Capacity and Non-steady state Generalizations to the Dynamic MEXCLP model for Distributed Sensing Networks

by

Nishant Mishra

Department of Industrial Engineering

University at Buffalo

State University of New York

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

Abstract

A distributed sensing network consists of more than one (spatially) separated sensors, each with possibly different characteristics and not all of them sensing the same environment. Due to their vast applicability, there has been a flurry of recent activity in the area of network design with respect to distributed sensing. Issues involved in the design of efficient networks include sensor mobility, reliability of links and capacity.

This work builds on the Dynamic Expected Coverage Model proposed earlier and incorporates the issue of bandwidth capacity in the model. A MILP formulation is proposed that includes first order preferential assignment with coverage and relocation of sensors. The solution methodology uses modifications of the greedy heuristics and the column generation scheme. It is found that the capacity constraint does change the structure of the problem requiring additional computational effort.

Additionally, some non-steady state generalizations of the Dynamic MEXCLP model are theoretically explored. It is shown that under some assumptions, the problem of optimally locating cluster heads under non-steady state conditions for maximum coverage can be reduced to the steady-state formulations.

Dedication

To my parents

Acknowledgement

I would like to express my sincere gratitude to my thesis advisors **Dr. Rajan Batta** and **Dr. Rakesh Nagi** for their constant support and guidance. I would also like to thank them for their understanding and patience throughout the course of this thesis.

I thank Dr. James Llinas and the CMIF Lab for their support, and Prof. Christopher Rump, my research advisor in the lab, for his support, guidance and faith in me.

I take this oppurtunity to thank the O.R. faculty of the Department of Industrial Engineering, in particular Dr. Batta, Dr. Nagi, Dr. Rump, Dr. Llinas, Dr. Moises Sudit, Dr. Wayne Bialas and Dr. Ram Akella for always keeping their doors open and for making my whole UB learning experience, a memorable one.

I would like to thank Mr. Dipesh Patel for sharing his work and for having those fruitful discussions which resulted in my better understanding of the topic.

I would like to thank all my friends in the University and outside, in particular, my room-mate Ashwin, for the support and understanding they extended to me.

Finally, I would like to thank my parents and my sister for their continued support and inspiration all along the course of this work.

Aug 2003 Nishant Mishra

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

1.1 Introduction to Network Design for Distributed Sensing and Fusion

A distributed sensing network consists of more than one (spatially) separated sensors, each with possibly different characteristics and not all of them sensing the same environment. Some examples of distributed sensing as illustrated in Fig. 1.1 [45] might be

1. Military sensor networks to detect enemy movements, for tracking of enemy targets (such as planes, missiles etc.) and for the detection of the presence of hazardous material (such as poison gases or radiation, explosions, etc.)

2. Environmental sensor networks (such as in plains or deserts or on mountains or ocean surfaces) to detect and monitor environmental changes.

3. Wireless traffic sensor networks to monitor vehicle traffic on a highway or in a congested part of a city.

4. Wireless surveillance sensor networks for providing security in a shopping mall, parking garage, or other facility.

5. Wireless parking lot sensor networks to determine which spots are occupied and which spots are free.

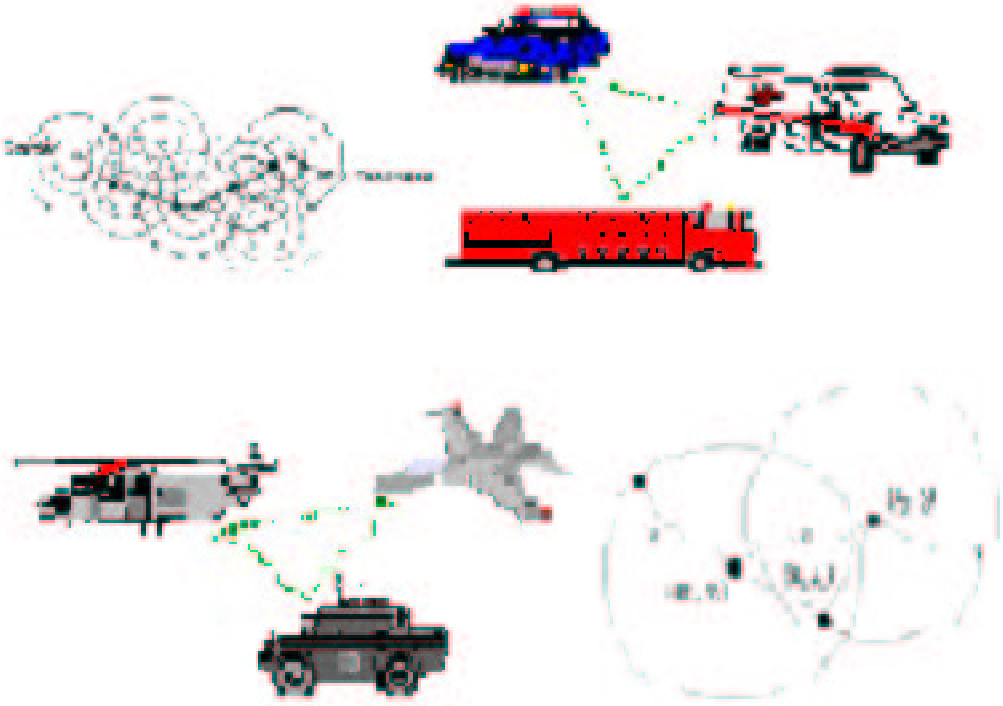


Figure 1.1: Some applications of Distributed Sensing and Fusion [45].

6. Other applications might include implementation of robotics, automated control of industrial systems (including monitoring of complex mechanical equipment such as turbo-machinery, helicopter gear trains or industrial manufacturing equipment), development of smart buildings and medical applications.

Besides offering certain capabilities and enhancements in operational efficiency in these conventional applications, distributed sensing can assist in the effort to increase alertness to potential terrorist threats [45].

Two ways to classify sensor networks are whether or not the nodes are individually addressable, and whether the data in the network is aggregated. The sensor nodes in a parking lot network should be individually addressable, so that one can determine the locations of all the free spaces. This application shows that it may be necessary to broadcast a message to all the nodes in the network. If one wants to determine the temperature in a corner of a room, then addressability may not be so important. Any node in the given region can respond. The ability of the sensor network to aggregate the data collected can greatly reduce the number of messages that need to be transmitted across the network. This idea of "data fusion" is formally defined below.

Data Fusion *is a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely estimates of situations and threats, and their significance [31].* The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results. The process of fusion over a distributed sensing environment might be *centralized* (a central fusion center which "fuses" the information from the sensors in the network) or *decentralized* (no single central fusion center).

These networks are usually constructed with one of these basic goals

- *Determine the value of some parameter at a given location* : In an environmental network, one might want to know the temperature, atmospheric pressure, amount of sunlight, and the relative humidity at a number of locations. This example shows that a given sensor node may be connected to a number of different types of sensors, each with a different sampling rate and range of allowed values.

- *Detect the occurrence of events of interest and estimate parameters of the detected event(s)* : In the traffic sensor network, one would like to detect a vehicle moving through an intersection and estimate the speed and direction of the vehicle.

- *Classify a detected object* : Is a vehicle in a traffic sensor network a car, a mini-van, a light truck, a bus, etc.

- *Track an object* : In a military sensor network, track an enemy tank as it moves through the network.

The *Distributed Sensing Network Design Problem* is the problem of designing a distributed sensing network with the (opposing) objectives of minimizing cost and maximizing the performance (or efficiency or effectiveness) of the network. The problem, due to its general nature and wide applicability might involve quite a few considerations such as the required data be disseminated to the proper end users. In some cases, there are fairly strict time requirements on this communication. For example, the detection of an intruder in a surveillance network should be immediately communicated to the police so that action can be taken. A general setting for this problem will include, besides other considerations, the following

- *Scalability* : The networks might comprise of as many as 100,000 nodes and so scalability is an important issue.

- *Sensor mobility/self-organization* : The sensors can be in motion (such as sensors on ocean surfaces or in military applications), the sensor location at any instant of time being known deterministically, probabilistically or might even be unknown. Given the large number of nodes and their potential placement in hostile locations, it is essential that the network be able to self-organize; manual configuration is not feasible.

- *Connectivity* : In general, the initial establishment of connection between a pair of sensors does not guarantee communication between them at all instances during the time horizon under consideration. This is because there might be link failures. These link failures are meant to model effects such as instrument malfunctions, lack of energy, terrain effects and in adversarial environments, the effect of jamming.

- *Energy Requirements* : The sensors might be so placed that in general they may not be always accessible, thus, making the energy required to keep the sensor functional, an important criterion.

- *Time Considerations* : As mentioned above, some applications might have restrictions on the maximum "delay" that might be allowed between the sensing and corresponding end communication of information.

- *Collaborative signal processing* : Yet another factor that distinguishes these networks from MANETs is that the end goal is detection/estimation of some events of interest, and not just communications. To improve the detection performance, it is often quite useful to fuse data from multiple sensors. This data fusion requires the transmission of data and control messages, and so it may put constraints on the network architecture.

- *Querying ability* : A user may want to query an individual node or a group of nodes for information collected in the region. Depending on the amount of data fusion performed, it may not be feasible to transmit a large amount of the data across the network. Instead, various local cluster heads will collect the data from a given area and create summary messages. A query will be directed to the cluster head nearest to the desired location.

An example of the sensor types and system architecture: With the coming availability of low cost, short range radios along with advances in wireless networking, it is expected that smart sensor networks will become commonly deployed. In these networks, each node will be equipped with a variety of sensors, such as acoustic, seismic, infrared, still/motion videocamera, etc. These nodes may be organized in clusters such that a locally occurring event can be detected by most of, if not all, the nodes in a cluster. Each node will have sufficient processing power to make a decision, and it will be able to broadcast this decision to the other nodes in the cluster. One node may act as the cluster master, and it may also contain a longer range radio using a protocol such as **IEEE 802.11** or **Bluetooth**.

Here, we consider the network design problem from a tracking viewpoint, and will be primarily interested in issues such as *sensor connectivity, sensor mobility, sensor communication (and data fusion) and jamming.*

1.2 Literature Review of Related Topics

Almost all the issues that we would be concerned with are characteristic of the *Wireless Ad Hoc networks*, an important sub-class of the Distributed Sensing Networks, and the corresponding literature review is presented below.

1.3 Literature review on Wireless Ad hoc Networks

Ad hoc networking is now an important part of mobile communication and 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 [50].

Sanchez *et. al.* [56] discuss some of the issues involved in these networks and broadly classified them as

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

These topics are discussed briefly here and a more comprehensive review could be found in Patel '03 [50].

