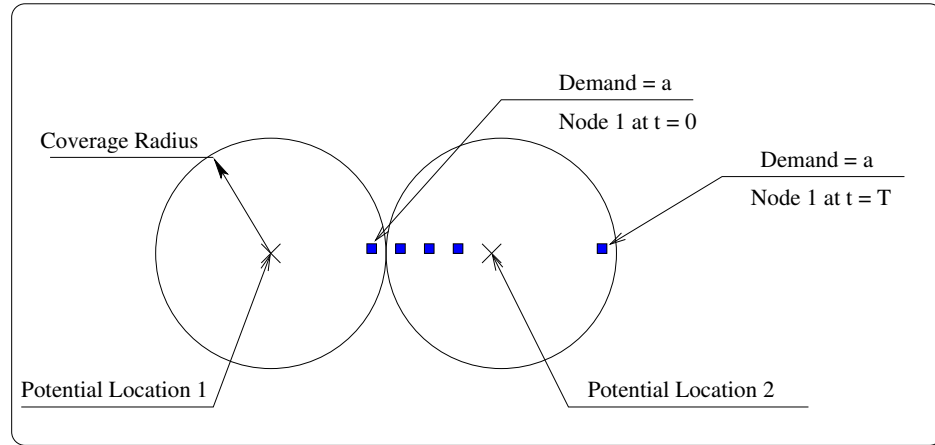


Figure 3.1: An example illustrating the worst case behaviour of RH

In the above example we consider:

Number of potential facility location = 2

Number of facilities to be placed = 1

Number of demand nodes = 1

Demand of node 1 =  $a$

Number of time slots =  $T + 1$

Probability of link failure =  $p$

Relocation cost =  $b$  ( $b > a$ )

The heuristic starts with finding the expected coverage for all the potential facility locations at time  $t = 0$  and locates a facility at the location covering maximum expected demand for that time period. In the above example, potential location 1 is selected for locating at time  $t = 0$  with expected coverage of  $(1 - p) * (a)$  for that time period. The heuristic again computes the expected demand covered for all the potential facility locations for the next time period and then considers the tradeoff between switching to another potential location. In the example for time period  $t = 1$ , potential location 2 has the maximum expected coverage among all potential locations. Thus RH computes the gain in coverage by switching to potential location 2 and its associated cost. Since the cost of relocating ( $b$ ) exceeds the gain in coverage ( $a$ ), the heuristic decides not to relocate and thus for time period  $t = 1$ , the facility is located at location 1. For all subsequent time periods, the heuristic chooses the facility to be located at location 1, thereby attaining a coverage of  $(1 - p) * (a * 1)$ . By simple observation we can say that the optimal solution is to locate the facility at potential location 2 for all time periods with total coverage of  $(1 - p) * (a * T)$ .

Thus we have

$$c(\hat{f}(I)) = (1 - p) * (a * T) \text{ and}$$

$$c(f_B(I)) = (1 - p) * (a * 1)$$

Hence

$$\begin{aligned} \frac{|c(f_B(I)) - c(\hat{f}(I))|}{c(\hat{f}(I))} &\leq \epsilon \\ \text{or, } \frac{|(1-p)*(a*1) - (1-p)*(a*T)|}{(1-p)*(a*T)} &\leq \epsilon \\ \text{or, } \frac{T-1}{T} &\leq \epsilon \end{aligned}$$

From the above equation, it is clear that the algorithm's relative error will violate all constant bounds ( $\epsilon$ ) for appropriate instances. Thus we conclude that the heuristic

is not an  $\epsilon$ -approximate algorithm for any  $1 > \epsilon > 0$ . In other words the algorithm performs arbitrarily bad with increasing number of time slots for a high relocation cost.

## 3.2 No Relocation Heuristic

No-Relocation Heuristic (NRH) is also a greedy heuristic where the optimal strategy will be to place facilities at locations which cover maximum demand for all time periods. The heuristic also takes care of link failure probabilities in a similar fashion as RH does. A detailed procedure of the NRH is as follows:

1. Calculate the total demand covered by each potential location for all time periods.
2. Select the potential location with maximum coverage and place the facility at this location for all the time periods. This potential location is precluded for future placement.
3. Update the demand of the demand nodes as discussed in the RH earlier.
4. Repeat the above steps until all the facilities are placed.

NRH is also found to perform considerably well for problem instances with high relocation costs, as described in Chapter 5. Now we will analyze the worst case error bound for this heuristic as we did in the earlier section. Consider an example illustrated in Fig. 3.2 with the following parameters:

Number of potential facility location =  $T + 2$

Number of facilities to be placed = 1

Number of demand nodes = 2

Demand of node 1 =  $a$  ( $a \gg b$ )

Demand of node 2 =  $b$

Number of time slots =  $T + 1$

Probability of link failure =  $p$

Relocation cost ( $C$ ) = 0

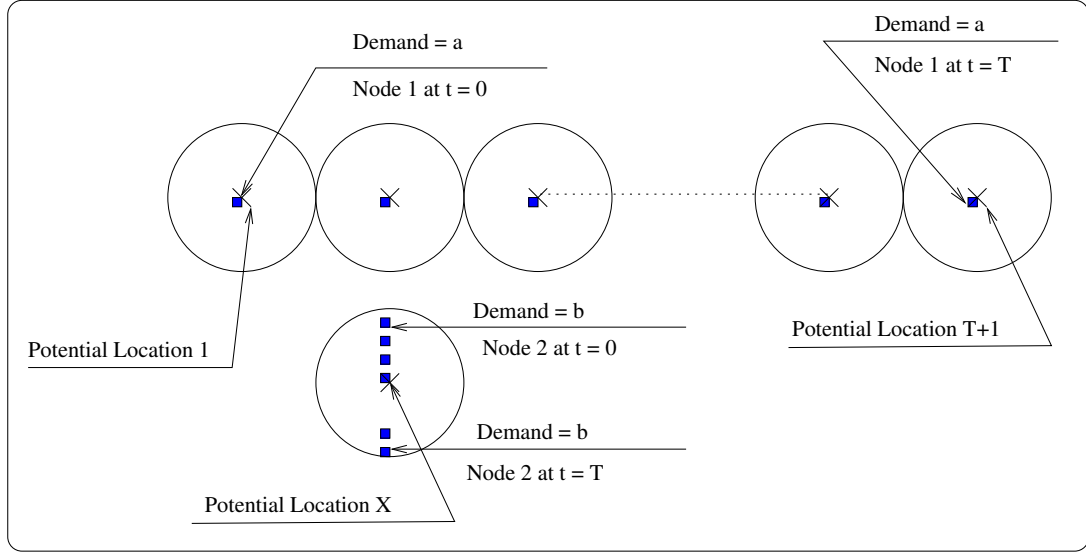


Figure 3.2: An example illustrating the worst case behaviour of NRH

NRH calculates the sum of the expected coverage for each potential location for all time periods, and selects a location which has the maximum sum for locating a facility at this location for all time periods. Potential location  $X$  is selected by the heuristic for locating a facility for all time periods with the coverage value of  $(1 - p) * b * (T + 1)$ . In this example we can see that demand node 1 moves with time from the coverage region of one potential location to another. In order to maximize expected coverage, the facility should be relocated to the potential location covering demand node 1 with each time period. Thus given a very low relocation cost, the optimal solution to the scenario is locating a facility at potential location 1 at  $t = 0$  and thereafter relocating the facility to the potential location covering demand node 1 in the subsequent time periods. The expected coverage attained by the solution is  $(1 - p) * a * (T + 1)$  (since  $C = 0$ .) Thus the worst case error bound of NRH, given that  $c(\hat{f}(I)) = (1 - p) * a * (T + 1)$  and

$c(f_B(I)) = (1 - p) * b * (T + 1)$  is

$$\begin{aligned} \frac{|c(f_B(I)) - c(\hat{f}(I))|}{c(\hat{f}(I))} &\leq \epsilon \\ \text{or, } \frac{|(1-p)*b*(T+1) - (1-p)*a*(T+1)|}{(1-p)*a*(T+1)} &\leq \epsilon \\ \text{or, } \frac{a-b}{a} &\leq \epsilon \\ \text{or, } 1 - \frac{b}{a} &\leq \epsilon \end{aligned}$$

Note that  $b * (T + 1) > a$ , since potential location  $X$  is chosen as the solution by the heuristic. Thus in the worst case, the heuristic can perform arbitrarily bad with the value of  $a \gg b$ . Hence we conclude that the relative error bound of the algorithm will violate all constant bounds ( $\epsilon$ ) for an appropriate instance as illustrated in the above example.

Thus we conclude that the heuristic is not an  $\epsilon$ -approximate algorithm for any  $1 > \epsilon > 0$ .

### 3.3 Dynamic Programming Approach

A dynamic programming (DP) approach has also been explored in our work to solve the Dynamic MEXCLP. The motivation for exploring the DP approach was obtained from the literature studied in the area of Dynamic Plant Layout Problem. Refer to the papers by Rosenblatt [57], Batta [9], Balakrishnan et al. [3].

In our case the number of time slots forms the number of stages and all the feasible solutions of each time period form the states of the dynamic programming model. Since the number of feasible solutions to each time period is combinatorial, a stopping criteria, i.e., a bounding technique is required to include the best few feasible solutions in a dynamic programming approach ensuring that optimality is attained.

Let  $Z_m^t$  be the expected demand covered with solution  $S_m$  in time period  $t$ .  $Z_m^t$  is the  $m^{th}$  best solution obtained in time period  $t$ . Here we intend to find the  $l_t$  best solutions for each time period  $t$  such that no optimal solution is lost from the state space diagram as shown in Fig. 3.3 of the dynamic programming model. The

number of best solutions for each time period that should be included in the state-space diagram can be restricted using the following results:

**Result 1** [57]:- If  $l_t$  is given by  $Z_1^t - Z_{l_t}^t \leq Z_{UB} - Z_{LB}$  and  $Z_1^t - Z_{l_t+1}^t > Z_{UB} - Z_{LB}$ , then, in any period  $t$ , no solution with value  $l > l_t$  will become part of the optimal solution.

One of the techniques to find a good upper bound is as follows:

1. Summing up the coverage obtained by the best (optimal) solution of each time period without considering the relocation cost [57].

Some of the techniques to find a good lower bound are as follows:

1. The coverage obtained by selecting the best solution in each time period and subtracting from it the relocation cost incurred due to change in the solution from one time period to the next [57].
2. Keeping the same solution throughout the time horizon [9]. This could be done by considering the locations of all demand nodes and their demands for different time periods as a distinct demand node in a single time period and solving the MEXCLP model to get a lower bound for the Dynamic Programming model.

**Result 2** [3]: If  $l_t$  is given by  $Z_1^t \geq Z_1^t - 4Cn$  and  $Z_{l_t+1}^t < Z_1^t - 4Cn$  where

$C$  = Cost per unit change in the number of facilities at any location  $i$ ,

$n$  = Number of facilities to be located,

then, any stage in the dynamic programming model will have no more than  $l_t$  states for time period  $t$ . In other words the optimal solution will not consist of states with an expected demand coverage less than  $Z_1^t - 4Cn$ .

This strategy works best for a case where the relocation cost is relatively low.

The best solution  $S_1$  with a value of  $Z_1^t$  for each time  $t$  can be obtained by solving the MEXCLP model proposed by Daskin [29]. In order to get the next best solution, the following additional constraint is added to the MEXCLP model:

$$\sum_{i \in S_m} x_i \leq n - 1$$

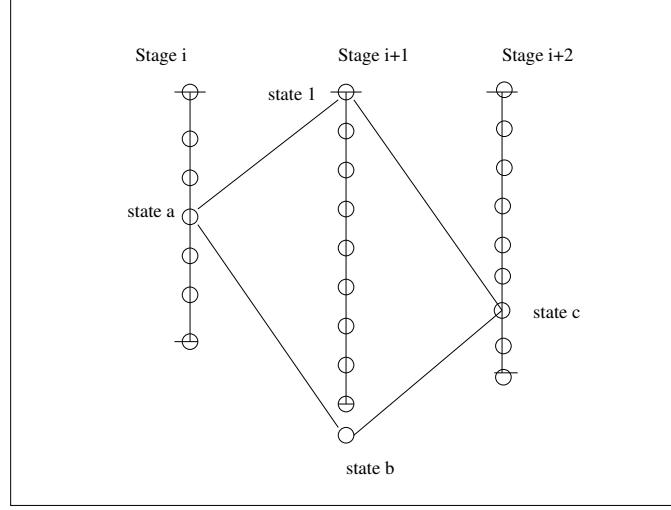


Figure 3.3: State space diagram: Dynamic Programming

where  $S_m = \{i : x_i > 0\}$

$S_m$  is the set of  $x_i$  which take a nonzero value in the  $m^{th}$  best solution for time  $t$ .

Thus we get  $Z_2^t$  by adding the previous constraint to the solution set of  $Z_1^t$ . Similarly by adding the previous constraint to the solution set of  $Z_2^t$ ,  $Z_3^t$  is obtained and so on. Once all the states for all stages are found, the relocation cost incurred between every states in two consecutive time periods are calculated and then the longest path algorithm is used to determine the optimal facility locations.

We found a large number of feasible solutions (states) for each stage with the bounds discussed previously. Since we are solving a MILP several times to get all the states of the state space diagram, the approach becomes computationally intensive as compared to solving the Dynamic MEXCLP using CPLEX. Hence we did not explore the approach any further.

### 3.4 Chapter Summary

In this chapter, we developed two greedy heuristics to solve the Dynamic MEXCLP model presented in Section 2.3.1. Also we analyzed the worst case error bounds for each heuristic and showed that both the heuristics can perform arbitrarily bad. A



dynamic programming based approach is also discussed to solve the Dynamic MEX-CLP.

# Chapter 4

## Solution Methodology: A Column Generation Approach

In the previous chapter two heuristics were proposed to solve the Dynamic MEXCLP and also their worst case error bounds were presented. Since the worst case behavior of both the heuristics is arbitrarily bad, an alternative approach is needed to solve the Dynamic MEXCLP in conjunction with Relocation Heuristic (RH) and No-Relocation Heuristic (NRH). Thus the idea here is to get the initial solution from the heuristic and then improve the solution using the column generation (CG) technique in cases where the heuristic performs bad.

### 4.1 Introduction

Column generation (CG) heuristic is an efficient and widely used technique to solve large-scale integer programs. We adopt a modified column generation approach to solve the Dynamic MEXCLP. The column generation formulation is obtained by decomposing the original MILP (Dynamic MEXCLP with relocation cost) described in Section 2.3.1. If the variables  $x_{it}$ 's are known then the remaining problem is to just evaluate the values of  $y_{jkt}$ 's and  $w_{it}$ 's to determine the expected demand covered and the relocation cost incurred. Thus we decompose the MILP formulation into a “Master Problem” and a “Sub-Problem”.

The Master Problem selects the best column (or solution) for each time period to

maximize the objective function. Where a column (or a solution) is defined as a complete set of  $x_{it}$  variable values for a particular time period. Whereas the sub-problem uses the dual multipliers from the master problem and generates a new column with a favourable reduced cost for each time period  $t$  separately.

We now define some additional parameters and variables used in the master problem

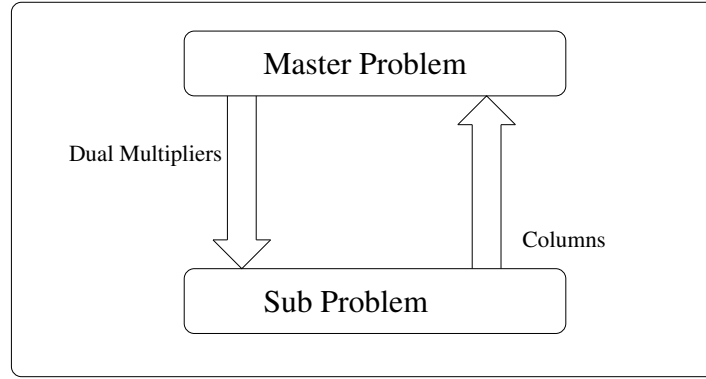


Figure 4.1: Column Generation Flow

and sub-problem for the column generation approach as follows:

$F_t$  = set of feasible solutions for time  $t$ .

$x_{sit}$  = value of  $x_{it}$  if solution  $s$  is selected at time  $t$ .

$y_{sjkt}$  = value of  $y_{jkt}$  if solution  $s$  is selected at time  $t$ .

The decision variables of the problem are:-

$$F_{st} = \begin{cases} 1 & \text{if solution } s \text{ is selected at time } t \\ 0 & \text{otherwise.} \end{cases}$$

## 4.2 Initial Basic Feasible Solution

The Column Generation(CG) approach works in the feasible domain. The CG heuristic requires an initial basic feasible solution to start with. It starts with this solution and keeps on improving the objective function value by generating new solutions termed as “columns” and selecting the one that improves the objective function the most. The effectiveness of the CG approach is enhanced by the quality of the initial

basic feasible solution. Hence it is important to develop a good heuristic for obtaining an initial basic feasible solution. In our work, we start with two basic feasible solutions for each time period generated by the RH and NRH discussed in previous chapter.

### 4.3 Column Generation Formulation-I (CG-I)

Here we decompose the Dynamic MEXCLP into master and sub-problem, such that the sub-problem fixes the value of  $x_{it}$ 's and introduces as a column in the master problem. The master problem evaluates the set of solutions and selects the one which gives the maximum objective value.

#### 4.3.1 Master Problem

The decomposed master problem for the CG-I approach is as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} - \sum_{t=1}^T \sum_{i \in \Delta} C w_{it} \\ \text{subject to} \quad & \end{aligned}$$

$$\sum_{j=1}^n y_{jkt} - \sum_{s \in F_t} \sum_{i \in \Delta} F_{st} r_{ikt} x_{sit} \leq 0 \quad \forall \quad k = 1, \dots, |\Theta|, t = 0, \dots, T \quad (4.1)$$

$$-w_{it} + \sum_{s \in F_{t-1}} F_{st-1} x_{sit-1} - \sum_{s \in F_t} F_{st} x_{sit} \leq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T \quad (4.2)$$

$$-w_{it} + \sum_{s \in F_t} F_{st} x_{sit} - \sum_{s \in F_{t-1}} F_{st-1} x_{sit-1} \leq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T \quad (4.3)$$

$$\sum_{s \in F_t} F_{st} = 1 \quad \forall \quad t = 0, \dots, T \quad (4.4)$$

$$F_{st} \in \{0, 1\} \quad \forall \quad s, t = 0, \dots, T \quad (4.5)$$

$$w_{it} \geq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T \quad (4.6)$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \dots, n, k = 1, \dots, |\Theta|, t = 0, \dots, T \quad (4.7)$$

In the column generation approach, the master problem is solved as a relaxed linear program with no binary constraint on the variables. The master problem starts with an initial basic feasible solution. The basic function of the master problem is to select the best feasible solution for each time period among the set of feasible

solutions available for each time period  $t$ . Constraints (4.1), (4.2) and (4.3) are similar to the basic model except that we have variable  $F_{st}$  instead of  $x_{it}$ , whereas  $x_{sit}$  is a constant. Constraint (4.4) ensures that only one solution should be selected for each time period  $t$ . Let  $\alpha, \beta, \gamma, \delta$  be the dual multipliers generated after solving the relaxed master problem for constraint (4.1), constraint (4.2), constraint (4.3) and constraint (4.4) respectively. The sub-problem generates the feasible solutions using these dual multipliers.

### 4.3.2 Sub-Problem

The sub-problem is formulated in such a way that the new column generated has a favorable reduced cost to enter the basis of the master problem. The sub-problem is solved for each time period  $t$  separately. The sub-problem for time periods  $t = 1$  to  $t = T - 1$  is as follows:

$$\begin{aligned} &\text{Maximize} && \sum_{i \in \Delta} [\sum_{k \in \Theta} \alpha_{kt} r_{ikt} + \beta_{it} - \beta_{it+1} - \gamma_{it} + \gamma_{it+1}] x_{it} - \delta_t \\ &\text{subject to} \end{aligned}$$

$$\sum_{i \in \Delta} x_{it} \leq n \tag{4.8}$$

$$x_{it} \in \{0, 1\} \quad \forall \quad i \tag{4.9}$$

Since the dual multipliers  $\beta$  and  $\gamma$  are available for  $t = 1$  to  $t = T$ , the sub-problem for  $t = 0$  and  $t = T$  has a different objective function. The objective function of the sub-problem for  $t = 0$  is as follows:

$$\sum_{i \in \Delta} [\sum_{k \in \Theta} \alpha_{kt} r_{ikt} - \beta_{it+1} + \gamma_{it+1}] x_{it} - \delta_t$$

and the objective function of the sub-problem for  $t = T$  is as follows:

$$\sum_{i \in \Delta} [\sum_{k \in \Theta} \alpha_{kt} r_{ikt} + \beta_{it} - \gamma_{it}] x_{it} - \delta_t$$

The solution to the above sub-problem is a new variable  $F_{st}$  which enters the master problem. Thus the sub-problem generates the new variable in such a way that it will enter the basis of the master problem.

The sub-problem is very simple to solve, since it is a special case of single constraint Knapsack problem where the coefficient of each variable in the constraint is same.

Thus it turns out that if we select the  $n$  best locations (in terms of maximum objective coefficient) and locate a facility at each of these locations, we get the solution to the integer program. Thus we need not solve the sub-problem using CPLEX.

It is clearly evident that the computational load is not evenly distributed between the master and the sub-problem. Thus we also propose an alternate way of decomposing the Dynamic MEXCLP in the following section.

## 4.4 Column Generation Formulation - II (CG-II)

In this formulation the sub-problem not only fixes  $x_{it}$ 's, but also evaluates the value of  $y_{jkt}$ 's thereby reducing the work of the master problem. Thus the master problem only evaluates the relocation cost incurred in selecting the solution and selects the solutions for each time period that gives the maximum objective function value.

### 4.4.1 Master Problem

The decomposed master problem for the CG-II approach is as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{t=0}^T \sum_{k \in \Theta} \sum_{j=1}^n \sum_{s \in F_t} (1-p)p^{j-1} d_k F_{st} y_{sjkt} - \sum_{t=1}^T \sum_{i \in \Delta} C w_{it} \\ \text{subject to} \quad & \end{aligned}$$

$$-w_{it} + \sum_{s \in F_{t-1}} F_{st-1} x_{sit-1} - \sum_{s \in F_t} F_{st} x_{sit} \leq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T \quad (4.10)$$

$$-w_{it} + \sum_{s \in F_t} F_{st} x_{sit} - \sum_{s \in F_{t-1}} F_{st-1} x_{sit-1} \leq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T \quad (4.11)$$

$$\sum_{s \in F_t} F_{st} = 1 \quad \forall \quad t = 0, \dots, T \quad (4.12)$$

$$F_{st} \in \{0, 1\} \quad \forall \quad s, t = 0, \dots, T \quad (4.13)$$

$$w_{it} \geq 0 \quad \forall \quad i = 1, \dots, |\Delta|, t = 1, \dots, T \quad (4.14)$$

Also in this column generation approach, the master problem is solved as a relaxed linear program with no binary constraint on the variables. The master problem starts with an initial basic feasible solution. The basic function of the master problem is to select the best feasible solution for each time period among the set of feasible solutions

available for each time period  $t$ . Constraints (4.10) and (4.11) are similar to the basic model except that we have variable  $F_{st}$  instead of  $x_{it}$  and  $y_{jkt}$ , whereas  $x_{sit}$  and  $y_{sjkt}$  are constants. Constraint (4.12) ensures that only one solution should be selected for each time period  $t$ . Let  $\beta, \gamma, \delta$  be the dual multipliers generated after solving the relaxed master problem for constraint (4.10), constraint (4.11) and constraint (4.12) respectively. The sub-problem generates the feasible solutions using these dual multipliers.

#### 4.4.2 Sub-Problem

The sub-problem is formulated in such a way that the new column generated has a favorable reduced cost to enter the basis of the master problem. The sub-problem is solved for each time period  $t$  separately. The sub-problem for time periods  $t = 1$  to  $t = T - 1$  is as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} + \sum_{i \in \Delta} [\beta_{it} - \beta_{it+1} - \gamma_{it} + \gamma_{it+1}] x_{it} - \delta_t \\ \text{subject to} \quad & \end{aligned}$$

$$\sum_{j=1}^n y_{jkt} - \sum_{i \in \Delta} r_{ikt} x_{it} \leq 0 \quad \forall \quad k = 1, \dots, |\Theta| \quad (4.15)$$

$$\sum_{i \in \Delta} x_{it} \leq n \quad (4.16)$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \dots, n, k = 1, \dots, |\Theta| \quad (4.17)$$

$$x_{it} \in \{0, 1\} \quad \forall \quad i = 1, \dots, |\Delta| \quad (4.18)$$

Since the dual multipliers  $\beta$  and  $\gamma$  are available for  $t = 1$  to  $t = T$ , the sub-problem for  $t = 0$  and  $t = T$  has a different objective function. The objective function of the sub-problem for  $t = 0$  is as follows:

$$\sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} + \sum_{i \in \Delta} [-\beta_{it+1} + \gamma_{it+1}] x_{it} - \delta_t$$

and the objective function of the sub-problem for  $t = T$  is as follows:

$$\sum_{k \in \Theta} \sum_{j=1}^n (1-p)p^{j-1} d_k y_{jkt} + \sum_{i \in \Delta} [\beta_{it} - \gamma_{it}] x_{it} - \delta_t$$

The solution to the above sub-problem is a new variable  $F_{st}$  which enters the master problem. Thus the sub-problem generates the new variable in such a way that it will

enter the basis of the master problem. Thus for each time period a small MILP is solved to get the new solution for that time period.

The nice thing about this formulation is that the computational load gets divided equally between the master problem and the sub-problem.

## 4.5 Strategies

In our work we adopt two different strategies to solve the problem using column generation technique.

### 4.5.1 Strategy I

In this strategy, we solve the LP relaxation of the master problem also called the relaxed master problem(RMP) and the subproblem alternatively (in a sequential way).

The algorithm is as follows:

Step 1:- Find the initial basic feasible solution and include it in the RMP. Set Iteration  $\# = 0$ .

Step 2:- Initialize time  $t = 0$ , and increment Iteration  $\#$  by 1.

Step 3:- Solve RMP.

Step 4:- Obtain dual multipliers for time  $t$  and pass to the subproblem.

Step 5:- Solve the subproblem for time  $t$  and calculate the reduced cost of the new variable (solution) generated.

Step 6:- Feed the new solution generated to the RMP and increment time  $t$ .

Step 7:- If  $t \leq T$  goto Step 3, otherwise goto Step 8.

Step 8:- If reduced cost of all the new solutions generated for all time periods  $t = 0$  to  $t = T$  is less than zero, goto Step 9, otherwise goto Step 2.

Step 9:- Solve the Integer Master Problem and stop.

In this strategy for CG-I, RMP solution converges to a value equal to the original problem's linear programming relaxation in a very few iterations, since we solve the



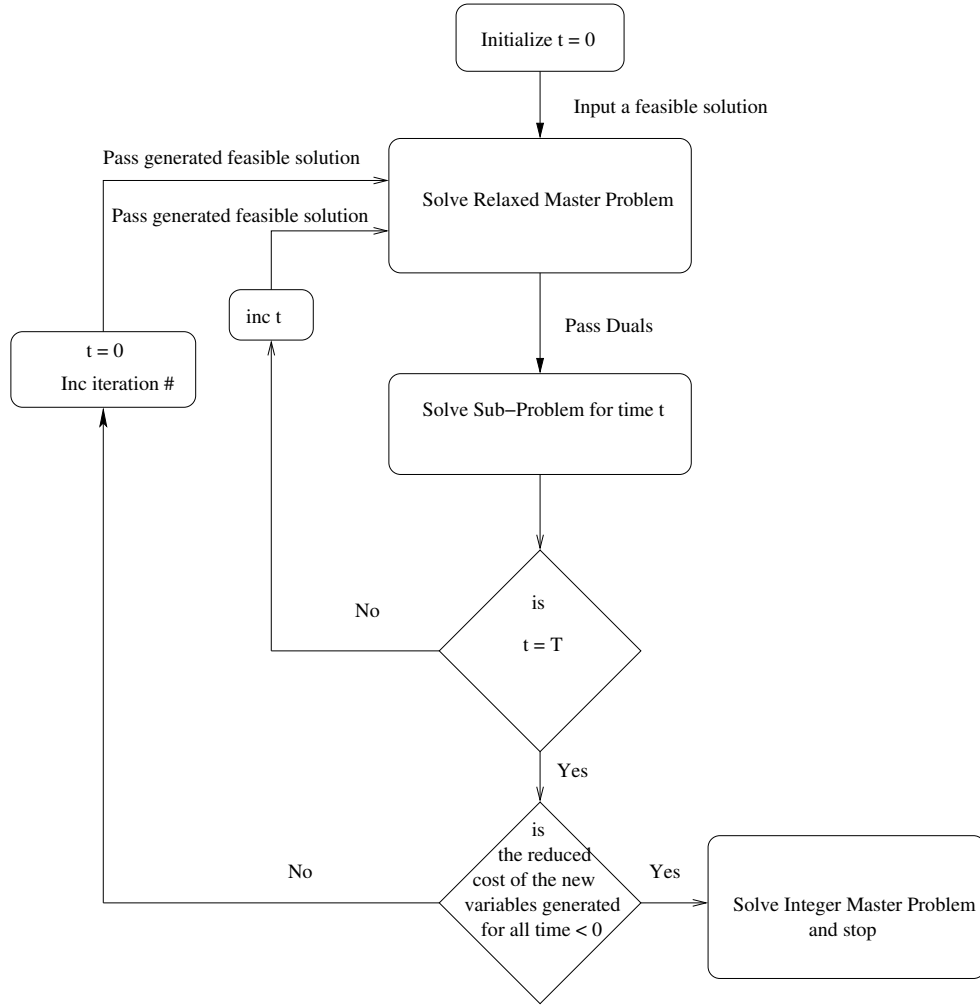


Figure 4.2: Strategy I

RMP and update the dual multipliers after solving the sub-problem for each time period. Due to the problem size, solving the RMP after solving the sub-problem for each time period is computationally intensive. Thus the solution time tends to be poor in this case. Whereas CG-II tends to work suitably with this strategy since the work load is evenly shared between master problem and the sub-problem.