1.3.1 Network Topology

Typically, the topology in an *ad hoc* network is either flat or hierarchical.

In a hierarchical topology, nodes are partitioned into groups called **clusters**, with one node chosen to perform the function of a **cluster head**. Depending on the number of hierarchies, such networks can have one or more *tiers* or *levels*. The cluster head is responsible for keeping track of locations of the nodes (also referred to as *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 needs to go hierarchically "upward") or to a neighboring node in the same cluster (if

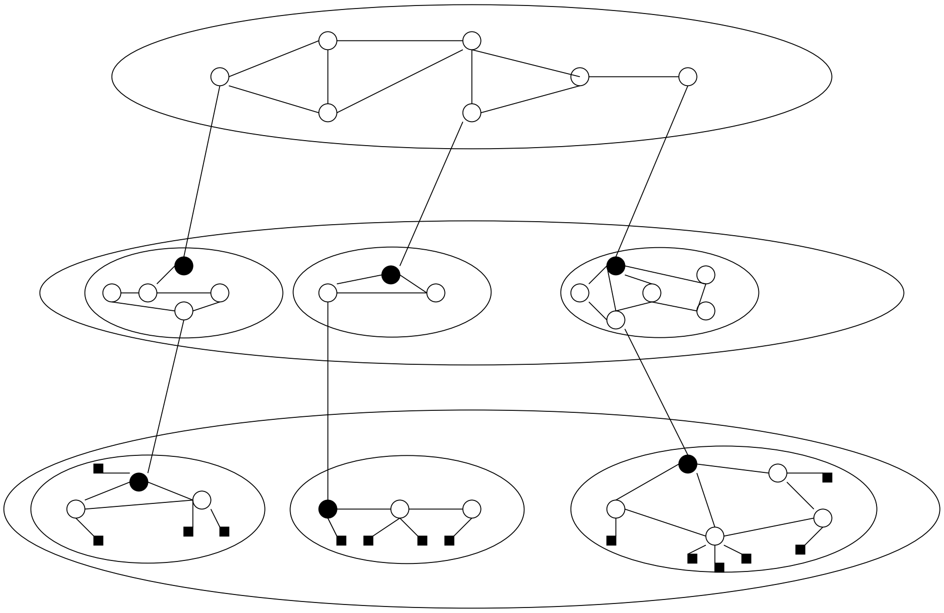


Figure 1.2: 3-tier Hierarchical Network Architecture

the message is to remain in the same level or needs to be sent ”downward”). Ramanathan and Steenstrup [53] present 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. 1.2.

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. 1.3 shows a diagram of a flat network.

Haas and Tabrizi [28] present a comparison between hierarchical and flat networks. The authors favor the flat architecture citing 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

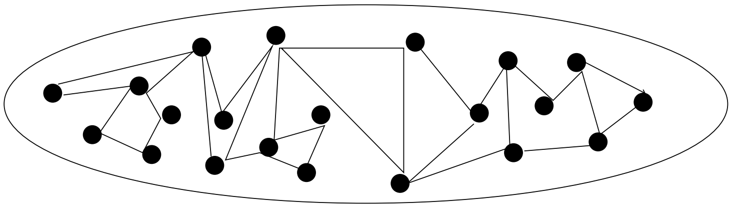


Figure 1.3: Flat Network Architecture

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 a ratio 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, a major disadvantage of flat networks is that they 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.

1.3.2 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 [34] consider 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 [34, 53]:

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.

Their approach doesn't seem to work in a flat network, however, since there is no equivalent to cluster heads.

However, in [28], Haas and Tabrizi have made a case for flat networks with zone routing as described in the next section.

1.3.3 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.

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) [51].

2. Clusterhead Gateway Switch Routing (CSGR) [21].

3. Wireless Routing Protocol (WRP) [44].

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) [52].

2. Dynamic Source Routing (DSR) [33].

3. Associativity Based Routing (ABR) [58].

4. Temporally Ordered Routing Algorithm (TORA) [49].

Hybrid protocols combine the advantages of both reactive and proactive protocols. Haas [27] 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.

1.3.4 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. A large variety of algorithms have been proposed in the literature for clustering in *ad hoc* networks, common features considered by these clustering algorithms being:

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 over 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 [6], 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 the one in [48]. 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 [26] and all nodes within its transmission range form the cluster. The process continues until all nodes have a designated clusterhead. Performance is better than highest-degree heuristic [21]. Excessive drainage of lower id nodes is found.

- **Node Weight Heursitic:** In this heuristic Basagni [5] assign 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. the number of clusterhead reassignment is small. No other features are captured in the heuristic.

- **Weight Based Clustering Algorithm:** Chatterjee et al. [18] propose a weight-based clustering algorithm where the weight of each node is updated periodically. Here the authors assign weight to each of the features 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. [7] 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 of their neighbors and each of the node finds it's relative mobility metric by a formula using the signal strength of the received message. Each node shares it's mobility metric with other nodes in it's transmission range and the node with least mobility metric assumes the position of clusterhead. The results in the paper show 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 (relative) velocity of the nodes and uses this 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 [20], 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.

1.3.5 Coverage Issues

One of the fundamental issues that arises in wireless *ad hoc* networks, in addition to location calculation, tracking, and deployment, is coverage. Due to the large variety of sensors and applications, coverage is subject to a wide range of interpretations. In general, coverage can be considered as the measure of *quality of service* of a sensor network. For example, in a fire detection sensor network, one may ask how well the network can observe a given area and

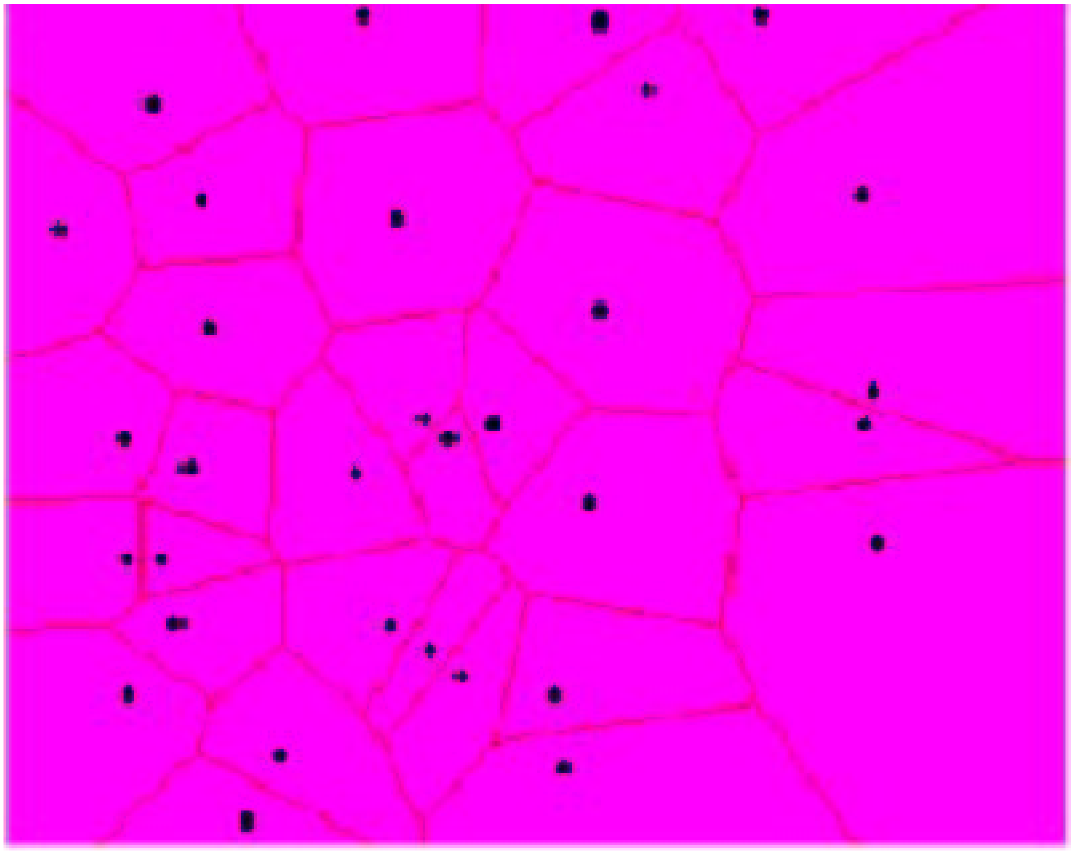


Figure 1.4: Voronoi Diagram of a Set of Randomly Placed Points in a Plane [39].

what the chances are that a fire starting in a specific location will be detected in a given time frame [39].

Meguerdichian *et. al.* [39], combined computational geometry techniques and constructs such as the *Voronoi Diagrams* with graph theoretical algorithmic techniques to propose a provably optimal polynomial time algorithm for coverage in sensor networks. The use of Voronoi diagram efficiently and without loss of optimality, transforms the continuous geometric problem into a discrete graph problem. In $2D$, the Voronoi Diagram of a set of discrete sites (points) partitions the plane into a set of convex polygons such that all points inside a polygon are closest to only one site. This construction effectively produces polygons with edges that are equidistant from neighbouring sites (Fig. 1.4).

It should be pointed out here that the coverage problems discussed in this section could be interpreted to refer to the coverage of terrain or system which needs to be “covered”, which is different from the coverage of sensors that we will be discussing in the subsequent chapters. The primary difference being that the former is generally a continuous spacial and temporal problem while the latter is a discrete one. Thus, the literature review on the Covering Models, discussed in the next section is more from an Operations Research perspective.

1.4 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 and thus their relevance to the sensor coverage problem on hand. A brief account is presented here and a more comprehensive review can be found in [50].

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

1. Deterministic Covering Models
2. Stochastic Covering Models

1.4.1 Deterministic Covering Models

Models in which all input parameters are deterministic are considered here.

The earliest work in the area of covering is dealt by Toregas et al. [59]. They model the location of emergency service facilities as a set covering location problem with equal costs in the objective function. 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 Revelle [23] 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).*

In order to overcome the uncertainity 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 Revelle in [32]. 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.

In the paper [10], 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 Revelle [32]. 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.

Church et al. [22] 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 [43] 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. [19] proposed seven greedy heuristics for solving the CCP.

Revelle et al. in [54] extend the notion of CCP [43] through the *Maximal Conditional Covering Problem* (MCCP I and MCCP II) and the *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. 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. 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 [12] 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. The GMCLP is shown to be equivalent to the Uncapacitated Facility Location Problem (UFLP).

1.4.2 Stochastic Covering Models

Models having one or more input parameters that are stochastic in nature are considered here.

Chapman and White [17] were the first to consider the unavailability of the facilities covering the region. They came up with the probabilistic version of SCLP. The model ensures that the probability of a demand node being served by at least one facility is greater than or equal to a specified reliability level α.

Later Daskin [25] 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 the *Maximum Expected Covering Location Problem* (MEXCLP), introduced by Daskin [25], 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. [9] relax 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.

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. [3] 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 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.

Marianov and Revelle in [37] 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 α. 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 [38] model the probabilistic MCLP with a restriction on the maximum waiting time for any demand or the maximum queue length. They model the problem as a 0-1 integer program and heuristic solutions are also presented.

Melachrinoudis and Helander in [40] consider the problem of locating a single facility on an undirected tree with n nodes in the presence of unreliable edges. 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. The authors also present two polynomial time algorithms for this problem which are possible because of the uniqueness of paths in trees and also because the problem is restricted to locating a single facility. They also

A more detailed review on covering literature can also be found in [46, 57, 61].

1.5 Literature Review of Dynamic Models

The problem to be considered is temporal in nature and correspondingly, we present a literature review of some dynamic models, in particular, *The Dynamic Plant Layout Problem* and *The Stochastic P-Median Problem.* Again a much more comprehensive review can be found in [50].

1.5.1 Dynamic 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.

Rosenblatt [55] 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.

Batta [8] establishes a class of best possible upper bounds for the DPLP described in [55]. 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 flow.

Balakrishnan et al. [2] 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.

Montreuil and Laforge in [42] 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. The authors formulate the problem as a linear program.

Yang and Peters [62] 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 [35] provide a genetic algorithm based heuristic for the dynamic plant layout problem for two consectutive 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 [11] solves the DPLP problem using a simulated annealing based procedure.

1.5.2 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 [29, 30], who that one of the optimal solutions of the p-median problem consists of locating the facilities only on the nodes of the network.

Mirchandani and Odoni [41] extended the p-median problem for networks where link traversal times 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 [29] 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).

Berman and Odoni [14] 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 present 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.

Later Berman and LeBlanc [13] developed a polynomial time heuristic for solving the stochastic p-median problem with relocation of facilities allowed with change in the state of the network.

Later Berman and Rahnama [15] extend the work of Berman and Odoni [14] with the only modification that the location of the p facilities depends on the previous state unlike assumption 4 (mentioned previously) of [14].

Carson and Batta [16] 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 [60] 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.

Averbakh and Berman [1] consider the 1-median problem with the uncertainty involved in the weights (demand) of the nodes of the network.

Current et al. [24] approach the dynamic facility location problem with uncertainty involved in the total number of facilities to be located.

1.6 Research Objectives and Thesis Organization

The research objective is to capture spacial and temporal dynamics of sensor movement and information flow in a distributed sensing framework. These can be enumerated as:

1. To propose a mathematical model for optimally locating cluster heads in a **capacitated** distributed sensing network.

2. To capture mobility of sensors and link failures due to *reliability* issues or *jamming.*

3. To propose solutions that are *effective in real time.*

This thesis, thus, intends to generalize some of the earlier work by Dipesh Patel [50].

1.6.1 Thesis Organization

Chapter 2 is comprised of the detailed description of the problem and the proposed *Mixed Integer Linear Program* MILP formulations, the two formulations differing in the way the capacity constraint is modelled. Chapter 3 presents the proposed solution methodologies. Two greedy heuristics are proposed and a Column Generation Heuristic is proposed to further improve their solutions. Numerical studies carried out for these solution methodologies are discussed in Chapter 4. Chapter 5 presents some theoretical generalizations under *non-steady state* conditions and also, under the effect of *jamming.* Finally, conclusions drawn from the research and possible avenues for future research are discussed in Chapter 6.

Chapter 2

Problem Formulation

The systems that are modelled in this research are primarily Defense related. However, the approach (in most cases) and the results (in some cases) could be extended to most of the Distributed Sensing systems illustrated in Section 1.1.

2.1 Description of the Problem

The scenario under consideration is that of a *battlefield* - land, sea, air or a combination of these. The sensors could be radars, soldiers, tanks, etc. (on the ground), sonars on ships, submarines (in water) or unmanned aircraft vehicles (UAVs), other airborne surveillance systems (in air). These entities are some of the sources of information that could be used to recognize threats to the system and thus aid in the task of gathering intelligence. The information from these various sources is then assimilated through the process of *Data Fusion*, defined in Section1.1.

Centralized data fusion, in which data is fused at a single processor can be carried out by Command and Control Centers or even on a smaller scale by satellites etc. Currently, however, the emphasis is shifting towards Decentralized data fusion, in which data from groups of sensors (reffered to as *clusters*) are processed together at their corresponding fusion centers (reffered to as *cluster heads*) and then broadcast or transmitted to other cluster heads. Incoming data for fusion at cluster heads might thus be *pre-fused* in some sense (though not always). This abstraction is represented in Fig.2.1, where the upper plane is that of the cluster heads and the lower one that of sensors.

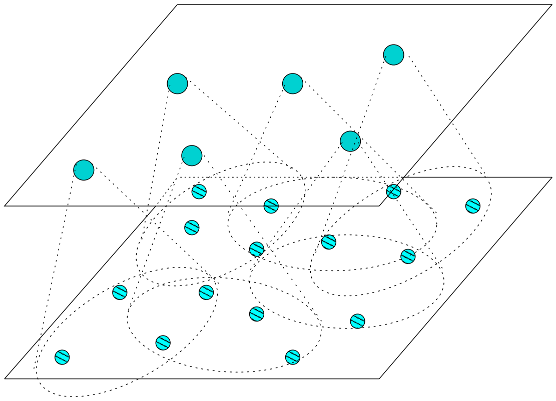


Figure 2.1: Abstract representation - Cluster heads and sensors [50].

In a military setting, such cluster heads are suitably represented (for the purpose of modelling) by Airborne Warning and Control Systems (AWACS), though it should be noted that they could as easily be a tank in a regiment or the captain of an infantry unit.

The sensors are in general spatially separated and may not be in close proximity of one another. Hence a pertinent question is: where should the data fusion take place? In other words, **how should the cluster heads be placed so as to maximize the information obtained (fused) from the sensors**? This is the primary question that this research aims to address. Realistic scenarios would involve sensors and/or cluster heads that could be 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. Under hostile conditions other important considerations might be *jamming* and/or *exposure* (the latter is not addressed as part of this work).

The fusion is typically carried out by AWACS which are aircrafts that possess fusion processors to carry out the data fusion process. The objective of the work is to locate these AWACS to maximize data gathered from the sensors.

In this work it has been assumed that the AWACS are located at discrete points. This may not be a very strong assumption since, as discussed in Section 1.3.5, continuous space might be discretized using the Voronoi diagrams. Even on a more simplistic scale, the space in which the AWACS are located could be broken down into *grids* (depending on the *time period* and speed of the AWACS) with the center of the grids being labelled as the point of occupancy (during that time period).

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.*

The data transfer takes place through wireless communication. Typical issues related to wireless communication are transmission range, available bandwidth, etc. Since the sensors are mobile they may move out of the range of the cluster heads 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 failure (which might include *instrument malfunction, foliage effects and terrain effects*). Thus it is extremely desirable that each sensor should have *multiple coverage*, i.e. each sensor should be covered by more than one cluster head. 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.

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 the sensor data. However to ensure a stable location strategy and for practical implementation purposes, we either restrict the maximum number of relocations for the entire "time horizon" or associate a cost to every relocation of an AWACS (termed as *relocation cost*). Thus we consider a trade-off between data coverage and relocation cost. Since the objective has two terms (data obtained and relocation cost), which may not share the same units, one might think of the relocation cost as the amount of information lost in the process of relocation. 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.

2.2 Mathematical Approach

As is quite evident, these problems are suited to graphical representation and graph theoretic approaches, with nodes representing the sensors (having various properties) and arcs between the sensors representing the feasibility of communication between the corresponding sensors (i.e., links). Each arc might have a cost (representing the cost of constructing and maintaining the connection) and a capacity (representing, for instance, the bandwidth) associated with it.

Given the general problem, we start with simpler versions of the problem and build models to incorporate more and more features. Consider, initially, the problem of optimally locating cluster heads with respect to sensors in a network. The cluster heads are assumed to only assimilate the information sensed by the sensors. The sensors do not communicate with one another and neither do the cluster heads. Thus, the sensors and the cluster heads are functionally different and this gives a special structure to the network, that of a bipartite graph (Fig. 2.2). Each cluster head is said to "cover" a sensor at a given time instant if it can communicate with it at that instant. Under steady-state conditions, the communication between the sensor and cluster head is assumed to be reliable with a fixed (known) probability. The cluster heads are to be located so as to maximize the reliability of the network. The sensors are all assumed to be identical and so are the cluster heads. The sensors and the cluster heads are both assumed to be mobile. Thus, given sensor locations at all instants of time, we need to come up with a sequence of optimal cluster head positions for the different time periods over a finite horizon. It is assumed that the possible cluster head locations might be discretised (to avoid non-linear formulations which arise in cluster head placements over a continuous state space), and since the sensor locations are known, the "coverage" of sensors with respect to possible cluster head locations is known and is essentially captured by a constant. Also, a possible sensor location is assumed to cover a given set of sensors which does not change during a given time period. This optimal cluster

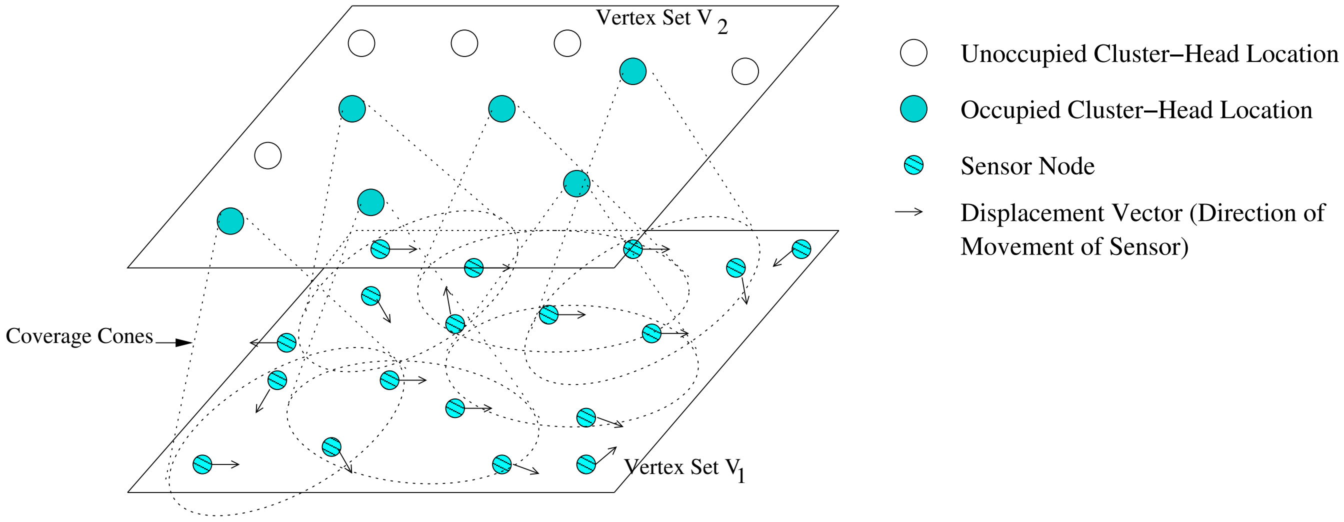

Vertex Set V 2
Unoccupied Cluster−Head Location
Occupied Cluster−Head Location
Sensor Node
Displacement Vector (Direction of Movement of Sensor)
Coverage Cones
Vertex Set V1

Figure 2.2: A Schematic showing cluster-heads and sensors [50].

head location problem has been attempted [50] and dealt with in the literature.