### 4.5.2 Strategy II

In this strategy we solve the RMP and then solve the subproblem for time period  $t = 0$  to  $t = T$  consecutively one after another. The algorithm is as follows:

Step 1:- Find the initial Basic Feasible Solution and include it in the RMP. Set Iteration  $\# = 0$ .

Step 2:- Initialize time  $t = 0$  and increment Iteration  $\#$  by 1.

Step 3:- Solve RMP.

Step 4:- Obtain dual multipliers for time  $t$  and pass to the sub-problem.

Step 5:- Solve the sub-problem for time  $t$  and calculate the reduced cost of the new variable (solution) generated.

Step 6:- Increment  $t = t + 1$ ; if  $t \leq T$  goto Step 4 otherwise goto Step 7.

Step 7:- If the reduced cost of the new solutions generated for all time periods  $t = 0$  to  $t = T$  is less than zero, goto Step 9; otherwise goto Step 8.

Step 8:- Feed solutions generated for all time periods to RMP and goto Step 2.

Step 9:- Solve the Integer Master Problem and stop.

In this strategy the solution time is lesser compared to Strategy I, since we solve the RMP after solving the subproblem for all time periods  $t$ . Thus we achieve the solution in lesser time but the number of iterations is larger, since we do not have updated dual multipliers at every step.

The cycle of passing the dual multipliers to the subproblem and passing a feasible solution to the master problem continue until no more improvement in the objective function value of the relaxed master problem is possible. At the end of both the strategies, we solve the Integer Master Problem since we are interested in selecting only one best solution for each time period.

## 4.6 Solving Integer Master Problem

After performing several iterations of solving RMP, when the termination criteria is reached the integer master problem (IMP) is solved. The IMP finds the best feasible solution for each time period for the overall problem. We can model the IMP as a Longest Path Problem (LPP) with time periods as stage and solutions for each time

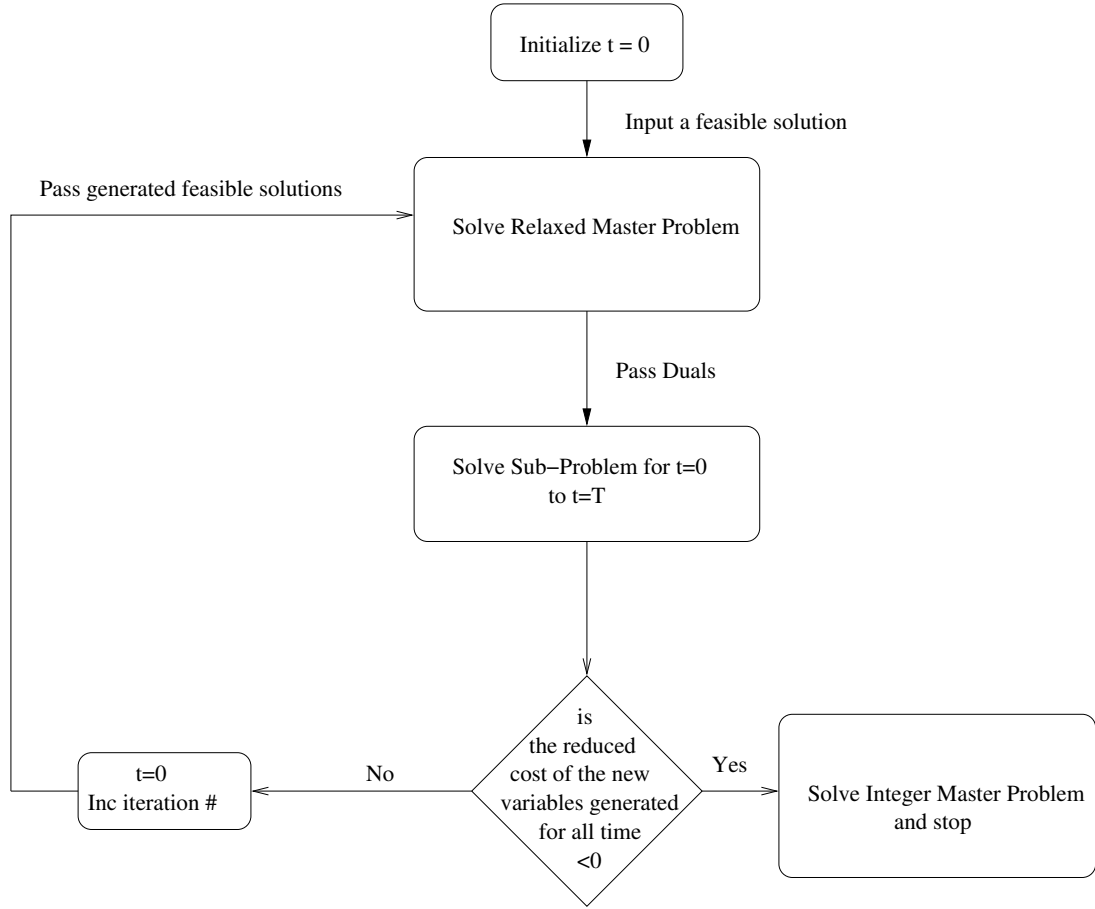


Figure 4.3: Strategy II

period as the states of the state space diagram. The arc lengths represent the gain in expected coverage if you travel through that arc. Thus the solution to the longest path algorithm consists of travelling through one solution for each time period. The solution time for solving the IMP drastically reduces by solving it by longest path algorithm.

## 4.7 Termination Criteria

In the previous section, the termination criterion for the RMP is the traditional CG termination criterion, where the reduced cost of the generated columns is less than zero (as evident from Fig. 4.2 and Fig. 4.3). This is true because a (nonbasic)

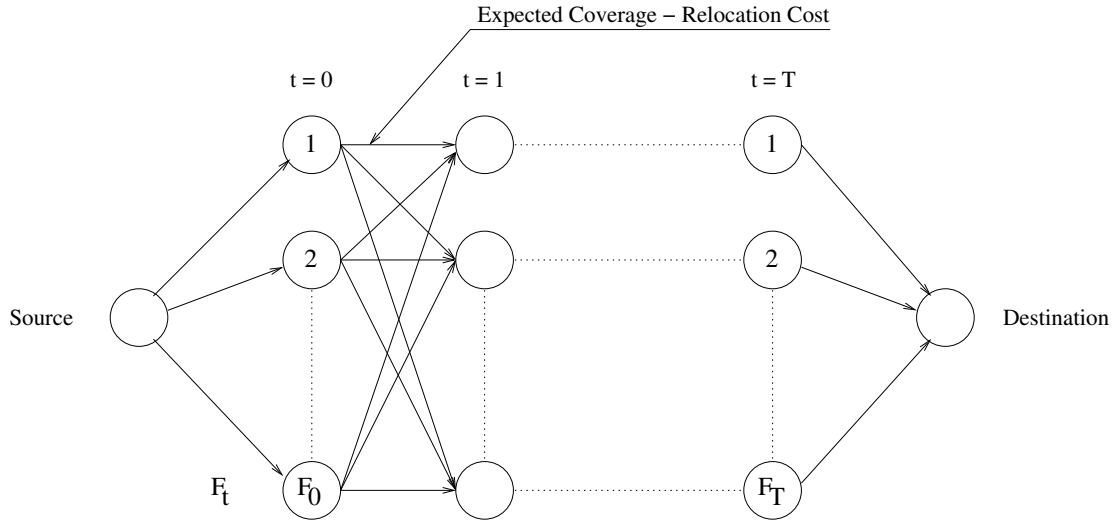


Figure 4.4: State Space Diagram: Integer Master Problem

variable with a negative reduced cost will never enter the basis of a RMP with a maximization objective function. In certain cases, the CG heuristic will keep on iterating with a very small increase in the objective function of the RMP. In such cases, we can terminate the CG heuristic when the desired solution quality has been obtained (i.e., the objective function value of the RMP is within a certain percentage of the linear programming relaxation of the original problem). This will improve the solution time of the CG approach. Another possible termination criterion can be a threshold time within which a solution is required. Bound on the number of iterations can serve as a good termination criteria.

## 4.8 Chapter Summary

In this chapter, we proposed a column generation based methodology to solve the Dynamic MEXCLP model presented in section 2.3.1. The Dynamic MEXCLP model is decomposed into a master problem and a sub-problem. Also two different strategies are proposed for solving the decomposed model using column generation technique.

## Chapter 5

# Computational Experiments and Detailed Numerical Analysis

In the previous chapters, we described the Dynamic MEXCLP in detail and presented several formulations for the same. In Chapter 3 and 4, solution methodologies are developed to solve the Dynamic MEXCLP. Two heuristics were presented in Chapter 3 and two alternate column generation approaches were discussed to solve the basic model (Dynamic MEXCLP with relocation) by decomposing it into a Master Problem and a Sub-Problem as shown in Chapter 4.

In this chapter, we present a detailed description of the code developed for the CG heuristic and also to carry out the numerical analysis. In Section 5.1, we describe the different parameters governing the problem size and complexity and the software implementation for solving the Dynamic MEXCLP. In Section 5.2, a detailed analysis of a scenario is presented, where we can observe the model behavior with varying parameter values and their effect on the computation time. In Section 5.3, analysis of variance is conducted to determine the effect of the parameters on the solution quality and the time to solve the problem. Also this forms the basis to statistically infer which solution methodology works better in terms of solution time and the value of objective function. Also we present computational time and solution for large size problems. In Section 5.4, a regret analysis is carried out for a particular problem instance when the value of  $p$  is uncertain. This ensures that a stable solution can be obtained for a range of  $p$  values. The computational experience is summarized in

## 5.1 Input Parameters and Software Implementation

### 5.1.1 Input Parameters

In this section we describe the parameters governing the problem structure.

1.  $n$ : Maximal number of facilities available for any given time period.  
The value of  $n$  changes the expected coverage attained. For a case where  $p = 0$ , the coverage increases with increasing  $n$  until a threshold value of  $n$  is reached. Further increase in  $n$  has no effect on the objective function value of the problem. For a case where  $p > 0$ , increase in  $n$  increases coverage unless all the potential locations covering at least a single demand node have been used for facility location (since the maximum number of facilities allowed at a location is restricted to 1).
2. **Displacement Range**: The maximum displacement possible in  $X$  and  $Y$  direction per unit time.
3. **Displacement**: This parameter specifies the displacement of the sensors in  $X$  and  $Y$  directions per unit time. It is randomly generated between 0 and displacement range.
4. **Potential Location Range**: This parameter defines the region within which the potential facility locations are randomly distributed. E.g., if the value of potential location range is 100,  $x$  and  $y$  coordinates of all the potential facility location are randomly distributed between 0 and 100. The  $z$  coordinate is fixed to 10.
5. **Sensor Location Range**: This parameter defines the region within which the sensors are located at time  $t = 0$ . The location of a particular sensor at

subsequent time slots is calculated based on the displacement of that sensor per unit time.

6. **Data Range:** This parameter specifies the interval between which the demand of the sensors lies.
7.  $|\Delta|$ : This parameter specifies the number of potential locations available for locating facilities, where  $\Delta$  is the set of all potential facility locations. We assume that the facilities are allowed to be located at only discrete locations.
8.  $|\Theta|$ : This parameter specifies the number of sensors, where  $\Theta$  is the set of all sensors.
9.  $U$ : This parameter defines the coverage radius within which a facility can cover a sensor.
10.  $d$ : Demand of each sensor that needs to be fused per time period (generally it is in megabytes). E.g. a radar scans a given region and generate its report and this is to be fused by the facility (clusterhead). Thus the amount of data sent at each periodic interval forms the demand in these models.
11.  $p$ : Specifies the probability of link failure between the facility and sensors.
12.  $C$ : The relocation cost incurred per unit change in the number of facilities placed at a location between two successive time periods.
13.  $T$ : The value of  $T$  signifies that the time horizon under consideration is 0 to  $T$ . Thus we have  $T + 1$  time slots for which the problem is to be optimized.

### 5.1.2 Software Implementation

The Column Generation heuristic is coded in C and CPLEX 7.1 solver is used for solving the Relaxed Master Problem at each iteration of the CG heuristic and the Integer Master Problem at the end of CG heuristic. The experiment was carried out

on a Intel Pentium 4 processor, 1700 MHz, 512 MB RAM operating on Windows platform.

The following is a detailed stepwise description of the software implementation:

1. **Input Parameters:** In this module of the program, we set the value of parameters that define the problem scenario, such as  $n$ ,  $p$ ,  $C$ ,  $|\Delta|$ ,  $|\Theta|$ ,  $U$ ,  $T$ , Potential Location Range, Sensor Location Range, Data Range and Displacement Range.
2. **Generate Data():** This module generates the problem randomly based on the value of the input parameters set in the previous module, such as coordinates of the potential facility locations (same for all time periods), sensor locations for time  $t = 0$ , sensor locations for the remaining time periods (calculated based on the displacement of each sensor per unit time). This module also calculates the reachability matrix which defines whether a potential facility location is capable of covering a demand node at time  $t$ . The demand of each sensor is also randomly generated by this module (between the data range specified earlier).
3. **Initial BFS():** In this module, basic feasible solutions are obtained for the problem scenario generated in the previous steps using two greedy heuristics. We start with two feasible solutions (columns) for the CG heuristic.
4. **Generate RMP():** This module writes the Relaxed Master Problem (RMP) formulation in a “.LP” format. This file is called RMP.lp.
5. **Solve the RMP with CPLEX:** In this module, we read the RMP.lp into the CPLEX environment and solve using CPXlpopt(). CPXlpopt() is an in built procedure of CPLEX callable library, which solves a linear programming formulation using the best technique applicable to the problem. In our case CPXlpopt() uses the Dual Simplex algorithm to solve the problem.
6. **Store Dual Multipliers:** Here the dual multipliers for the relevant constraints are stored using CPXgetpi() (CPLEX callable library function).