Called the *Dynamic MEXCLP model* [50], this is a generalized version of the *MEXCLP model* proposed by Daskin in [25]. A brief review of the MEXCLP model is presented here.

2.2.1 MEXCLP

The objective of MEXCLP is to maximize the expected demand covered by locating a given number of facilities. 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 MEXCLP [25] is formulated as follows:

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

subject to

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

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

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

$$y_{jk} \quad \in \quad \{0,1\} \qquad\qquad \forall \quad j=1,\ldots,n, k=1,\ldots,N. \qquad (2.4)$$

where the variables and their corresponding indices are defined as follows [25]:

i = index for potential facility locations, $i = 1, \ldots, N$,

k = index for demand nodes, $k = 1, \ldots, N$,

N = number of demand nodes,

n = number of facilities to be located,

D = the distance beyond which a demand node is considered "uncovered",

D_{ik} = distance between potential facility location i and demand node k,

d_k = demand of node k,

p = probability of a facility failure $(0 < p < 1)$,

$$r_{ik} = \begin{cases} 1, \text{ if } D_{ik} < D \\ 0 \text{ otherwise.} \end{cases}$$

The decision variables of the problem are:

x_i = number of facilities placed at location i,

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

From [25, 50], 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 $y_{1k}, y_{2k}, \ldots, y_{mk}$ 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.

In his work, Patel proposed several different formulations for the Dynamic MEXCLP model. We enumerate these and present a review of the main model proposed [50]:

1. Dynamic MEXCLP *with Relocation.*

2. Dynamic MEXCLP *with restriction on the maximum number of relocations for the entire time horizon.*

3. Dynamic MEXCLP *with location dependent relocation cost.*

4. Dynamic MEXCLP *for Clustering with Mobile Facilities.* In this variant of the Dynamic MEXCLP, the assumption that the potential location of clusterheads (facilities) are different from the location of the sensors is relaxed. Thus, instead of locating facilities at potential locations, sensors are chosen that act as the cluster head (fusion node) for that time period. Since the sensors are mobile, the elected clusterhead is itself a sensor. Thus cluster heads are also mobile and follow the same trajectory as they followed when they were sensors.

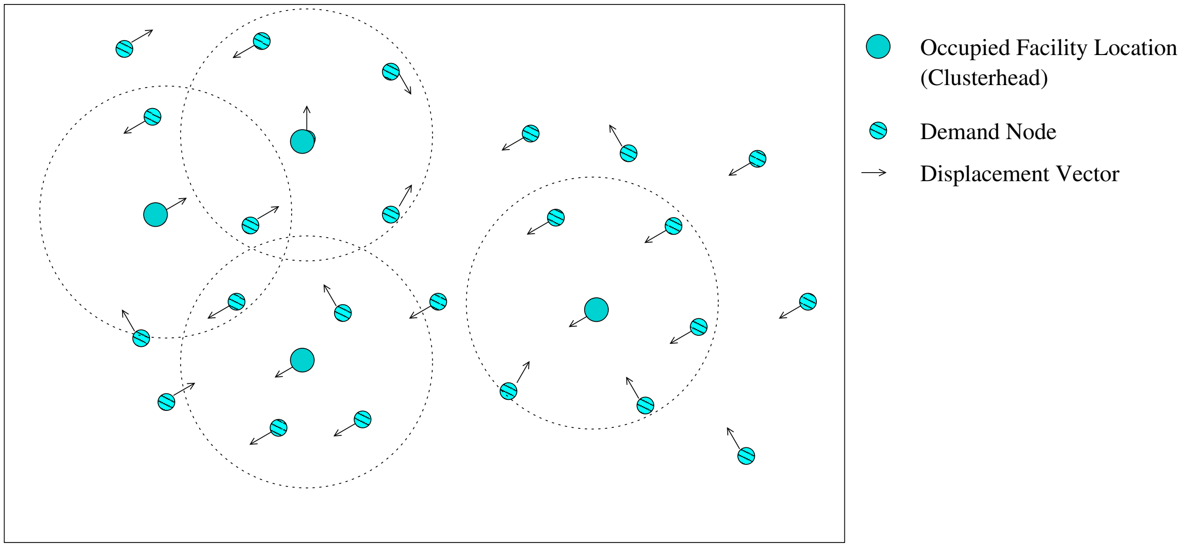


Figure 2.3: A snapshot of the network [50].

The formulation is similar to the one described in Section 2.2.2 with the only difference being that the set of potential facility location is same as that of the set of sensors in this case. It should be noted that it is assumed that the performance of an entity as a sensor is not affected when it is chosen to play the role of a cluster head.

In the Dynamic MEXCLP model, it is assumed that the facilities are perfectly reliable. The demand nodes are assumed mobile, but their velocity vectors are assumed to be known and hence there is an *a priori* knowledge about the exact location of demand nodes at any given time. 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.

The following assumptions are made in the 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.

2.2.2 Dynamic MEXCLP with Relocation [50]

Due to the mobility of the sensors, 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 the facilities are allowed to change location with time at an additional cost.

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:

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

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

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

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

$$x_{it} \in \{0,1\} \quad \forall \quad i = 1,\ldots,|\Delta|, t = 0,\ldots,T, \tag{2.9}$$

$$w_{it} \geq 0 \quad \forall \quad i = 1,\ldots,|\Delta|, t = 1,\ldots,T, \tag{2.10}$$

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

where the variables and their indices are defined as:

Δ = set of potential facility locations.

Θ = 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} = positive difference in the number of facilities at location i between time $t-1$ and time t

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 $y_{1kt}, y_{2kt}, \ldots, y_{mkt}$ 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 an 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. Patel made 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:

Even though the variables y_{jkt} are continuous variables between 0 and 1, at optimality y_{jkt} assumes binary values.

Patel [50] modelled the problem as a covering location problem with the objective of maximizing the expected demand covered by locating a given number of cluster heads. However, even though modelling it as a communications network, one important aspect was not considered - *bandwidth.* The importance of capacity in communication networks, specially for large scale distributed sensing networks cannot be over-emphasized. This research builds on the Dynamic MEXCLP model by incorporating capacity constraints.

2.3 Capacitated Dynamic MEXCLP model

The Dynamic MEXCLP model was proposed under an important assumption - that of unlimited capacity. This assumption though simplifying, was quite unrealistic. The *Capacitated Dynamic MEXCLP model* incorporates the capacity constraint and thus, is more realistic in terms of capturing the dynamics of the distributed sensing system. Assumptions (explicitly enumerated below) regarding sensor and cluster behaviours remain the same otherwise. Cluster heads are assumed completely reliable while all links have a (identical) steady-state probability of failure. Sensors are mobile with *known* velocity vectors. The time horizon is divided into equal time periods and relocation of cluster heads takes place at the beginning of each time period.

Assumptions made while building the model:

1. Location of the sensors are known at any given instance.

2. Potential cluster head locations constitutes a set of discrete points.

3. Relocation of facilities takes place at discrete time periods.

4. The cluster heads are perfectly reliable.

5. The probability of failure of all links is the same.

6. Cluster heads are identical in all respects.

7. Transmission time is negligible and data is not transmitted in packets.

As can be seen, all the assumptions are the same as in the Dynamic MEXCLP model except the last two which are essential to model the capacity constraint. The last but one is required since a sensor is preferentially assigned to a *cluster head location* and not to a cluster head. The last one is required to avoid consideration of queueing effects.

In order to incorporate the capacity constraint, each sensor needs to be assigned to one of the cluster heads that is “covering” it. However, we still have link failure probabilities and correspondingly, need multiple coverage of sensors. Thus multiple coverage still remains while preferential assignment of sensor to cluster head is incorporated. Once again (Fig. 2.4), it can be seen that due to the mobility of sensors, the optimal location of cluster heads for one time period may not be optimal for the entire horizon. The trade off now is between multiple coverage and preferential assignment on one side and relocation cost on the other.

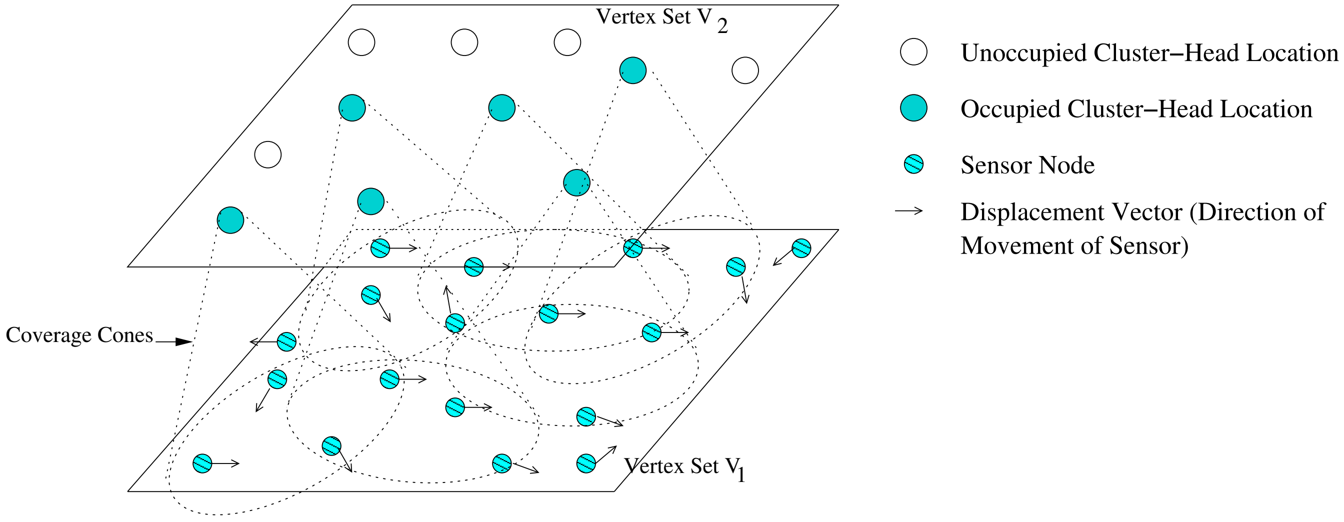

Vertex Set V 2
Unoccupied Cluster–Head Location
Occupied Cluster–Head Location
Sensor Node
Displacement Vector (Direction of Movement of Sensor)
Coverage Cones
Vertex Set V1

Figure 2.4: A snapshot of the network

Two different formulations incorporating capacity restrictions are proposed. The formulations differ in the way the capacity constraint is applied. In Formulation 1, the capacity constraint is applied for each time period, while in Formulation 2, it is applied for the entire time horizon. Thus, the second formulation is a relaxed version of the first.

The Capacitated Dynamic MEXCLP with relocation for the *operational capacity* case is formulated as follows:

<u>Formulation</u> 1

$$\text{Maximize} \quad \sum_{t=0}^{T}\sum_{k\in\Theta}\sum_{j=1}^{n}(1-p)p^{j-1}d_k y_{jkt} + \sum_{i\in\Delta}\sum_{k\in\Theta} c_{ik}z_{ikt} - \sum_{t=1}^{T}\sum_{i\in\Delta} Cw_{it},$$

subject to

$$\sum_{j=1}^{n} y_{jkt} - \sum_{i\in\Delta} r_{ikt}x_{it} \;\leq\; 0 \qquad \forall \quad k=1,\ldots,|\Theta|, t=0,\ldots,T, \tag{2.12}$$

$$\sum_{i \in \Delta} x_{it} \leq n \quad \forall \quad t = 0, \ldots, T, \tag{2.13}$$

$$w_{it} \geq x_{it-1} - x_{it} \quad \forall \quad i = 1, \ldots, |\Delta|, t = 1, \ldots, T, \tag{2.14}$$

$$w_{it} \geq x_{it} - x_{it-1} \quad \forall \quad i = 1, \ldots, |\Delta|, t = 1, \ldots, T, \tag{2.15}$$

$$z_{ikt} \leq r_{ikt} x_{it} \quad \forall \quad i = 1, \ldots, |\Delta|, k = 1, \ldots, |\Theta|, t = 0, \ldots, T, \tag{2.16}$$

$$\sum_{i \in \Delta} z_{ikt} \leq 1 \quad \forall \quad k = 1, \ldots, |\Theta|, t = 0, \ldots, T, \tag{2.17}$$

$$\sum_{k \in \Theta} d_k z_{ikt} \leq Q_{it} \quad \forall \quad i = 1, \ldots, |\Delta|, t = 0, \ldots, T, \tag{2.18}$$

$$x_{it} \in \{0, 1\} \quad \forall \quad i = 1, \ldots, |\Delta|, t = 0, \ldots, T, \tag{2.19}$$

$$w_{it} \geq 0 \quad \forall \quad i = 1, \ldots, |\Delta|, t = 1, \ldots, T, \tag{2.20}$$

$$y_{jkt} \leq 1 \quad \forall \quad j = 1, \ldots, n, k = 1, \ldots, |\Theta|, t = 0, \ldots, T, \tag{2.21}$$

$$z_{ikt} \in \{0, 1\} \quad \forall \quad i = 1, \ldots, |\Delta|, k = 1, \ldots, |\Theta|, t = 0, \ldots, T. \tag{2.22}$$

where the variables and their indices are defined as:

Δ = set of potential facility locations.

Θ = 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_{ik} = value of preference of assignment of sensor k to cluster head location i

Q_{it} = capacity of cluster head location i during time period t

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:

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

$$z_{ikt} = \begin{cases} 1, & \text{if sensor } k \text{ is assigned to cluster head } i \text{ during time period } t \\ 0 & \text{otherwise.} \end{cases}$$

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

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

The objective function has three terms as compared to the Dynamic MEXCLP with relocation. The first and last terms are the same, while the second term represents *preferred assignment* of sensor to cluster head location (Assumption 7). Thus the objective function tends to maximize demand covered and preferential assignment while allowing for relocation of cluster heads over a time horizon. Constraints 2.16 ensure that sensor k is assigned to cluster head location i only if location i is occupied by a cluster head and sensor k can be covered from location i. Constraints 2.17 ensure that a sensor is assigned to only one cluster head during a time period. Constraints 2.18 are the capacity constraints for each cluster head location for each time period. Constraints 2.22 force z_{ikt} to take a value of 0 or 1.

The formulation for the *planning capacity* case is:

Formulation 2

$$\text{Maximize} \quad \sum_{t=0}^{T}\sum_{k\in\Theta}\sum_{j=1}^{n}(1-p)p^{j-1}d_k y_{jkt} + \sum_{i\in\Delta}\sum_{k\in\Theta} c_{ik}z_{ikt} - \sum_{t=1}^{T}\sum_{i\in\Delta} Cw_{it},$$

subject to

$$\sum_{j=1}^{n} y_{jkt} - \sum_{i\in\Delta} r_{ikt}x_{it} \leq 0 \quad \forall \quad k = 1,\ldots,|\Theta|, t = 0,\ldots,T, \tag{2.23}$$

$$\sum_{i\in\Delta} x_{it} \leq n \quad \forall \quad t = 0,\ldots,T, \tag{2.24}$$

$$w_{it} \geq x_{it-1} - x_{it} \quad \forall \quad i = 1,\ldots,|\Delta|, t = 1,\ldots,T, \tag{2.25}$$

$$w_{it} \geq x_{it} - x_{it-1} \quad \forall \quad i = 1,\ldots,|\Delta|, t = 1,\ldots,T, \tag{2.26}$$

$$z_{ikt} \leq r_{ikt}x_{it} \quad \forall \quad i = 1,\ldots,|\Delta|, k = 1,\ldots,|\Theta|, t = 0,\ldots,T, \tag{2.27}$$

$$\sum_{i\in\Delta} z_{ikt} \le 1 \quad \forall \quad k = 1,\dots,|\Theta|, t = 0,\dots,T, \tag{2.28}$$

$$\sum_{t=1}^{T}\sum_{k\in\Theta} d_k z_{ikt} \le Q_i \quad \forall \quad i = 1,\dots,|\Delta|, \tag{2.29}$$

$$x_{it} \in \{0,1\} \quad \forall \quad i = 1,\dots,|\Delta|, t = 0,\dots,T, \tag{2.30}$$

$$w_{it} \ge 0 \quad \forall \quad i = 1,\dots,|\Delta|, t = 1,\dots,T, \tag{2.31}$$

$$y_{jkt} \le 1 \quad \forall \quad j = 1,\dots,n, k = 1,\dots,|\Theta|, t = 0,\dots,T, \tag{2.32}$$

$$z_{ikt} \in \{0,1\} \quad \forall \quad i = 1,\dots,|\Delta|, k = 1,\dots,|\Theta|, t = 0,\dots,T. \tag{2.33}$$

Here the variables and their indices remain the same as defined earlier except for Q_{it}, which is now replaced by Q_i standing for the capacity of cluster head location i over the entire time horizon (note again that all cluster heads are assumed identical).

A few remarks regarding the formulations:

1. Using arguments along the same lines as in [50], *one can prove that the formulation with regard to maximizing expected coverage remain the same as in the case of the MEXCLP.* Thus link failures or facility failures would both (independently) result in the same formulation with respect to the coverage.

2. **Observation 1:**

   In the formulation variables x_{it} are restricted to take only binary values. Constraints (2.14) and (2.15) 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. Thus, the same argument as in [50] holds.

   Observation 2:

   Even though the variables y_{jkt} are continuous variables between 0 and 1, at optimality y_{jkt} assumes binary values. This can be argued along the same lines as in [50].

3. z_{ikt} need not take $0-1$ values unless defined to do so.

4. *It might be noted here that the preferential assignment is an attempt to capture capacity restrictions and is not completely accurate. This is because it only considers first order*

link failures and does not take into consideration subsequent assignments. That is, if the link between a sensor and its most preferred cluster head location breaks down, subsequent link assignments are not accounted for.

2.4 Chapter Summary

The problem is described and motivated starting with the MEXCLP and the Dynamic MEXCLP models. Two different MILP formulations of the Capacitated Dynamic MEXCLP are proposed. It is observed that properties holding for the Dynamic MEXCLP model also hold true for the Capacitated Dynamic MEXCLP model. We also establish that the structure of the formulation remains unaltered if link failure is considered instead of facility failure. In Chapter 3, solution procedures proposed to solve the formulation are presented.

Chapter 3

Solution Strategies

The Dynamic MEXCLP is NP-hard [50] since for fixed values of w_{it} (relocation variable), it can be decomposed into $T + 1$ Maximal Covering Location Problems, each NP-hard by itself. Since the Dynamic MEXCLP is a special case of the Capacitated Dynamic MEXCLP (corresponding to infinite capacity), the Capacitated Dynamic MEXCLP also is NP-hard. Thus, in general, the Capacitated Dynamic MEXCLP can be expected to be computationally intensive and tough to solve.

An initial thought might be: *Why to develop special solution procedures and not just use a standard commercial solver like CPLEX?* The reasons are two fold:

1. With increasing problem size it becomes more and more difficult for commercial solvers and in fact, large problems (representative of the real world applications) cannot even be read by most solvers.

2. Commercial solvers employing standard Branch and Bound procedures may not use the special structure of the problem and thus perform inefficiently with respect to computational time. Since this problem has been developed with the aim of aiding real-time decision making (under hostile conditions), computational time taken by solution procedures is a critical issue. In the case when time is not of the essence (such as network design during peacetime scenarios), solution quality would be important.

The formulation for the capacitated dynamic MEXCLP is quite similar to that of the Dynamic MEXCLP, and correspondingly, we explore the solution procedures that are slight modifications of the ones proposed for the latter.

3.1 Solution Methodology: Heuristics

In [50], Patel proposed various heuristics for solving the Dynamic MEXCLP. They are:

1. Relocation Heuristic (RH), and
2. No-Relocation Heuristic (NRH).
3. Column Generation Heuristic (CGH).

Below, each of these are discussed in the context of the Capacitated Dynamic MEXCLP problem.