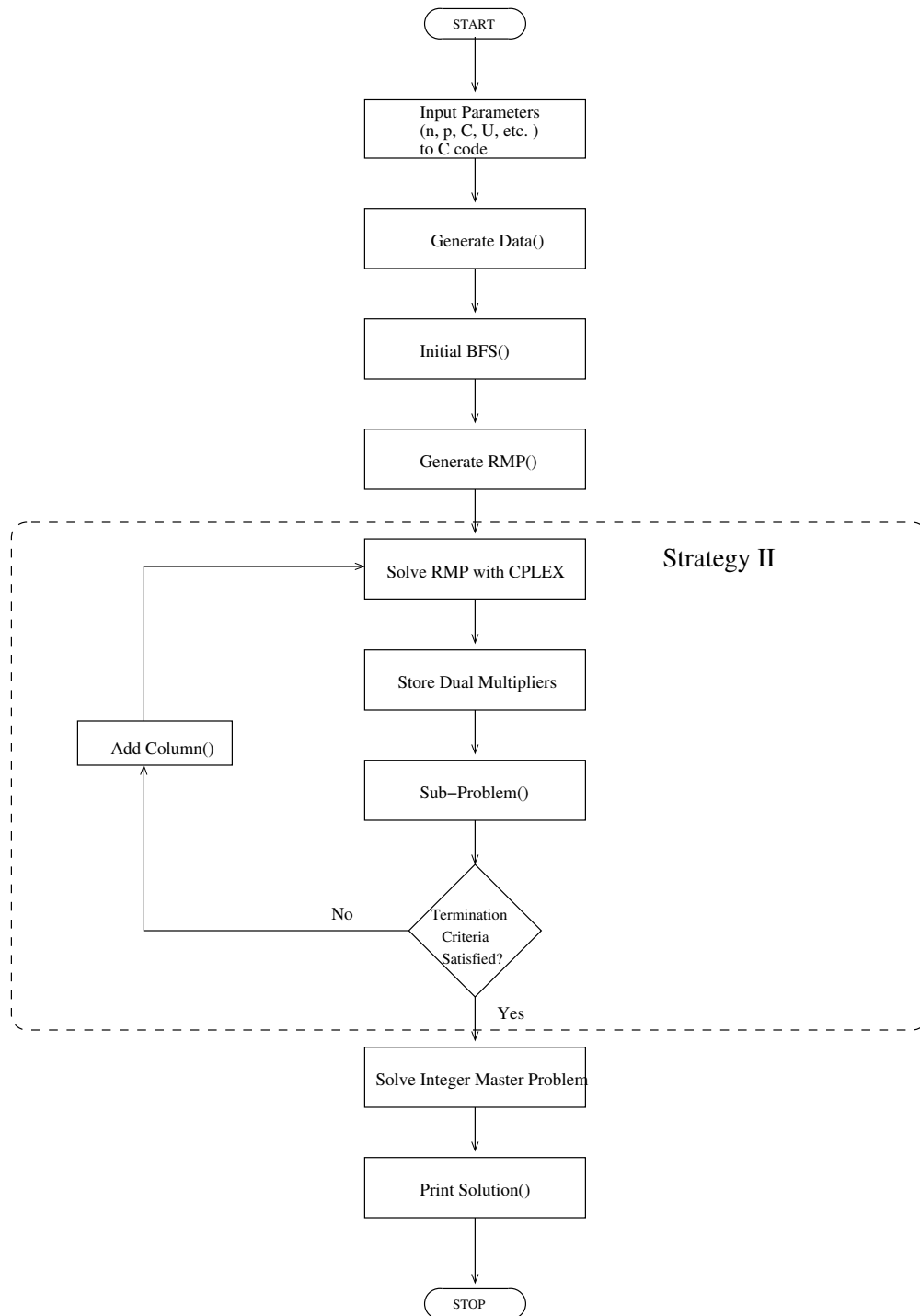


Figure 5.1: Software Implementation Flow Chart

7. Sub-Problem(): In this module, depending on the strategy selected to solve the CG, the sub-problem for different time periods are constructed based on the dual multipliers obtained from the RMP. For CG-I, the constructed sub-problem is a single constraint Knapsack problem with the coefficient of each variable in the constraint being same. Thus the sub-problem has an intuitive solution and hence we need not solve it using CPLEX. For CG-II the sub-problem for each time period is solved using CPLEX MIP solver.
8. Add Column(): In this module, the newly generated solution for each time period is added to the basis of the RMP.lp with the function CPXaddcols().
9. Solve Integer Master Problem: In this step a longest path algorithm is solved to get the optimal feasible solution for the CG procedure.

Finally we print the final solution obtained. It is important to note that some steps of the code are strategy specific. In this section, we describe the software implementation for a case where strategy II is selected.

## 5.2 Detailed Analysis of a scenario

In this section we consider a particular problem instance with the parameters listed in Table(5.1).

We study the effect of varying  $p$  and  $C$  on the objective function value and solution time.  $p$  varies between 0 and 1 in intervals of 0.1. Whereas  $C$  varies between 0 and 50 in intervals of 1. The basic aim is to study the model behavior by varying these parameters.

### 5.2.1 Effect of varying $p$ on the model behavior

#### Comparison of objective function value with varying $p$ value

It is obvious that with increasing probability of link failure ( $p$ ), the expected coverage decreases and becomes zero at  $p = 1$ , as depicted in Table(5.2).

Parameter	Range/Value
Potential Location Range	0-1000
Sensors Location Range	0-500
Data Range	10-10
Displacement Range	0-10
$n$	10
$ \Delta $	300
$ \Theta $	200
$U$	100
Time slots ( $T + 1$ )	11
$p$	varying between 0 and 1
$C$	varying between 0 and 50

Table 5.1: Range of parameters

<b>Comparison between CPLEX, RH, NRH and CG solution for <math>C = 3</math></b>				
$p$	CPLEX	RH	NRH	CG
0	20820	20126	20330	20686
0.1	19054.44	18496.41	18310.5	18967.92
0.2	17337.52	16903.04	16705.6	17201.6
0.3	15597.47	15256.81	15020.81	15449.12
0.4	13793.04	13528.32	13326	13641.6
0.5	11890.25	11766.5	11526.25	11784.75
0.6	9896.32	9815.82	9641.6	9830.88
0.7	7765.68	7690.77	7590.81	7733.34
0.8	5431.04	5375.29	5327.04	5399.52
0.9	2853.13	2786.47	2805.45	2809.34
1	0	0	0	0

Table 5.2: Effect of varying  $p$  on objective function value

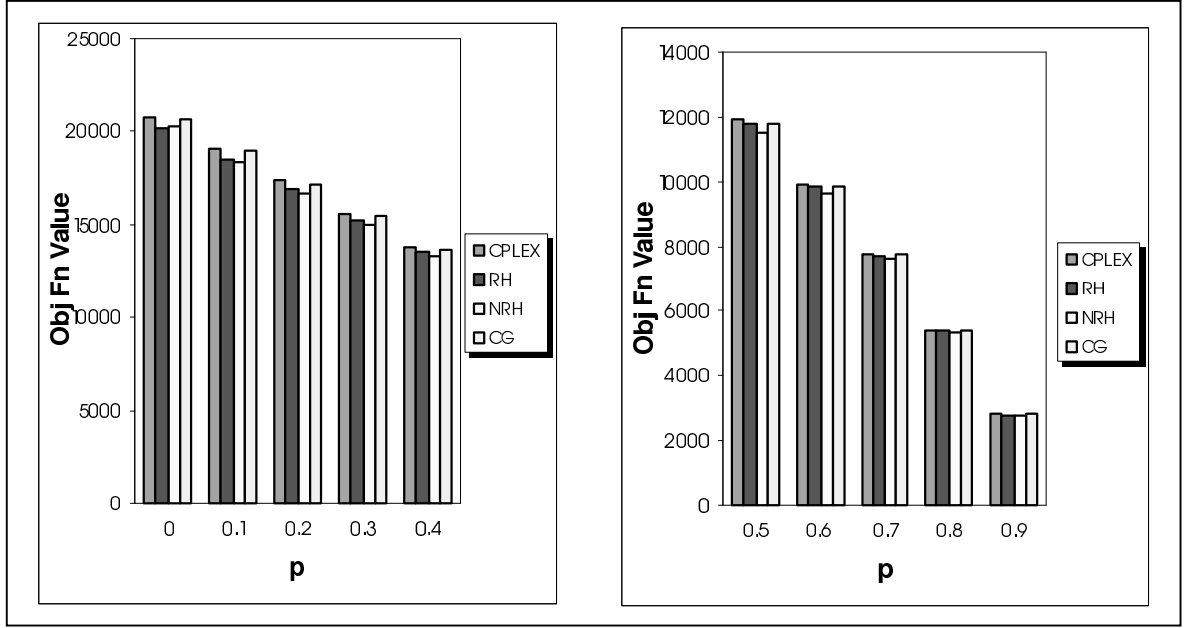


Figure 5.2: Effect of varying  $p$  on objective function value

In Table(5.2), we show the value of the objective function using CPLEX, RH, NRH and CG heuristic with varying  $p$  values for  $C = 3$  where the rest of the parameters are as defined in Table(5.1). Also the graph shown in Fig. 4.2 depicts that CG objective function value is almost equal to the CPLEX objective function since CG is a heuristic. We also observe that the gaps between the CG and CPLEX solutions are negligible.

### Comparison of solution time with varying $p$ value

In this section, we observe the effect of  $p$  varying on the solution times for both CPLEX and CG heuristic.

From Table(5.3), we clearly observe that problems with low values of  $p$  are tougher to solve with both CPLEX and CG heuristic. This may not be true in all cases for the CG heuristic since CG depends on the initial basic feasible solution. The solution time for RH and NRH does not vary with  $p$ . Also the graph shown in Fig. 5.3 clearly shows that CG performs better than solving the integrated problem (Dynamic MEXCLP

Solution time in seconds for $C = 3$				
$p$	CPLEX	RH	NRH	CG
0	1741	0.203	0.203	4.172
0.1	444	0.203	0.203	3.218
0.2	115	0.187	0.204	3.109
0.3	91	0.203	0.203	3.172
0.4	71	0.203	0.203	3.125
0.5	67	0.203	0.203	3.109
0.6	25	0.203	0.219	2.938
0.7	4	0.203	0.204	2.828
0.8	3	0.203	0.203	2.796
0.9	3	0.203	0.204	2.703

Table 5.3: Effect of varying  $p$  on solution time

Solution time for varying  $p$

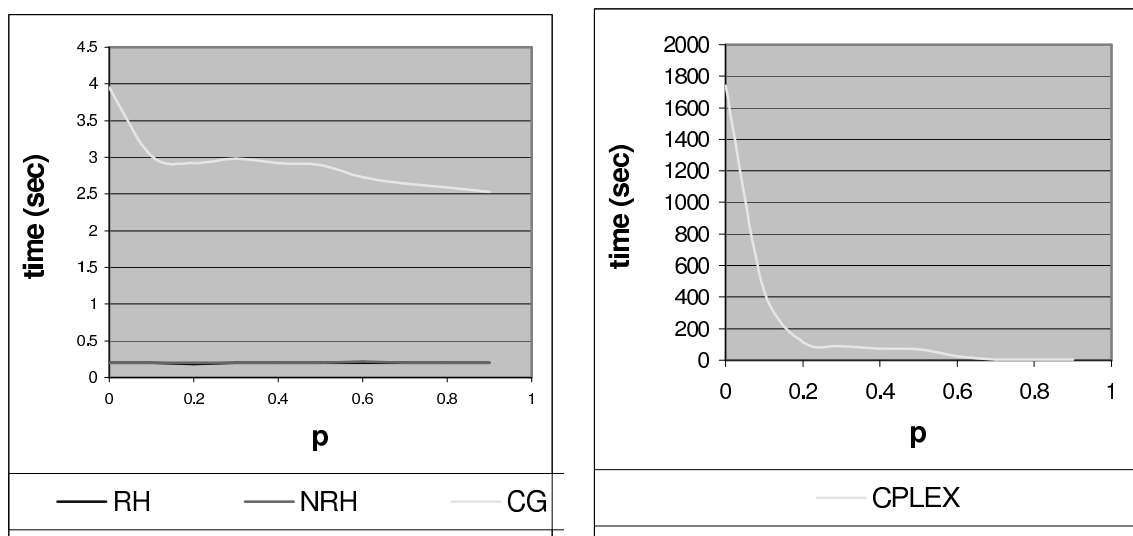


Figure 5.3: Effect of varying  $p$  on solution time

with relocation as explained in Section 2.3.1) with CPLEX.

The termination criteria selected for the CG heuristic is to stop if the gap between RMP solution and LP relaxation of the original problem is within 2 %.

### **5.2.2 Effect of varying $C$ on the model behavior**

The relocation cost is an important factor determining the frequency with which a facility changes its location between two consecutive time periods. In this section, we observe the effect of varying the relocation cost incurred for a unit change in the number of facilities at a location ( $C$ ) between two consecutive time periods.

#### **Comparison of objective function value with varying $C$ value**

In this section, we compare the objective function value obtained by CPLEX and CG heuristic. The objective function value decreases with increase in the relocation cost until a threshold is reached. Further increase in  $C$  has no effect on the objective function value. It is clearly depicted in the graph of Fig. 5.4. Also from the graph, we can see that CG procedure improves the heuristic solution for the cases where relocation cost is low. For high relocation cost NRH performs considerably good and CG is not able to improve it any further in the stipulated time (or in the limited iterations allowed).

#### **Comparison of solution time with varying $C$ value**

In this section, we observe the effect of  $C$  value on the solution time for both CPLEX and CG heuristic. CG tends to work well with low  $C$  value for all  $p$ 's, but with increasing relocation cost, mix results are seen. Fig. 5.5 shows the effect of  $C$  on the solution time for both CPLEX and CG for different  $p$  values.

### **5.2.3 Effect of varying $n$ on the model behavior**

Here we vary the number of facilities available for each time period and observe its effect on the objective function value and the solution time. For both CPLEX

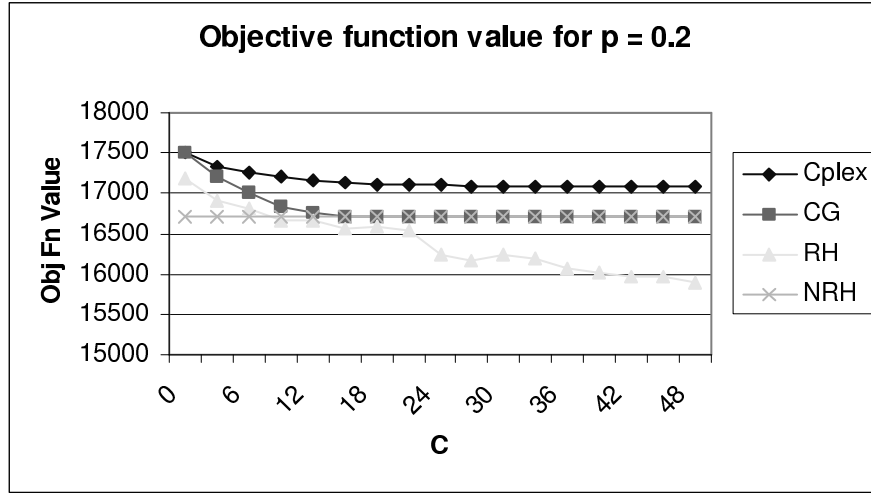


Figure 5.4: Effect of varying  $C$  on objective function value

and CG, the objective function value increases with increase in  $n$  until a threshold is reached. Beyond this, increase in  $n$  has no effect on the objective function value. CPLEX solution time is maximum for intermediate values of  $n$ . Whereas CG solution time is always negligible as compared to CPLEX solution time.

#### 5.2.4 CPLEX solution within CG solution time

In Table(5.4), the objective function value obtained using CPLEX and CG heuristic are shown along with the CPLEX solution found within CG solution time. In none of the above case a feasible solution was found using CPLEX within CG solution time.

#### 5.2.5 Improvement in solution quality using CG over RH and NRH

Here, we observe improvement achieved by CG over Relocation Heuristic (RH) and No Relocation Heuristic (NRH) solution. Table(5.5) and summarizes RH, NRH and CG solution for  $C = 3$  with varying  $p$ . Also % improvement of CG heuristic over RH and NRH is shown.

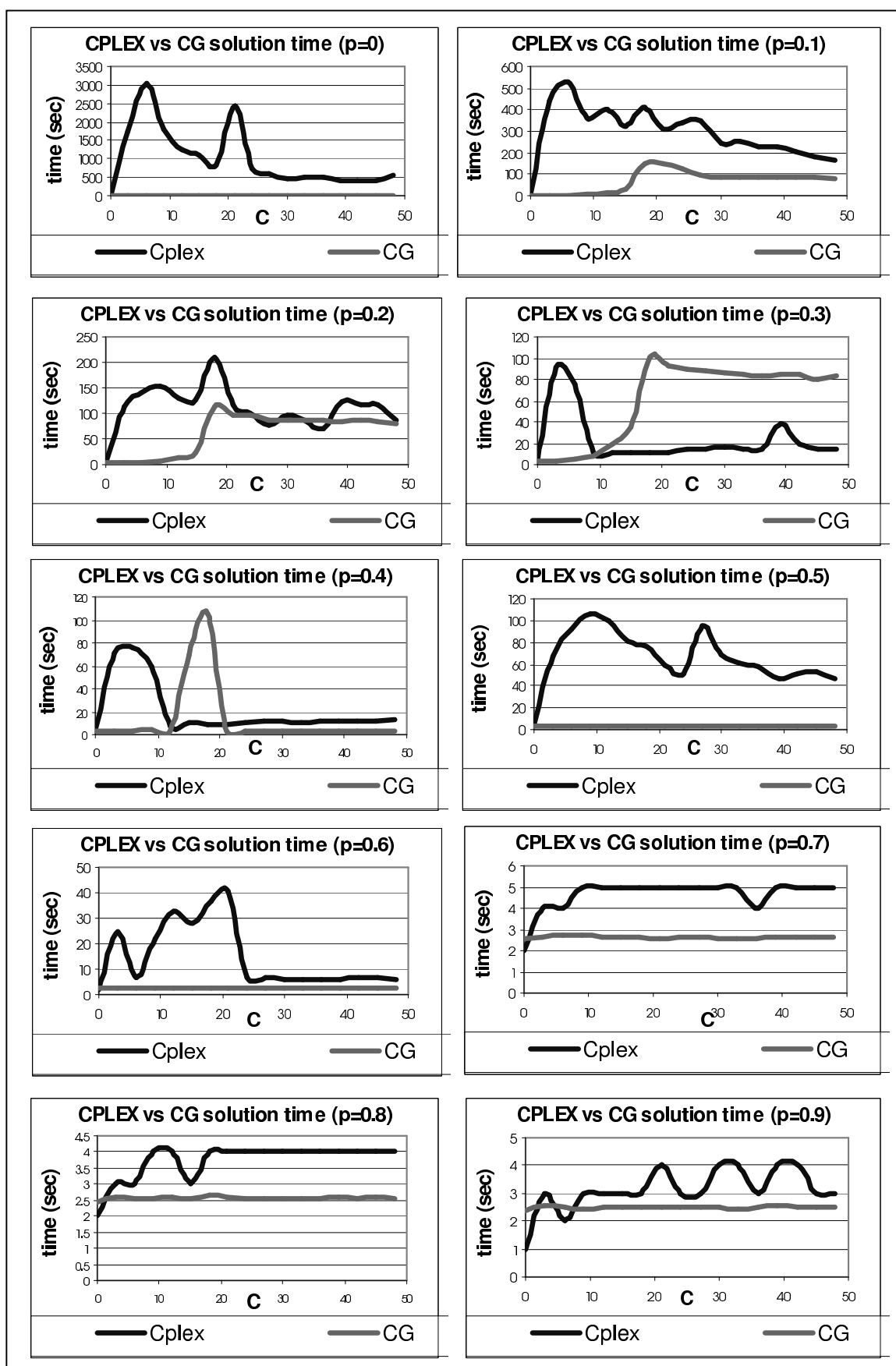


Figure 5.5: Effect of varying  $C$  on solution time for different values of  $p$



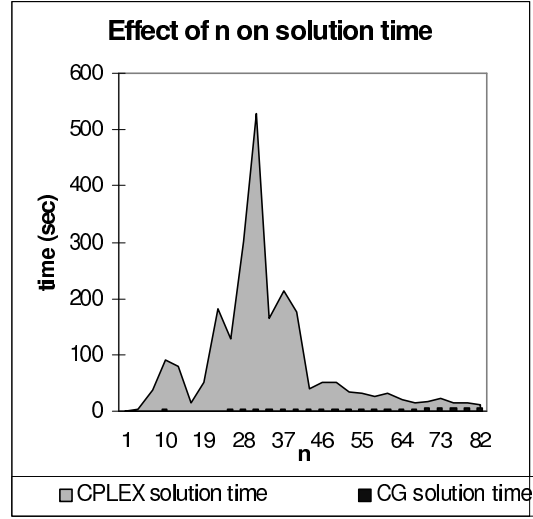


Figure 5.6: Effect of varying  $n$  on solution time

CPLEX solution value found within CG solution time for $C = 3$						
$p$	CPLEX		CG		% gap between (4) and (2)	CPLEX solution within CG time
	solution	time	solution	time		
(1)	(2)	(3)	(4)	(5)	(6)	(7)
0	20820	1741	20686	4.172	0.64	NFSF
0.1	19054.44	444	18967.92	3.218	0.45	NFSF
0.2	17337.52	115	17201.6	3.109	0.78	NFSF
0.3	15597.47	91	15449.12	3.172	0.95	NFSF
0.4	13793.04	71	13641.6	3.125	1.1	NFSF
0.5	11890.25	67	11784.75	3.109	0.89	NFSF
0.6	9896.32	25	9830.88	2.938	0.66	NFSF
0.7	7765.68	4	7733.34	2.828	0.42	NFSF
0.8	5431.03	3	5399.52	2.796	0.58	NFSF
0.9	2853.13	3	2809.34	2.703	1.33	NFSF
NFSF: No Feasible Solution Found						

Table 5.4: CPLEX objective function value within CG solution time

<b>RH, NRH and CG solutions for <math>C = 3</math></b>					
$p$	RH	NRH	CG	% improvement over RH	% improvement over NRH
0	20126	20330	20686	2.71	1.72
0.1	18496.41	18310.5	18967.92	2.49	3.47
0.2	16903.04	16705.6	17201.6	1.74	2.88
0.3	15256.81	15020.81	15449.12	1.24	2.77
0.4	13528.32	13326	13641.6	0.83	2.31
0.5	11766.5	11526.25	11784.75	0.15	2.19
0.6	9815.82	9641.6	9830.88	0.15	1.93
0.7	7690.77	7590.81	7733.34	0.55	1.84
0.8	5375.29	5327.04	5399.52	0.45	1.34
0.9	2786.47	2805.45	2809.34	0.81	0.14

Table 5.5: Comparison between RH, NRH and CG solution

### 5.3 Analysis of Variance

Here we carry out an experiment to determine whether there is a significant difference in the solution methodologies viz. CPLEX and CG (using heuristic solution as the initial basic feasible solution) in terms of the following stated performance measures (Henceforth will be referred as responses).

1. Response 1: Deviation (percentage gap between objective function of the solution methodology and the LP relaxation of the problem)
2. Response 2: Solution Time

The primary motive of this effort is to compare the solution methodologies with the above defined responses at different combinations of the problem parameters (also referred as factors) and to verify the effect on responses with interactions between the parameters.

The following factors were considered to be relevant to the quality of the response:-

1. Solution Methodology
2. Number of Facility

3. Number of Potential Location
4. Number of Demand Node (Sensors)
5. Velocity (Displacement)
6. Demand Data (Data Range)
7. Probability of failure
8. Relocation Cost
9. Number of time slots

In this analysis, coverage radius was ignored, since displacement and coverage radius are correlated.

### 5.3.1 Screening Experiment: Fractional Factorial Design

Considering the enormity of the number of factors, that may influence the response, it was decided that a screening experiment be performed to begin with, and screen out the insignificant factors and their insignificant higher level interactions, so that the responses could be analyzed and explained in terms of the significant ones.

A  $2^{k-m}$  fractional factorial design of resolution IV was designed as the screening experiment with a single replicate. This means each of the factor would be considered at 2 levels arbitrarily (Low and High designated as  $-1$  and  $+1$  respectively) and the  $k$  refers to the number of factors = 9 and  $m = 4$ . A single replicate would require only 32 runs as compared to  $2^9$  experiments if a full factorial design was conducted. By choosing such a design we usually assume that the high order interactions are negligible and this greatly simplifies the problem structure.

The  $\alpha$ -level or level of significance considered was 0.05. The level of the factors chosen are as shown in Table(5.6).

Sr. No.	Factors	Low level (-1)	High level (+1)
A	Method	CPLEX	CG
B	Probability of Failure (p)	0.2	0.7
C	Relocation Cost (C)	10	100
D	Number of Facility (n)	5	10
E	Number of Potential Location (Delta)	100	200
F	Number of Sensors (Theta)	100	200
G	Number of Time Slots (T)	5	15
H	Data Range	10	50
J	Displacement	10	40
Coverage radius is fixed at 80			

Table 5.6: Fractional Factorial Design: Parameter Value

### Design of Fractional Factorial experiment: Minitab Results

The screening experiment was carried out on Minitab and the Minitab results are as follows:

Factors: 9 Base Design: 9, 32 Resolution: IV  
Runs: 32 Replicates: 1 Fraction: 1/16  
Blocks: none Center pts (total): 0  
Design Generators: F = BCDE G = ACDE H = ABDE J = ABCE

Defining Relation: I = BCDEF = ACDEG = ABDEH = ABCEJ = ABFG = ACFH  
= ADFJ = BCGH = BDGJ = CDHJ = DEFGH = CEFGJ = BEFHJ = AEGHJ =  
ABCDGHI

The Randomized data matrix generated by Minitab and the response value obtained is as shown in Table(5.7).

Response variable 1: Deviation from LP Relaxation:

In the following table we show results only for the parameters and interactions which were found significant.

Term	Effect	Coef	SE Coef	T	P
Method	1.7355	0.8677	0.02851	30.44	0.021
T	0.6541	0.327	0.02851	11.47	0.055
p*Data Range	0.648	0.324	0.02851	11.37	0.056
n*Delta	0.6766	0.3383	0.02851	11.87	0.054

Thus we see that Solution Methodology, number of Time Slots, probability of link

Std Order	Run Order	Method	p	C	n	Delta	Theta	T	Data Range	D.ment	Dev	Time
1	1	-1	-1	-1	-1	-1	1	1	1	1	0.00	2.44
28	2	1	1	-1	1	1	-1	-1	1	-1	1.80	0.61
32	3	1	1	1	1	1	1	1	1	1	2.70	107.89
31	4	-1	1	1	1	1	1	-1	-1	-1	0.00	1.67
13	5	-1	-1	1	1	-1	1	1	-1	-1	0.00	15.31
6	6	1	-1	1	-1	-1	-1	1	-1	1	3.34	26.41
5	7	-1	-1	1	-1	-1	-1	-1	1	-1	0.00	0.45
18	8	1	-1	-1	-1	1	-1	1	1	1	0.33	1.13
26	9	1	-1	-1	1	1	1	-1	-1	1	2.34	15.41
15	10	-1	1	1	1	-1	-1	1	1	1	0.85	7.34
11	11	-1	1	-1	1	-1	1	-1	1	-1	0.00	2.83
16	12	1	1	1	1	-1	-1	-1	-1	-1	0.20	0.11
19	13	-1	1	-1	-1	1	1	-1	1	1	0.00	1.34
24	14	1	1	1	-1	1	-1	-1	-1	1	0.51	0.13
14	15	1	-1	1	1	-1	1	-1	1	1	2.04	5.11
7	16	-1	1	1	-1	-1	1	-1	-1	1	0.00	1.16
2	17	1	-1	-1	-1	-1	1	-1	-1	-1	1.97	0.11
17	18	-1	-1	-1	-1	1	-1	-1	-1	-1	0.00	1.08
21	19	-1	-1	1	-1	1	1	1	-1	1	0.00	23.34
4	20	1	1	-1	-1	-1	-1	-1	1	1	1.84	0.08
10	21	1	-1	-1	1	-1	-1	1	1	-1	1.78	0.28
8	22	1	1	1	-1	-1	1	1	1	-1	1.57	0.30
12	23	1	1	-1	1	-1	1	1	-1	1	2.15	9.58
22	24	1	-1	1	-1	1	1	-1	1	-1	0.00	0.17
20	25	1	1	-1	-1	1	1	1	-1	-1	2.65	12.31
29	26	-1	-1	1	1	1	-1	-1	1	1	0.00	4.11
3	27	-1	1	-1	-1	-1	-1	1	-1	-1	0.00	1.16
9	28	-1	-1	-1	1	-1	-1	-1	-1	1	0.11	8.92
23	29	-1	1	1	-1	1	-1	1	1	-1	0.00	2.78
27	30	-1	1	-1	1	1	-1	1	-1	1	0.72	20.34
25	31	-1	-1	-1	1	1	1	1	1	-1	0.47	26.33
30	32	1	-1	1	1	1	-1	1	-1	-1	4.69	70.73

Table 5.7: Fractional Factorial Design: Data Matrix (Randomized)

failure, Data Range, number of facilities and number of potential locations directly affects or the interactions between these parameters have effect on the response variable Deviation.