3.2 Modified Relocation Heuristic

The Modified Relocation Heuristic (MRH) is a greedy heuristic, which picks the best n locations, one at a time, for each time period. The RH algorithm is presented as follows:

1. Start at time period zero.
2. Calculate the sum of the demand covered and the preferred assignment for each potential location. The preferred assignment cannot exceed the capacity for that location.
3. Select the potential location with the maximum sum and place a facility at this location. This potential location is now precluded from future placement of a facility for this particular time period.
4. 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, update the demand nodes covered by this facility by multiplying them by p. Use the updated demand for future calculation of coverage for only this time slot.
5. Repeat steps $2 - 4$ until all the facilities are placed for the time period.

6. Repeat this procedure for all time periods. Relocation (between successive time periods t and $t+1$) only occurs if the net gain after relocation is positive (i.e., $coverage + assignment - relocation > 0$). In order to take care of the relocation cost, add $2C$ (since a change in number of facility at two locations constitutes a relocation of a facility) to the sum of coverage and assignment of the potential location for time $t+1$, if that location was selected in the previous time period t. Chose the maximum with this new total and repeat steps 3 and 4 for that time slot.

The heuristic takes into consideration the link failure probabilities as well as the relocation cost (but only between successive time periods).

Although the heuristic seems to perform reasonably well as discussed later in the results section, it can be proved that it *performs arbitrarily bad in the worst case.* This can be proved (along similar lines as in [50]) citing a particular example.

For this, we formally define the concept of an ϵ-approximate algorithm (Papadimitriou and Steiglitz [47] and Patel [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 ϵ-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.

Consider the following example:

For this example, 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 $= T + 1$

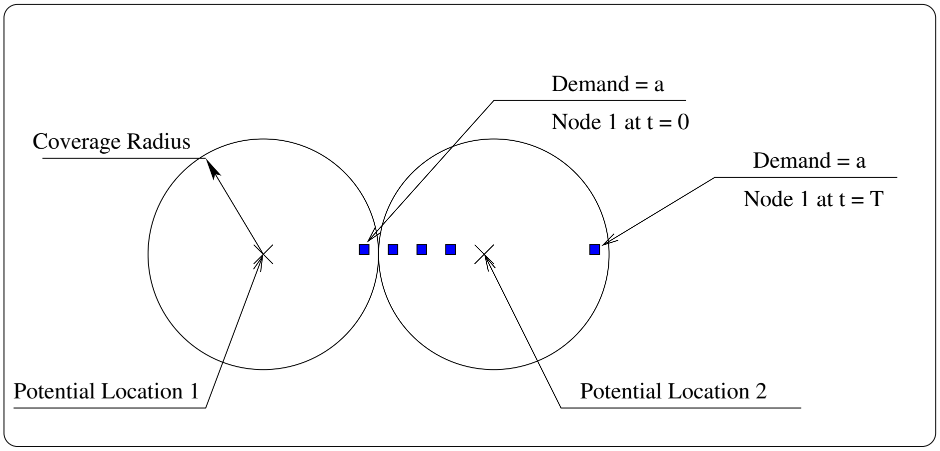


Figure 3.1: Worst case analysis of NRH [50]

Probability of link failure $= p$

Preference to location 1 $= c_1$

Preference to location 2 $= c_2$ $(a + c_1 > c_2 > c_1)$

Relocation cost $= b$ $(b > a + c_2)$

Capacity of each potential location $= a$

In this example, MRH selects potential location 1 for locating at time $t = 0$ with the sum of expected coverage and preferred assignment being $(1 - p) * (a) + c_1$ for that time period. For time period $t = 1$, potential location 2 has the maximum sum of expected coverage and assignment among all potential locations. Thus MRH computes the gain in coverage by switching to potential location 2 and its associated cost. Since $(b > (a + (c_2 - c_1)))$, 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 sum total of $coverage + assignment = (1 - p) * (a * 1) + c_1$. However, from observation, the optimal solution is to locate the facility at potential location 2 for all time periods with the total of $(1 - p) * (a * T) + c_2 * T$.

Thus we have

$c(\hat{f}(I)) = (1-p) * (a * T) + c_2 * T$ and

$c(f_B(I)) = (1-p) * (a * 1) + c_1$

Hence, for MRH to be an ϵ-approximate algorithm

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

$$\text{or, } \frac{|(1-p)*(a*1)+c_1-((1-p)*(a*T)+c_2*T)|}{(1-p)*(a*T)+c_2*T} \leq \epsilon$$

$$\text{or, } \frac{|\frac{c_1}{(1-p)a} - (T + \frac{c_2*T}{(1-p)a})|}{T + \frac{c_2*T}{(1-p)a}} \leq \epsilon$$

$$\text{or, } \frac{T + \frac{c_2*T-c_1}{(1-p)a}}{T + \frac{c_2*T}{(1-p)a}} \leq \epsilon$$

Thus, MRH is not ϵ-approximate and further, as $T \longrightarrow \infty$, $\frac{T + \frac{c_2*T-c_1}{(1-p)a}}{T + \frac{c_2*T}{(1-p)a}} \longrightarrow 1$. In other words, te error between the optimal solution and the the MRH solution $\longrightarrow 100\%$. Thus, the heuristic can perform arbitrarily bad in the worst case.

3.3 Modified No Relocation Heuristic

The Modified No-Relocation Heuristic (MNRH) 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 MNRH algorithm is as follows:

1. Calculate the sum of total demand and preference covered by each potential location for all time periods (noting that preference cannot exceed capacity in each time period).

2. Select the potential location with the maximum value 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 MRH earlier.

4. Repeat the above steps until all the facilities are placed.

This heuristic also takes care of link failure probabilities in a similar fashion as MRH does and is found to perform reasonably well for problem instances with high relocation costs. However, again it can be proven to give arbitrarily bad solutions in the worst case. 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 $= a$ $(a > b)$

Demand of node 2 $= b$ $((1 - p) * b * (T + 1) > a)$

Number of time slots $= T + 1$

Probability of link failure $= p$

Relocation cost $(C) = 0$

Preference of node 1 to all potential locations = Preference of node 2 to all potential locations $= c$

Capacity of each potential location $= a$

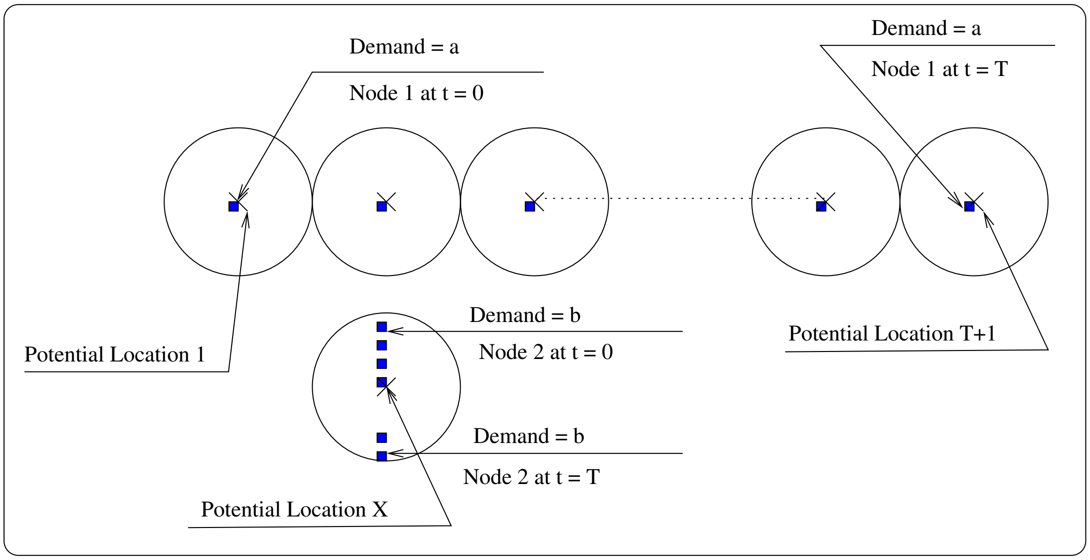


Figure 3.2: Worst case behaviour of MNRH [50]

Potential location X is selected by the heuristic for locating a facility for all time periods with the coverage value of $(1-p)*b*(T+1)+c*(T+1)$. 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. 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 $(1-p)*a*(T+1)+c*(T+1)$ (since relocation cost $=0$).

Thus, for MNRH to be an ϵ-approximate algorithm

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

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

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

$$\text{or, } \frac{a-b}{a} \leq \epsilon$$

$$\text{or, } 1 \text{ - } \frac{b}{a} \leq \epsilon$$

Thus, MNRH is not ϵ-approximate and further, as $a \longrightarrow \infty$ (and correspondingly $T \longrightarrow \infty$), 1 - $\frac{b}{a} \longrightarrow 1$. In other words, the error between the optimal solution and the MNRH solution $\longrightarrow 100\%$. Thus, the heuristic can perform arbitrarily bad in the worst case.

In a manner similar to the ones used for the two heuristics above, we can show that the best amongst the two solutions provided by the heuristics when they are both indpendently applied to the same problem instance can also be *arbitrarily bad.* It is again obtained using a suitable combination of paramenters.

3.4 Column Generation

The greedy heuristics, in the worst case, can give arbitrarily bad solutions. Their advantage lies in the fact that they are *fast*. However, where the *quality of the solution* is of importance and computation time is not, the solution provided by the heuristics may not be acceptable. Thus, a *Column Generation (CG)* heuristic is proposed to further refine the solution obtained from these greedy heuristics. This application of CG is similar to the one used by Patel [50].

Column generation (CG) heuristic is an efficient and widely used technique to solve large-scale integer programs (for more information please refer [4, 36]). A modified column generation approach is used here to solve the Capacitated Dynamic MEXCLP (refer [50] for details. The column generation formulation is obtained by decomposing the original MILP formulation *Formulation* 1 of the (Capacitated Dynamic MEXCLP with relocation cost) into a "Master Problem" and a "Sub-Problem". Their roles are explained in Fig. 3.3.

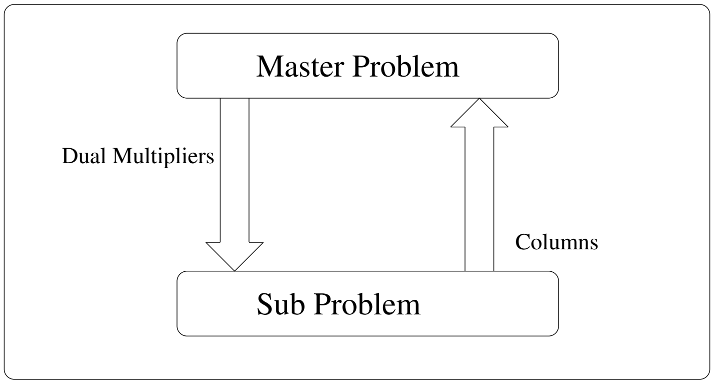


Figure 3.3: Column Generation Flow [50]

Some additional parameters and variables used in the formulation of the Master Problem and the Sub-Problem:

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:-

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

3.4.1 Initial Basic Feasible Solution

CG works in the feasible region with the initial basic feasible solution (BFS) serving as the seed for the algorithm. The CG algorithm then generates solutions with non-decreasing objective function values. Thus, a good initial BFS is essential for the effectiveness of the CG approach.