Response variable 2: Time to solve:

From the ANOVA table it was observed that none of the factors or the two-level interactions are significant. But by experience we know that some of the factors do affect the time to solve a problem. Hence, we increased the level of significance ( $\alpha = 0.18$ ) to take into account at least a few important factors and interactions as shown in the following table.

Term	Effect	Coef	SE Coef	T	P
n	13.888	6.944	1.702	4.08	0.153
Delta	12.987	6.494	1.702	3.82	0.163
T	17.775	8.887	1.702	5.22	0.12
Method*C	11.226	5.613	1.702	3.3	0.187
p*Data Range	12.376	6.188	1.702	3.64	0.171
n*Delta	11.714	5.857	1.702	3.44	0.18
Delta*T	12.269	6.134	1.702	3.6	0.172

The number of cluster heads, number of potential location, number of time slots, solution method, relocation cost , probability of failure and data range has an effect on the solution time either directly or through second level interactions.

From the screening experiment, the significant factors were noted and a parametric experiment was run at different levels of the significant factors.

### 5.3.2 Parametric Experiment

When the significant factors were screened out from the screening experiment for both the responses, it is observed that almost the same parameters affected the responses. Since the ultimate aim of this experiment is to compare both the solution methodologies with respect to Solution quality and Solution time, same factors were tested for the responses. We performed a six-way ANOVA with the factors listed in Table(5.8) with an additional factor of Solution Method (CG and CPLEX). The result showed that the Solution Method was significant for both Deviation and Solution Time.

The means for deviation and time are plotted for both CG and CPLEX. It is

evident from the plot that CG performs better than CPLEX in terms of Solution time. Whereas CPLEX gives a better objective function value. Fig. 5.7 and Fig. 5.8 shows the mean deviation and time for both the solution methodology.

### Main Effects Plot - Data Means for DEV

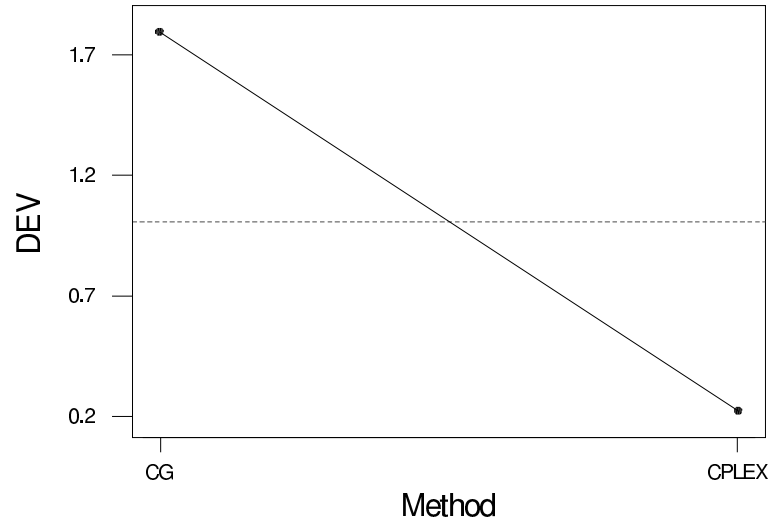


Figure 5.7: CG vs CPLEX: Deviation

Since solution method is itself significant and thus the response variable becomes dependent on what solution methodology was chosen. Thus in order to see the true effect of the parameters on the response variables, we perform a five-way ANOVA for each solution methodology. The levels of the factors are shown in Table(5.8).

Factor	No. of levels	Value			
		1	2	3	4
Facilities (n)	2	5	10	*	*
Time Slots (T)	2	5	10	*	*
Location (Delta)	3	200	400	600	*
p	3	0.1	0.5	0.9	*
C	4	5	20	80	320

Table 5.8: Parametric Experiment: Factor Levels and Value

This defines a Five-way ANOVA for each method which has 5 main effects; 10, 2-

### Main Effects Plot - Data Means for Time

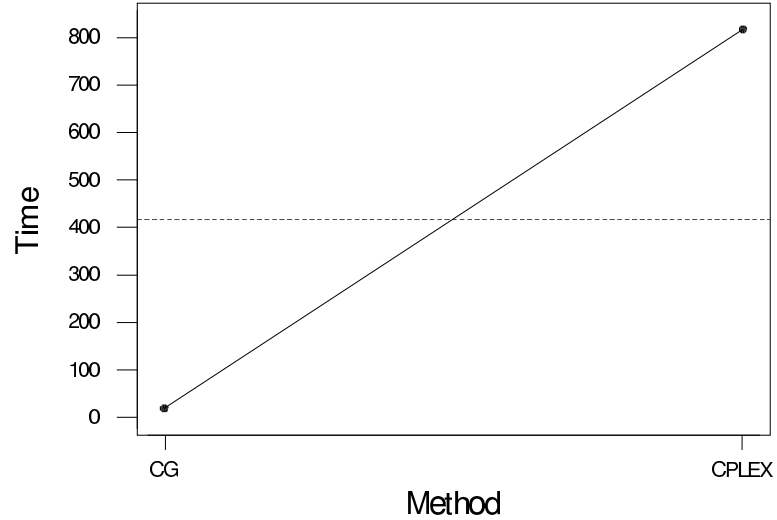


Figure 5.8: CG vs CPLEX: Time

level interactions; 10, 3-level interactions; 5, 4-level interactions and finally 1, 5-level interaction.

By the Sparsity of Effects Principle [44], most of the systems are dominated by some of the main effects and low order interactions, and most high order interactions are negligible. Also higher levels of interactions are difficult to interpret. Thus in this analysis, we will restrict ourselves to the 5 main effects and 10, 2-level interactions only. We throw the high level of interaction as error term in the model.

#### Parametric Experiment: Minitab Results for CG

An experiment was conducted with the 5 factors at the levels described in Table(5.8). The Minitab results for the parametric experiment is as follows:

Response variable 1: Deviation from LP Relaxation:

In the Table(5.9) the factors and the interactions which were found to be significant at  $\alpha = 0.05$  significance level is listed.

Response variable 2: Time to solve:



Source	DF	Seq SS	Adj SS	Adj MS	F	P
n	1	2.6881	2.6881	2.6881	4.67	0.033
T	1	32.2923	32.2923	32.2923	56.09	0
Delta	2	3.511	3.511	1.7555	3.05	0.052
p	2	54.276	54.276	27.138	47.13	0
C	3	66.7008	66.7008	22.2336	38.62	0
n*C	3	17.0839	17.0839	5.6946	9.89	0
T*C	3	19.1042	19.1042	6.3681	11.06	0
p*C	6	144.5894	144.5894	24.0982	41.85	0

Table 5.9: Significant Factors and Interactions: CG Deviation

### CG Main Effects Plot - Data Means for DEV

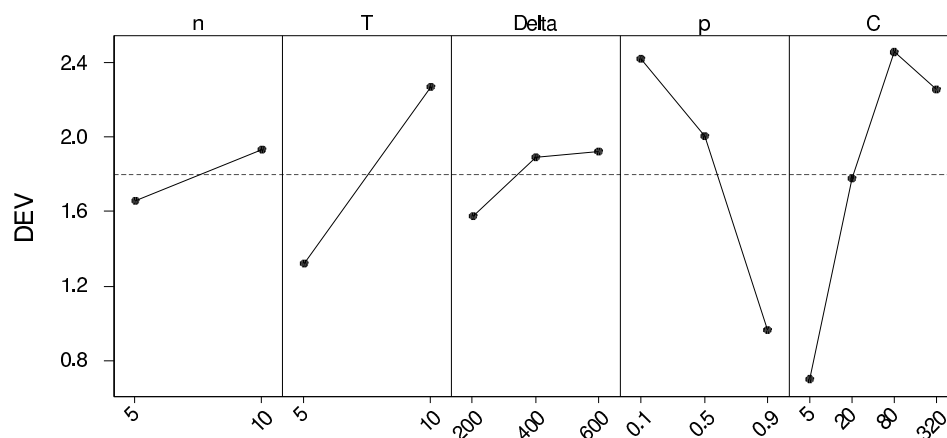


Figure 5.9: CG Main Effect Plot for Deviation

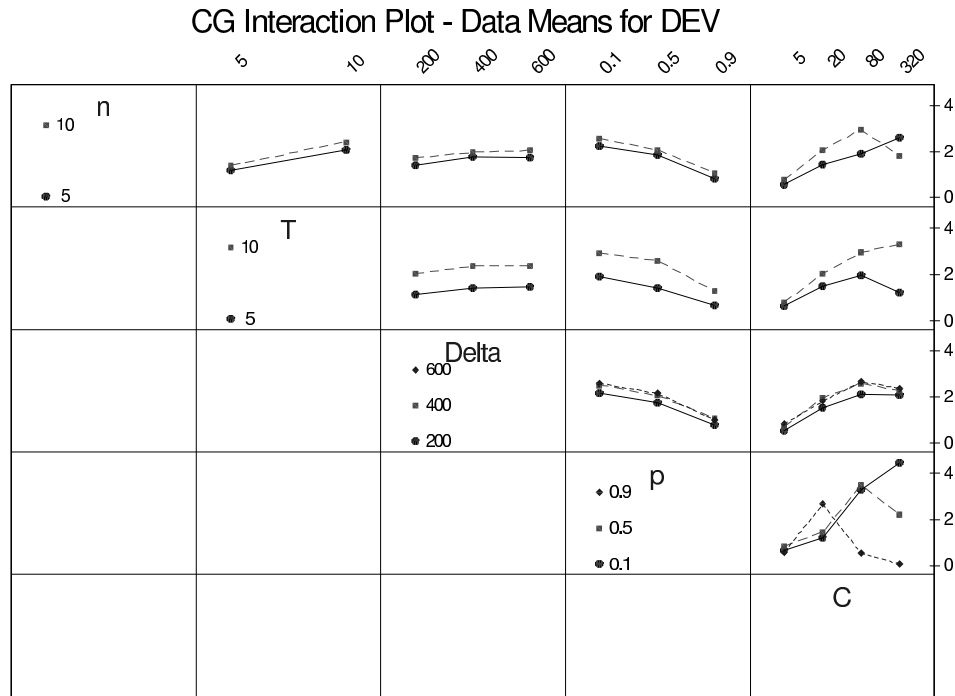


Figure 5.10: CG Interaction Plot for Deviation

Source	DF	Seq SS	Adj SS	Adj MS	F	P
n	1	10066.5	10066.5	10066.5	25.34	0
T	1	10512.3	10512.3	10512.3	26.46	0
Delta	2	7218.5	7218.5	3609.3	9.08	0
p	2	10649.6	10649.6	5324.8	13.4	0
C	3	26872.7	26872.7	8957.6	22.55	0
n*T	1	1750.6	1750.6	1750.6	4.41	0.038
n*C	3	12942	12942	4314	10.86	0
T*p	2	3017.2	3017.2	1508.6	3.8	0.026
T*C	3	7437.1	7437.1	2479	6.24	0.001
p*C	6	31892.9	31892.9	5315.5	13.38	0

Table 5.10: Significant Factors and Interactions: CG Time

### CG Main Effects Plot - Data Means for Time

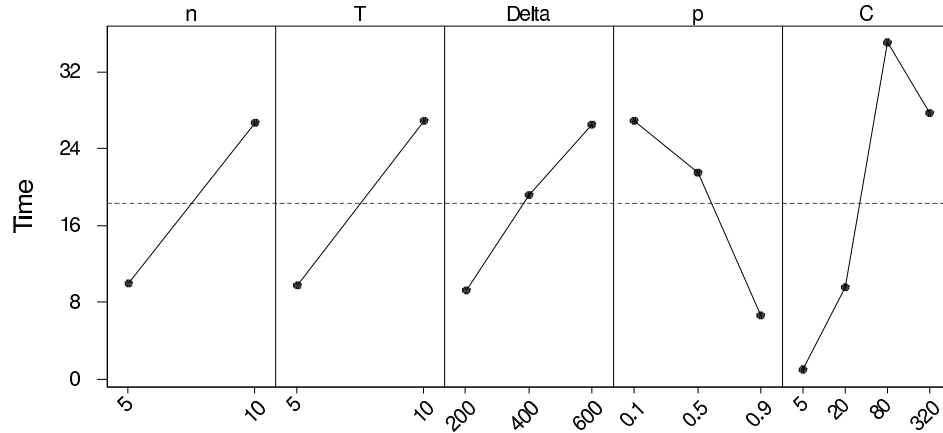


Figure 5.11: CG Main Effect Plot for Solution Time

Table(5.10) shows that all the parameters are significant for response variable Time. The main effect and the interaction plots for Deviation are shown in Fig. 5.9 and Fig. 5.10 respectively. The main effect and the interaction plots for Time are shown in Fig. 5.11 and Fig. 5.12 respectively.

### Parametric Experiment: Minitab Results for CPLEX

The same set of parameters were used for both CG and CPLEX and the analysis for CPLEX are presented as follows.

#### Response variable 1: Deviation from LP Relaxation:

In Table(5.11), the significant factors and the interactions affecting Deviation of the CPLEX solution from LP relaxation are listed. The main effect plot and the interaction plots are as shown in Fig. 5.13 and Fig. 5.14 respectively.

#### Response variable 2: Time to solve:

From Table(5.12), we see that except Delta (Number of Potential Locations) all the factors are significant for response Time, whereas for Deviation all are significant. The main effect plot and the interaction plot for Time is as shown in Fig. 5.15 and Fig. 5.16 respectively.