We use the MRH and the MNRH to provide the initial BFS. Thus, if X_1, X_2 are the solutions found using the MRH and the MNRH respectively, and if $z(x)$ is the objective function for the capacitated problem, then the initial BFS X for the CG is given by $\{X|\ X = X_1$ or X_2, and z(X) = max.(z(X_1), z(X_2))$\}$

3.4.2 Column Generation Formulation

The capacity constraint is applied as part of the sub-problem. This is because in *Formulation* 1, which is the stricter case and reflects constraint on the operational capacity, capacity requirements need to be satisfied every time period. The sub-problem evaluates the y_{jkt} variables and also fixes the x_{it} variables. The Master problem selects the solutions for each time period that gives the maximum objective function value. The master problem is solved as a relaxed LP with no binary constraints.

3.4.3 Master Problem

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

Maximize $\displaystyle\sum_{t=0}^{T}\sum_{k\in\Theta}\sum_{j=1}^{n}\sum_{s\in F_t}(1-p)p^{j-1}d_kF_{st}y_{sjkt} + \sum_{t=0}^{T}\sum_{i\in\Delta}\sum_{k\in\Theta}\sum_{s\in F_t}c_{ik}z_{sikt}f_{st} - \sum_{t=1}^{T}\sum_{i\in\Delta}Cw_{it}$

subject to

$$-w_{it} + \sum_{s\in F_{t-1}} F_{st-1}x_{sit-1} - \sum_{s\in F_t} F_{st}x_{sit} \leq 0 \ \forall \ i = 1, \ldots, |\Delta|, t = 1, \ldots, T \qquad (3.1)$$

$$-w_{it} + \sum_{s\in F_t} F_{st}x_{sit} - \sum_{s\in F_{t-1}} F_{st-1}x_{sit-1} \leq 0 \ \forall \ i = 1, \ldots, |\Delta|, t = 1, \ldots, T \qquad (3.2)$$

$$\begin{aligned}\sum_{s\in F_t} F_{st} &= 1 \qquad \forall \quad t=0,\dots,T && (3.3)\\ F_{st} &\in \{0,1\} \qquad \forall \quad s,t=0,\dots,T && (3.4)\\ w_{it} &\geq 0 \ \forall \ i=1,\dots,|\Delta|, t=1,\dots,T && (3.5)\end{aligned}$$

Variables F_{st} stand for the feasible solutions and replace the x_{it} and y_{jkt} variables which are constants. Constraint (3.3) ensures that only one solution should be selected for each time period t. If β, γ, δ be the dual multipliers generated after solving the relaxed master problem for constraints (3.1), constraints (3.2) and constraints (3.3) respectively then the sub-problem uses these dual multipliers to generate feasible solutions.

3.4.4 Sub-Problem

The sub-problem for time periods $t = 1$ to $t = T - 1$ is as follows:

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

subject to

$$\begin{aligned}\sum_{j=1}^{n} y_{jkt} - \sum_{i\in\Delta} r_{ikt}x_{it} &\leq 0 \qquad \forall \quad k=1,\dots,|\Theta| && (3.6)\\ \sum_{i\in\Delta} x_{it} &\leq n && (3.7)\\ z_{ikt} &\leq r_{ikt}x_{it} \ \forall \quad i=1,\dots,|\Delta|, k=1,\dots,|\Theta|, t=0,\dots,T, && (3.8)\\ \sum_{i\in\Delta} z_{ikt} &\leq 1 \quad \forall \quad k=1,\dots,|\Theta|, t=0,\dots,T, && (3.9)\\ \sum_{k\in\Theta} d_k z_{ikt} &\leq Q_{it} \quad \forall \quad i=1,\dots,|\Delta|, t=0,\dots,T, && (3.10)\\ z_{ikt} &\in \{0,1\} \quad \forall \quad i=1,\dots,|\Delta|, k=1,\dots,|\Theta|, t=0,\dots,T. && (3.11)\\ y_{jkt} &\leq 1 \ \forall \quad j=1,\dots,n, k=1,\dots,|\Theta| && (3.12)\\ x_{it} &\in \{0,1\} \qquad \forall \qquad i=1,\dots,|\Delta| && (3.13)\end{aligned}$$

The sub-problem is itself formulated in such a way that the new column generated has a favorable reduced cost to enter the basis of the master problem. Due to non-availability of the dual multipliers β and γ for $t = 0$ and $t = T$, the objective function has a different form for these time periods.

For $t = 0$:

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

For $t = T$:

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

The solution to the above sub-problem is a new variable F_{st} which enters the master problem.

3.4.5 Complete Solution Using Column Generation

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 [50]:

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.

The cycle of passing the dual multipliers to the subproblem and passing a feasible solution to the master problem continues until no more improvement in the objective function value of the relaxed master problem is possible.

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

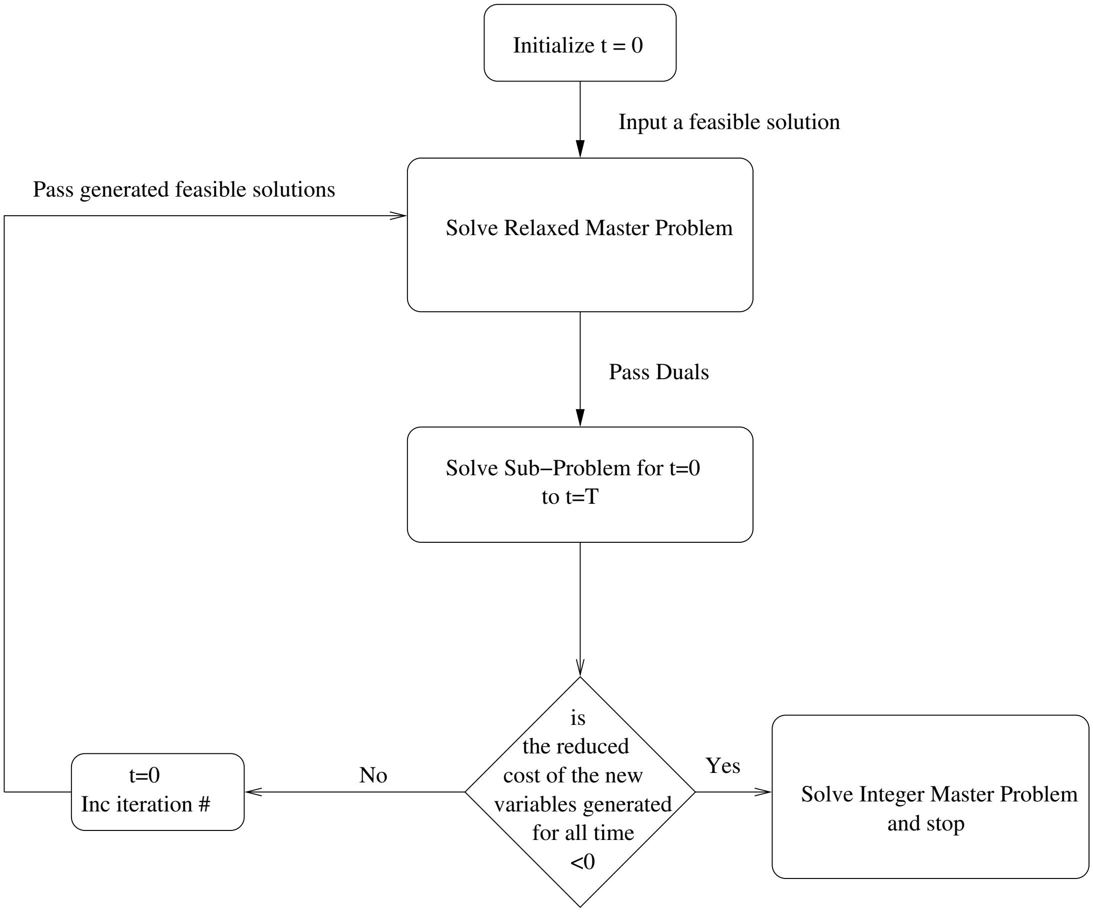

Initialize t = 0
Input a feasible solution
Pass generated feasible solutions
Solve Relaxed Master Problem
Pass Duals
Solve Sub−Problem for t=0 to t=T
is the reduced cost of the new variables generated for all time <0
No
Yes
t=0
Inc iteration #
Solve Integer Master Problem and stop

Figure 3.4: Solution Flowchart [50]

for each time period for the overall problem. The RMP terminates when the reduced cost of the generated columns is less than zero. It follows from the fact that 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. Other possible termination criterions can be a threshold time within which a solution is required or a bound on the number of iterations.

It should be noted that the Column Generation procedure described in this section is only a *heuristic*. It does not even guarantee optimality. It has been explored due to its ability to handle large sized problems (not possible in standard solvers) and for performing "reasonably" well for medium and small sized ones.

3.5 Chapter Summary

In this chapter, we modified two greedy heuristics of the Dynamic MEXCLP model to solve the Capacitated version of the Dynamic MEXCLP model presented in Sections 3.2, 3.3. Also we analyzed the worst case error bounds for each heuristic and showed that both the heuristics can perform arbitrarily bad. Where solution quality is of the essence and computational time is not important, we extended the Column Generation Scheme to incorporate the capacity constraint per time period. The next chapter presents the numerical studies carried out using the heuristics proposed in this chapter.

Chapter 4

Implementation and Numerical Studies

Based on the Solution Methodologies proposed in the previous chapter, we present numerical studies for the Capacitated Dynamic MEXCLP model.

4.1 Input Parameters

The parameters governing the problem structure (also refer [50].

1. n: Maximal number of facilities available for any given time period.

2. **Displacement Range**: The maximum displacement possible in X and Y direction per unit time.

3. **Displacement**: 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**: 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**: 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**: Range of the demand generated by the sensors lies.

7. $|\Delta|$: Specifies the number of potential locations available for locating facilities, Δ being the set of all potential facility locations.

8. $|\Theta|$: The number of sensors, Θ is the set of all sensors.

9. U: Coverage radius within which a facility can cover a sensor.

10. d: Demand of each sensor that needs to be fused per time period (can be assumed to be in megabytes).