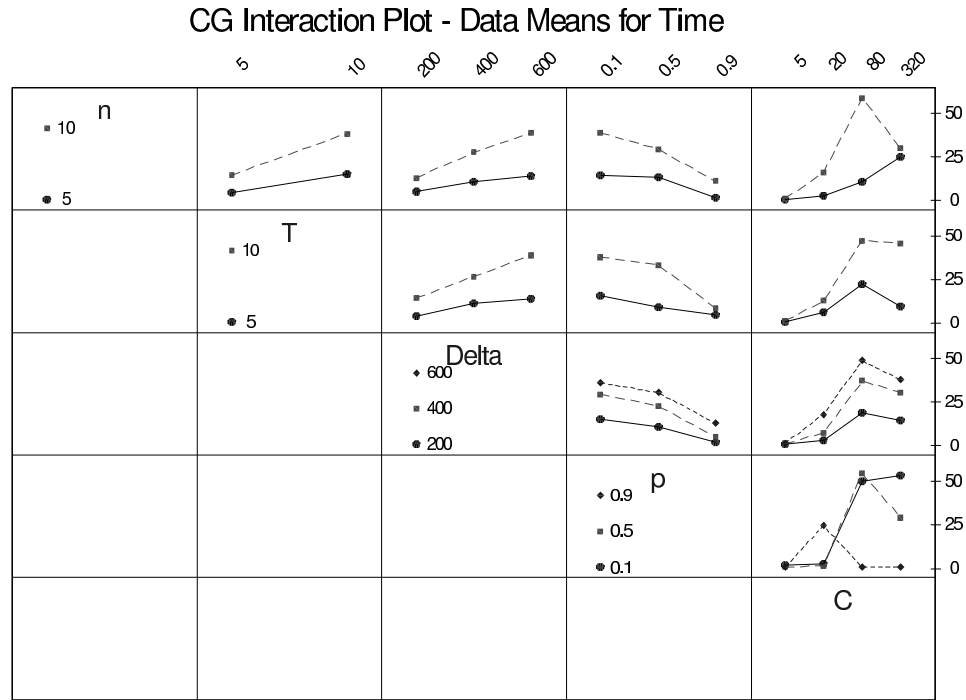


Figure 5.12: CG Interaction Plot for Solution Time

Source	DF	Seq SS	Adj SS	Adj MS	F	P
n	1	1.67057	1.67057	1.67057	37.72	0
T	1	0.99298	0.99298	0.99298	22.42	0
Delta	2	0.5228	0.5228	0.2614	5.9	0.004
p	2	5.42748	5.42748	2.71374	61.27	0
C	3	1.50217	1.50217	0.50072	11.31	0
n*Delta	2	0.32843	0.32843	0.16421	3.71	0.028
n*p	2	1.68803	1.68803	0.84402	19.06	0
T*p	2	0.44955	0.44955	0.22478	5.07	0.008
T*C	3	0.55531	0.55531	0.1851	4.18	0.008
Delta*p	4	0.47685	0.47685	0.11921	2.69	0.035
p*C	6	1.01118	1.01118	0.16853	3.81	0.002

Table 5.11: Significant Factors and Interactions: CPLEX Deviation

CPLEX Main Effects Plot - Data Means for DEV

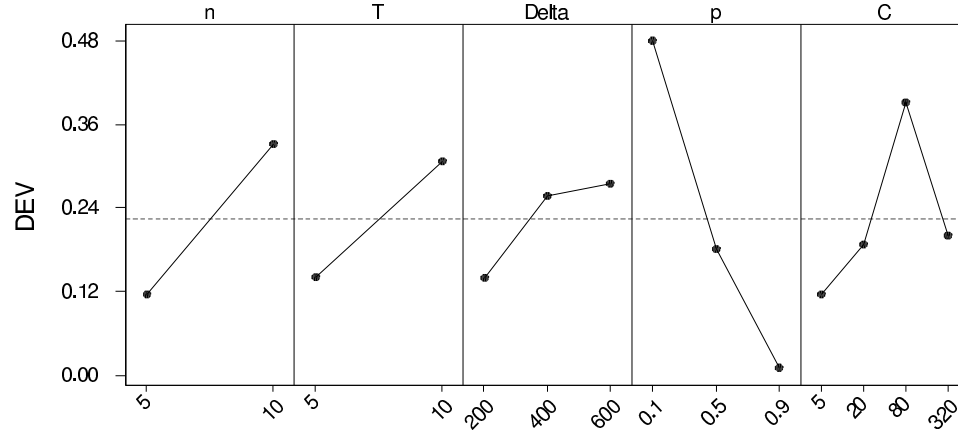


Figure 5.13: CPLEX Main Effect Plot for Deviation

Source	DF	Seq SS	Adj SS	Adj MS	F	P
n	1	87568068	87568068	87568068	22.36	0
T	1	40742691	40742691	40742691	10.4	0.002
p	2	1.66E+08	1.66E+08	82873669	21.16	0
C	3	59429417	59429417	19809806	5.06	0.003
n*T	1	37309706	37309706	37309706	9.53	0.003
n*p	2	1.55E+08	1.55E+08	77349953	19.75	0
n*C	3	53810388	53810388	17936796	4.58	0.005
T*p	2	66955727	66955727	33477864	8.55	0
p*C	6	1.04E+08	1.04E+08	17357108	4.43	0

Table 5.12: Significant Factors and Interactions: CPLEX Time

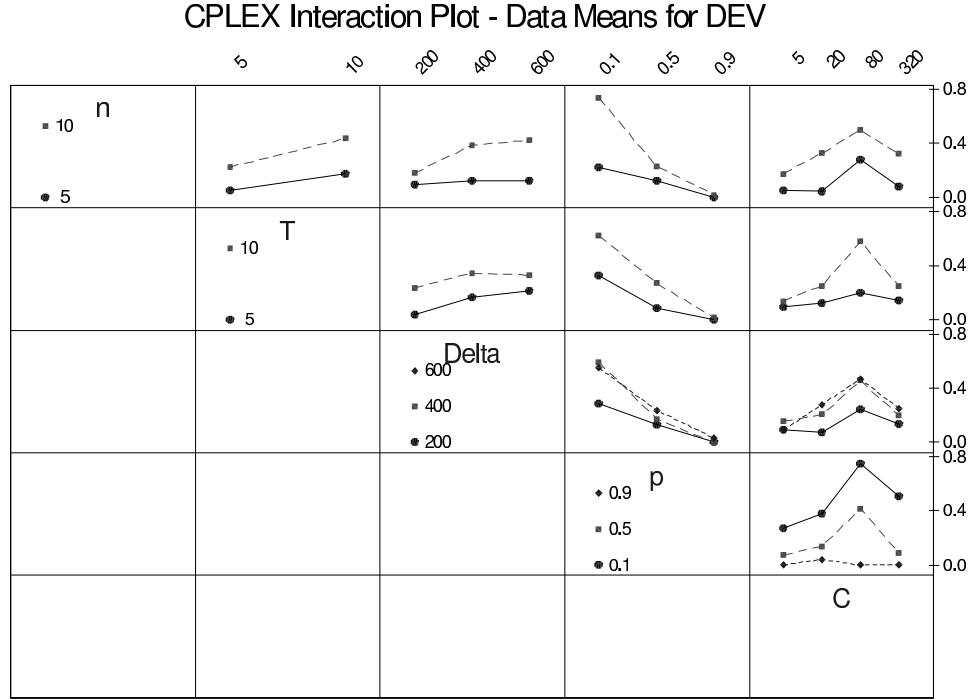


Figure 5.14: CPLEX Interaction Plot for Deviation

### 5.3.3 Discussion

The parametric experiment validates that CG performs better in terms of solution time and the mean Deviation is 1.8%. From the main effects plot for CG time and CPLEX time, we observe that solution time decreases with increasing  $p$ . Solution time is maximum for intermediate values of relocation cost. Other effects observed were with increasing number of facilities ( $n$ ), number of Time Slots ( $T$ ) and number of potential Locations ( $\Delta$ ), the solution time increases. It is obvious that with increasing  $T$ ,  $\Delta$  the time increases since the problem size increases. The effect of  $n$  on time does not truly reflect the behavior since we had only two levels of  $n$ . In Section 5.2, we observed the solution time with increasing  $n$  and found that for intermediate values of  $n$  it takes the maximum time to solve the problem. A similar pattern is observed for Deviation.

CPLEX Main Effects Plot - Data Means for Time

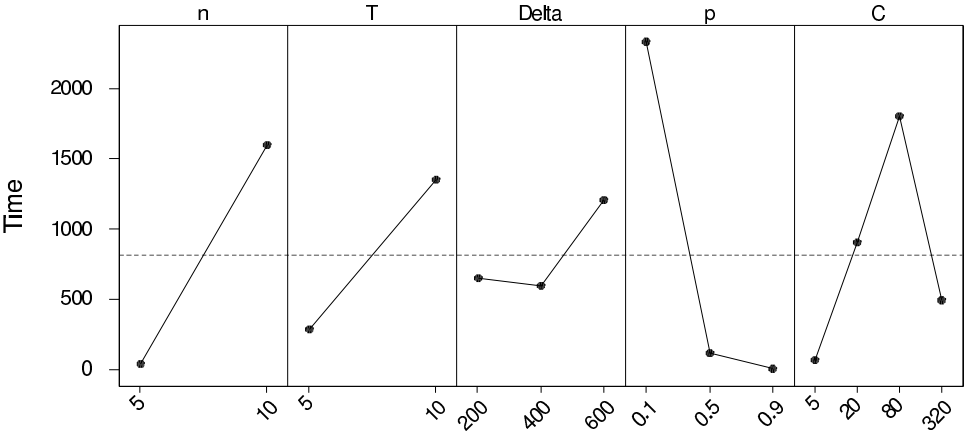


Figure 5.15: CPLEX Main Effect Plot for Solution Time

CPLEX Interaction Plot - Data Means for Time

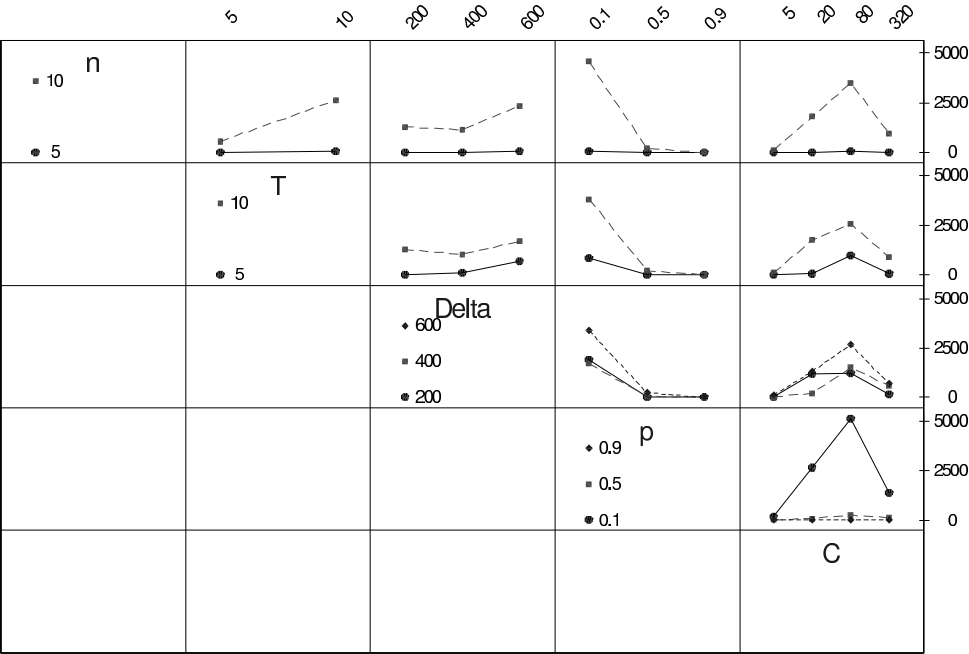


Figure 5.16: CPLEX Interaction Plot for Solution Time

Number of Facilities = 15 - 30; Number of Potential Locations = 800 - 1000 Number of Sensors = 400 - 500; Number of Time Slots = 10 -15; Probability of Link Failure = 0.1 - 0.3; Relocation Cost = 60 - 100 Demand = 50 - 100; Coverage Radius = 80; Displacement = 0 - 60						
Sr. No.	CG Solution	CG Time (sec)	CPLEX Solution	CPLEX Time	CG % gap with LP	% Improvement over heuristic solution
1	389143.47	321.77	NFSF	> 4 hrs	3.73	1.46
2	379934.40	4.99	NFSF	> 4 hrs	1.57	0.00
3	318656.64	4.34	NFSF	> 4 hrs	1.36	0.00
4	340129.02	4.84	NFSF	> 4 hrs	1.32	0.00
5	472456.44	6.77	NFSF	> 4 hrs	0.95	0.00
NFSF: No Feasible Solution Found						

Table 5.13: Computational Results

### 5.3.4 Computational Results for large size problem

The analysis of variance revealed the effect of each parameter on the solution time and deviation. In this section results are presented for instances that showed the maximum computational time to solve. Here we terminate the CG procedure if the Relaxed Master Problem is within 2% of the LP relaxation.

From Table(5.13), we can see that CPLEX is not able to find a feasible solution within 4 hrs of computational time in all of this cases. Whereas Column Generation (CG) heuristic along with the RH and NRH heuristic finds a feasible solution within 4% of the LP relaxation. We know that solution of RH and NRH heuristic forms the initial basic feasible solution for the CG. In the last 4 instances, we can see that the heuristic solution is itself within 2% of the LP relaxation and hence it shows the solution time less than 10 seconds. In the first instance the CG performs 11 iterations before terminating and solving the Integer Master Problem. The Relaxed Master Problem is within 2% of the LP relaxation and the final integer solution gap is 3.73% from the LP relaxation. Also it shows that CG improves over the heuristic solution by 1.46%.



## 5.4 Regret Analysis

In the previous chapters and sections, we provided a tool for solving the Dynamic MEXCLP with relocation to maximize the expected demand coverage. The optimal facility location strategy was formulated assuming that the probability of link failure is known. However, this may not be true in some scenarios. We may have knowledge of the range within which  $p$  lies, as opposed to the exact value of  $p$ . To determine the optimal location strategy in such a scenario, the “minimax regret analysis” is carried out. Minimax regret analysis is outlined as follows:

Let us consider a range of probabilities of failure  $P \in [p_l, p_u]$  such that  $p_l < p_u$ . Let the optimal solution corresponding to probability of failure  $p_j$  be  $s_j \in S$  and the corresponding objective function value be  $f(s_j, p_j)$ .

Let  $r(s_j, p_i) = f(s_i, p_i) - f(s_j, p_i)$ , ( $i \neq j$ ) denote the regret (or loss incurred in expected coverage) due to selection of the solution  $s_j$  for probability  $p_i$ , instead of the solution  $s_i$ . We note here that  $r(s_i, p_i) = 0$ .

Let  $r(s_j) = \max_{p_i \in P} r(s_j, p_i)$ , which denotes the maximum regret corresponding to solution  $s_j$  over  $P$ . Our objective is to determine the solution  $s_j \in S$  that minimizes the maximum regret, expressed mathematically as follows:

$$\min_{s_j \in S} r(s_j), \text{ i.e. } \min_{s_j \in S} \max_{p_i \in P} r(s_j, p_i)$$

Let us consider a scenario with the parameters as listed in Table(5.14). Here we carry out the minimax regret analysis with the optimal solutions obtained for the given range of probabilities at intervals of 0.1.

Table(5.15) shows the objective function value obtained by selecting a solution for a particular probability of link failure  $p$ . The optimal solution for each  $p$  value are shown in bold.

Table(5.16) displays the regret of selecting a particular solution for different values of  $p$  in the given range. The bold values in the first 6 columns represent the maximum regret for that particular solution and are listed in the last column. Whereas the bold value in the last column corresponding to the solution  $s_{0.5}$  is the minimax

Parameter	Range/Value
Potential Location Range	0-1000
Sensors Location Range	0-500
Data Range	10-50
Displacement Range	0-10
$n$	10
$ \Delta $	400
$ \Theta $	200
$D$	50
Time slots ( $T + 1$ )	6
$C$	20
$p$	0.3-0.7

Table 5.14: Range of parameters for regret analysis

Objective Function Value $f(s_j, p_i)$					
Solutions	$p = 0.3$	$p = 0.4$	$p = 0.5$	$p = 0.6$	$p = 0.7$
$s_{0.3}$	<b>10969.45</b>	9391.80	7806.25	6212.80	4611.45
$s_{0.4}$	10949.46	<b>9414.64</b>	7860.5	6287.04	4694.26
$s_{0.5}$	10908.43	9399.12	<b>7867.75</b>	6314.32	4738.83
$s_{0.6}$	10855.61	9374.44	7866.25	<b>6331.04</b>	4768.81
$s_{0.7}$	10571.65	9189.20	7762.25	6290.80	<b>4774.85</b>

Table 5.15: Objective function table

Regret Function value $r(s_j, p_i)$						
Solutions	$p = 0.3$	$p = 0.4$	$p = 0.5$	$p = 0.6$	$p = 0.7$	Maximum regret
$s_{0.3}$	0	22.84	61.5	118.24	<b>163.4</b>	163.4
$s_{0.4}$	19.99	0	7.25	44	<b>80.59</b>	80.59
$s_{0.5}$	<b>61.02</b>	15.52	0	16.72	36.02	<b>61.02</b>
$s_{0.6}$	<b>113.84</b>	40.2	1.5	0	6.04	113.84
$s_{0.7}$	<b>397.8</b>	225.44	105.5	40.24	0	397.8
Minimax regret						61.02

Table 5.16: Regret table

regret obtained. Thus the optimal solution with least maximum regret is the optimal solution for  $p = 0.5$  ( $s_{0.5}$ ) with maximum regret of 61.02 for  $p = 0.3$ .

Since we considered solutions for  $p$  in intervals of 0.1, we might have missed few more solutions for  $p$  values between the intervals used for the analysis. Thus we carry out the analysis for the same set of parameters (or same scenario) with an interval of 0.01 for  $p \in [0.3, 0.7]$ . Due to space constraint we do not show the entire tabular results. Instead we present only the final result.

The optimal solution with least maximum regret is  $s_{[0.45, 0.48]}$  with the maximum regret of 58.71. Regret analysis aids in making decision when the value of the parameters is uncertain.

## 5.5 Chapter Summary

In this chapter, a detailed description of the software implementation is presented followed by a detailed analysis of a scenario. Analysis of Variance is conducted to determine the effect of the problem parameters on the response variable Time and Deviation. The results of the experiment shows that CG performs better in terms of computational time and CPLEX performs better in terms of solution quality. Computational Results for some large size problems are also shown. A regret analysis is carried out to find the solution with least maximum regret for all values of  $p$  within the specified range.

## Chapter 6

# Conclusion and Future Research

In this work, we modeled a communication network problem of finding optimal location of AWACS for collecting data from sensors spread over a geographically dispersed area for the purpose of data fusion. We model the problem as a covering location problem incorporating issues of link failure and mobility of the system entities (sensors). Several variants of the problems were modeled. One of the variant deals with clustering sensors in wireless *ad hoc* network in order to maximize the expected demand covered. In this variant, the cluster heads are elected among the set of sensors and thus they are also mobile. Thus our work can be viewed as a contribution to the covering location literature as well as a new clustering technique for the *ad hoc* networks.

Two greedy heuristics and a Column Generation heuristic is proposed to solve the Dynamic MEXCLP model. The results show that the Column Generation heuristic used in conjunction with the two greedy heuristic: Relocation Heuristic (RH) and No-Relocation Heuristic (NRH) performs better in terms of computational time than solving Dynamic MEXCLP using commercially available CPLEX mixed integer program (MIP) solver. Also the results of the analysis of variance shows that CG gap with LP relaxation of the original problem is within 1.8%.

We can conclude that the solution methodology proposed in this work solves the problem with a resonable accuracy, efficiently. Although we have not completely captured the scenario since we are dealing at only one level of the data fusion network

architecture. Future work can involve locating the cluster heads considering that all clusterheads should have atleast one path between themselves in order to share the data collected from the sensors in their clusters. In this work we assume that each cluster head is connected to other clusterheads via a virtual “clusterhead manager”.

This work does not capture the issue of load balancing, i.e. capacity of each cluster head in terms of handling data or handling number of sensors.

Developing a simulation model with commercially available network simulator such as OPNET and integrating the solution methodology could be a mechanism to validate the performance of these clusters under realistic (simulated) conditions.

# Bibliography

- [1] This is the definition developed by the Joint Directors of Laboratories Data Fusion Subpanel, with two revisions introduced by the authors:(1) The function of detection has been added, and (2) the estimation of position has been replaced by estimation of state to include broader concept of kinematic state (e.g., higher-order derivatives, velocity) as well as other states of behavior(e.g., electronic state, fuel state).
- [2] I. Averbakh and O. Berman. Minimax regret median location on a network under uncertainty. *INFORMS Journal on Computing*, 12(2):104–110, SPR 2000.
- [3] J. Balakrishnan, F. R. Jacobs, and M. A. Venkataramanan. Solutions for the constrained dynamic facility layout problem. *European Journal of Operational Research*, 57(2):280–286, MAR 1992.
- [4] M. O. Ball and F. L. Lin. A reliability model applied to emergency service vehicle location. *Operations Research*, 1993.
- [5] M. O. Ball, J. S. Provan, and D. R. Shier. Reliability covering problems. *Networks*, 21:345–357, 1991.
- [6] S. Basagni. Distributed clustering for ad hoc networks. *International Symposium on Parallel Architectures, Algorithms, and Networks*, 1999.
- [7] S. Basagni, I. Chlamtac, and A. Farago. A generalized clustering algorithm for peer-to-peer networks. *ICALP '97*, 1997.

- [8] P. Basu, N. Khan, and T. D. C. Little. A mobility based metric for clustering in mobile ad hoc networks. *IEEE*, 2001.
- [9] R. Batta. Comment on the dynamics of plant layout. *Management Science*, 33(8):1065–1065, AUG 1987.
- [10] R. Batta, J. M. Dolan, and N. N. Krishnamurthy. The maximal expected covering location problem: Revisited. *Transportation Science*, 23(4):277–287, NOV 1989.
- [11] R. Batta and N. R. Mannur. Covering-location models for emergency situations that require multiple response units. *Management Science*, 36(1):16–23, JAN 1990.
- [12] A. Baykasoglu and N. N. Z. Gindy. A simulated annealing algorithm for dynamic layout problem. *Computers and Operations Research*, 28(14):1403–1426, DEC 2001.
- [13] O. Berman. Locating a facility on a congested network with random lengths. *Networks*, 15(3):275–293, 1985.
- [14] O. Berman and D. Krass. The generalized maximal covering location problem. *Computers and Operations Research*, 29:563–581, 2002.
- [15] O. Berman and B. LeBlanc. Location-relocation of mobile facilities on a stochastic network. *Transportation Science*, 18(4):315–330, NOV 1984.
- [16] O. Berman and A. R. Odoni. Locating mobile servers on a network with markovian properties. *Networks*, 12:73–86, 1982.
- [17] O. Berman and M. R. Rahnema. Optimal location-relocation decisions on stochastic networks. *Transportation Science*, 19(3):203–221, AUG 1985.
- [18] L. Brotcorne, G. Laporte, and F. Semet. Fast heuristics for large scale covering-location problems. *Computers and Operations Research*, 29:651–665, 2002.

- [19] Y. M. Carson and R. Batta. Locating an ambulance on the amherst campus of the state university of new york at buffalo. *Interfaces*, 20(5):43–49, SEP-OCT 1990.
- [20] S. C. Chapman and J. A. White. Probabilistic formulations of emergency service facilities location problems. *ORSA/TIMS Conference, San Juan, Puerto Rico*, 1974.
- [21] M. Chatterjee, S. K. Das, and D. Turgut. A weight based distributed clustering algorithm for mobile ad hoc networks. *Proceedings, Seventh International Conference on High Performance Computing (HiPC)'00*, 2000.
- [22] S. S. Chaudhry, I. D. Moon, and S. T. McCormick. Conditional covering: Greedy heuristics and computational results. *Computers and Operations Research*, 14(1):11–18, 1987.
- [23] W. Chen, N. Jain, and S. Singh. Anmp: Ad hoc network management protocol. *IEEE on Selected Areas in Communications*, 1999.
- [24] C.-C. Chiang. Routing in clustered multihop, mobile wireless networks with fading channel. *Proc. IEEE SICON'97*, 1997.
- [25] R. Church, J. Current, and J. Storbeck. A bicriterion maximal covering location formulation which considers the satisfaction of uncovered demand. *Decision Science*, 22(1):38–52, WIN 1991.
- [26] R. Church and C. ReVelle. The maximal covering location problem. *Papers Regional Science Association*, 32:101–118, 1974.
- [27] D. G. Conway and M. A. Venkataramanan. Genetic search and the dynamic facility layout problem. *Computers and Operations Research*, 21(8):955–960, OCT 1994.



- [28] J. Current, S. Ratick, and C. Revelle. Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *European Journal of Operational Research*, 110(3):597–609, NOV 1998.
- [29] M. S. Daskin. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, FEB 1983.
- [30] M. Gerla and J. T.-C. Tsai. Multiclustet, mobile, multimedia radio network. *ACM-Baltzer Journal of Wireless Networks*, 1995.
- [31] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. *Proceedings of IEEE ICUPC'97*, 1997.
- [32] Z. J. Haas and S. Tabrizi. On some challenges and design choices in ad hoc communications. *Proceedings of IEEE MILCOM'98*, 1998.
- [33] S. L. Hakimi. Optimum location of switching centers and the absolute centers and the medians of a graph. *Operations Research*, 12:450–459, 1964.
- [34] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13:462–475, 1965.
- [35] K. Hogan and C. Revelle. Concepts and applications of backup coverage. *Management Science*, 32(11):1434–1444, NOV 1986.
- [36] D. B. Johnson and D. A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. *Mobile Computing*, 1999.
- [37] K. Kasera and R. Ramanathan. A location management protocol for hierarchically organized multihop mobile wireless networks. *Proceedings of IEEE ICUPC'97*, 1997.
- [38] J. S. Kochhar and S. S. Heragu. Facility layout design in a changing environment. *International Journal of Production Research*, 37(11):2429–2446, JUL 1999.

- [39] P. Kouvelis, A. A. Kurawarwala, and G. J. Gutierrez. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63(2):287–303, DEC 1992.
- [40] V. Marianov and C. Reville. The queuing probabilistic location set covering problem and some extensions. *Socio-Economic Planning Sciences*, 28(3):167–178, 1994.
- [41] V. Marianov and D. Serra. Probabilistic maximal covering location-allocation models for congested systems. *Journal of Regional Science*, 38(3):401–424, AUG 1998.
- [42] E. Melachrinoudis and M. E. Helander. A single facility location problem on tree with unreliable edges. *Networks*, 27:219–237, 1996.
- [43] P. B. Mirchandani and A. R. Odoni. Location of medians on stochastic networks. *Transportation Science*, 13(2):85–97, May 1979.
- [44] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, Inc., 1997.
- [45] B. Montreuil and A. Laforge. Dynamic layout design given a scenario tree of probable futures. *European Journal of Operational Research*, 63(2):271–286, DEC 1992.
- [46] I. D. Moon and S. Chaudhry. An analysis of network location problems with distance constraints. *Management Science*, 30:290–307, 1984.
- [47] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *ACM Mobile Networks and App. J. Special Issue on Routing in Mobile Communication Networks*, 1996.
- [48] S. H. Owen and M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, DEC 1998.

- [49] U. S. Palekar, R. Batta, R. M. Bosch, and S. Elhence. Modeling uncertainties in plant layout problems. *European Journal of Operational Research*, 63(2):347–359, DEC 1992.
- [50] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., 1982.
- [51] A. K. Parekh. Selecting routers in ad-hoc wireless networks. *Proceedings of the IEEE International Telecommunication Symposium*, 1994.
- [52] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *Proceedings of INFOCOM'97*, 1997.
- [53] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing for mobile computers. *Proceedings of ACM SIGCOMM'94*, 1994.
- [54] C. E. Perkins, E. M. Royer, and S. R. Das. Multicast operation of the ad hoc on-demand distance vector routing. *Proceedings of Mobicom'99*, 1999.
- [55] R. Ramanathan and M. Steenstrup. Hierarchically organized multihop mobile wireless networks for quality-of-service support. *Mobile Networks and Applications*, 1998.
- [56] C. Revelle, J. Schweitzer, and S. Snyder. The maximal conditional covering problem. *INFOR*, 34(2):77–91, 1996.
- [57] M. J. Rosenblatt. The dynamics of plant layout. *Management Science*, 32(1):76–86, JAN 1986.
- [58] R. Sanchez, J. Evans, and G. Minden. Networking on the battlefield: Challenges in highly dynamic multi-hop wireless networks. *Proceedings of IEEE MILCOM'99*, 1999.
- [59] D. A. Schilling, V. Jayaraman, and R. Barkhi. A review of covering problems in facility location. *Location Science*, 1(1):25–55, 1993.

- [60] M. B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16:955–961, 1968.
- [61] C.-K. Toh. A novel distributed routing protocol to support ad hoc mobile computing. *Proceedings, IEEE 15th Annual International Phoenix Conference on Computers and Communication*, 1996.
- [62] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19:1363–1373, 1971.
- [63] G. L. Vairaktarakis and P. Kouvelis. Incorporation dynamic aspects and uncertainty in 1-median location problems. *Naval Research Logistics*, 46(2):147–168, MAR 1999.
- [64] J. R. Weaver and R. L. Church. Computational procedures for location problems on stochastic networks. *Transportation Science*, 17(2):168–180, MAY 1983.
- [65] J. A. White and K. E. Case. On covering problems and the central facilities location problem. *Geographical Analysis*, 1974.
- [66] T. Yang and B. A. Peters. Flexible machine layout design for dynamic and uncertain production environments. *European Journal of Operational Research*, 108(1):49–64, JUL 1998.