11. A: The preference assignment constant. Note that it has replaced the c_{ij}'s in the formulation. It has been assumed that all cluster-head locations are equally preferred by the sensors.

12. M: The capacity of the cluster head locations. Again, this replaces Q_{it}'s in the formulation signifying that all cluster head locations have the same capacity.

13. p: Probability of link failure between the facility and sensors.

14. C: Relocation cost incurred per unit change in the number of facilities placed at a location between two successive time periods.

15. 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.

The software implementation took place on a machine with a Windows Platform and having the Intel Pentium 4 processor (1400 MHz) with 256 MB RAM. The Greedy and the Column Generation Heuristics were coded in the C Programming language with the commercial software CPLEX (Version 7.5) used for solving the Relaxed Master Problem (at each CG iteration) and the Integer Master Problem at the end. It is also used to solve the LP relaxation of the Capacitated Dynamic MEXCLP model and the integer formulation (wherever possible, for obtaining the optimal solution). The step-by-step description of algorithm implemented can be found in [50].

The results for some problem instances are discussed below in the next section.

4.2 Numerical Results for Some Problem Instances

The results are tabulated in Tables 4.1 to 4.12 with the results for each problem instance tabulated in two successive tables (thus 6 instances in all). The first two rows in each table reflect results for the uncapacitated version of the Dynamic MEXCLP problem and are presented for the purposes of comparison. For each instance, the problem is solved for six different capacity restrictions. The preference is kept constant at 2. The column headings are defined below:

1. **integ. LPObj.** : The obj. fn. value of the solution to the LP relaxation of the problem.
2. **MAX** : The maximum amongst the MRH and the MNRH heuristics.
3. **RMPObj.** : The Relaxed Master Problem obj. fn. value.
4. **RMP time** : The Relaxed Master Problem solution time.
5. **# itr** : Number of CG iterations. 1 CG iteration consists of solving Master Problem once and the Sub-Problem once.
6. **IMPObj.** : The Integer Master Problem obj. fn. value. This is the final solution given by the CG.
7. **IMP time** : The Integer Master Problem solution time.
8. **CG time** : The total CG solution time. This is the total of the RMP time and the IMP time.
9. **integ. IPObj.** : The optimal obj. fn. value provided by CPLEX.

10. **IP time** : Solution time for CPLEX.

11. **% improv. due to CG** : The relative improvement by CG of the obj. fn. value over the max. value provided by the two heuristics.

12. **Duality Gap(%)** : The relative gap between the integer optimal solution to the problem and the optimal solution to the relaxed problem.

13. **Optimality Gap(%)** : The relative gap between the solution obtained using CG and the optimal solution given by CPLEX.

It was found, as one might expect, that the addition of the capacity constraint significantly increased the complexity of the problem. This is quite evident from the increase in computational time for similar sized problems in the Dynamic MEXCLP model.

Some observations from numerical analysis are listed below:

1. It was found that adding the capacity constraint increased the complexity of the problem by a great extent. Eg: For the case with unlimited capacity, CPLEX was able to solve the problem with 150 potential locations quite easily as compared to the case with limited capacity (all other parameters and system resources remaining same). Also, problem sizes considered "small" for the Dynamic MEXCLP case turned out to be "large" with repect to system resources for the capacitated case. For instance, the CG heuristic could not even solve a case with $p = 0.7$ and 70 potential locations due to lack of system memory available, while it was easily able to do so for the uncapacitated case.

2. The gap between the LP Relaxation of the Integrated Problem and the Optimal Solution is much wider in the case with capacity. In some cases, this gap could be as wide as 30%.

3. The MRH and the MNRH heuristics provided "reasonably good" initial BFS - since CG could only improve, on an average, $< 2\%$ for small sized problems.

	p	C	integ. LPObj	LP time	MRH	MNRH	MAX	RMPObj	RMP time	# itr
	0.3	3	1712.6	0.43	1676.1	1670.2	1676.1	1695.95	21.611	33

M	p	C	int. LP	LP time	MRH	MNRH	MAX	RMPObj	RMP time	# itr
5	0.3	3	1120.8	0.821	1006.5	1026.2	1026.2	1039.8	116.187	104
10	0.3	3	1095.8	0.781	953.1	968.1	968.1	971.5	59.796	57
15	0.3	3	1160.5	0.782	979.7	1004.99	1004.99	1038.4	217.052	171
20	0.3	3	1148.1	0.781	968.4	982.1	982.1	1131.9	103.789	84
25	0.3	3	1148.9	0.782	936.5	978.25	978.25	1130.7	67.988	61
30	0.3	3	1131.9	0.781	959.5	966	966	1126.4	106.443	88

Table 4.1: Results (Part *a*) for Instance 1

4. **For varying** p : It was found that increaing p made the problem harder to solve. It increased the solution time for both the CG and CPLEX. Also, an increase in p tended to increase the Duality Gap. The improvement provided by the CG increased showing that the heuristics did not perform very well for high p's. However, the Optimality Gap also increased showing that CG itself may not perform so well as far as the optimal solution is concerned.

5. **For varying M** : No significant trend was observed for most performance values with respect to the varying M, even though one might observe that most of these performance indexes peaked around a value of M. Thus the average plot of these with respect to M might tend to be concave downwards.

4.3 Chapter Summary

In this chapter, the numerical studies done using the heuristics proposed in Chapter 3 were discussed. It was found that the greedy heuristics provided reasonably good solutions and the Column Generation heuristic was able to further refine these. It was also found that CG can solve larger problems where CPLEX fails to perform. The next chapter discusses some non-steady state generalizations of the problem.

IMPObj	**IMP time**	**CGtime**	**integ. IPObj**	**IP time**	**% improv. due to CG**	**Duality Gap (%)**	**Optimality Gap(%)**
1692.9	0.17	21.781	1712.6	0.33	1.002327	0	1.150298
IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
1039.8	0.06	116.247	1039.8	0.671	1.325278	7.7937	0
971.5	0.03	59.826	971.5	0.691	0.351203	12.79	0
1036.3	5.318	222.37	1038.4	0.711	3.115454	11.7582	0.202234
1131.9	0.06	103.849	1131.9	0.771	15.25303	1.43438	0
1126.1	17.054	85.042	1130.7	0.741	15.11372	1.61148	0.406828
1126.4	0.07	106.513	1126.4	0.861	16.60455	0.48595	0

Table 4.2: Results (Part *b*) for Instance 1

	p	**C**	**int. LP**	**LP time**	**MRH**	**MNRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
	0.5	3	1216.5	0.43	1167.8	1193	1193	1204.34	58.774	64
M	**p**	**C**	**int. LP**	**LP time**	**MRH**	**MNRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
5	0.5	3	828.3	0.811	716.5	733	733	739.063	117.549	100
10	0.5	3	822.6	0.792	680.5	691.5	691.5	691.5	65.334	55
15	0.5	3	877.99	0.832	702	722.75	722.75	751.552	146.921	100
20	0.5	3	859.79	0.771	682.5	701.5	701.5	845.875	124.99	100
25	0.5	3	862.19	0.761	653.5	702.25	702.25	845.625	129.446	100
30	0.5	3	849.97	0.731	678.5	690	690	843.5	107.745	100

Table 4.3: Results (Part *a*) for Instance 2

IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
1202	0.551	59.325	1216.5	0.35	0.754401	0	1.191944
IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
735	25.637	143.186	741	0.681	0.272851	11.7818	0.809717
691.5	0.04	65.374	691.5	0.731	0	18.9595	0
722.75	600.283	747.204	756.75	0.722	0	16.0206	4.492897
840.5	7.13	132.12	846.5	0.771	19.81468	1.57043	0.708801
836.25	53.647	183.093	846.25	0.711	19.08152	1.88307	1.181684
843.5	0.07	107.815	844.5	0.841	22.24638	0.64816	0.118413

Table 4.4: Results (Part *b*) for Instance 2

	p	C	int. LP	LP time	MRH	MNRH	MAX	RMPObj	RMP time	# itr
	0.7	3	444.36	0.14	409.89	441.06	441.06	441.06	0.26	0
M	**p**	**C**	**int. LP**	**LP time**	**RH**	**NRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
5	0.7	3	546.34	0.801	409.89	441.06	441.06	442.062	181.751	150
10	0.7	3	552.5	0.781	394.38	419.22	419.22	419.22	135.935	124
15	0.7	3	597.69	0.791	422.82	445.023	445.023	495.63	239.003	150
20	0.7	3	575.93	0.761	384.3	420.9	420.9	562.064	211.784	150
25	0.7	3	579.38	0.751	367.29	424.23	424.23	566.18	191.405	150
30	0.7	3	569.37	0.741	387.6	414	414	561.936	253.654	150

Table 4.5: Results (Part *a*) for Instance 3

IMPObj	IMP time	CGtime	int. IP	IP time	% improv.	% D Gap	% O Gap
441.06	0	0.26	444.36	0.14	0	0	0.742641
IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
441.06	3.795	185.546	444.36	0.661	0	22.9507	0.742641
419.22	0.091	136.026	419.22	0.661	0	31.7913	0
451.12	8584.4	8823.41	502.54	0.691	1.368918	18.9333	10.23302
545	335.773	547.557	564.9	0.761	29.48444	1.95226	3.522747
550.95	458.76	650.165	568.23	0.711	29.87059	1.96142	3.041022
536.7	563.35	817.004	563.9	0.852	29.63768	0.97068	4.82355

Table 4.6: Results (Part *b*) for Instance 3

	p	C	int. LP	LP time	MRH	MNRH	MAX	RMPObj	RMP time	# itr
	0.5	3	677	0.1	658.5	677	677	677	0.21	0
M	**p**	**C**	**int. LP**	**LP time**	**MRH**	**MNRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
5	0.5	3	743.84	0.48	658.5	677	677	677	36.513	57
10	0.5	3	796.69	0.461	662	682.5	682.5	682.5	13.459	21
15	0.5	3	801.78	0.441	669.25	669.5	669.5	693.5	107.014	150
20	0.5	3	816.07	0.451	621.5	661	661	805	59.636	79
25	0.5	3	784.43	0.451	628	635	635	779	46.077	72
30	0.5	3	847.68	0.44	679	695	695	841	76.089	120

Table 4.7: Results (Part *a*) for Instance 4

IMPObj	IMP time	CGtime	int. IP	IP time	% improv.	% D Gap	% O Gap
677	0.01	0.22	677	0.11	0	0	0
IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
677	0.04	36.553	677	0.401	0	9.87303	0
682.5	0.02	13.479	682.5	0.401	0	16.7305	0
693.5	0.1	107.114	693.5	0.421	3.584765	15.6133	0
787	73.345	132.981	805	0.441	19.06203	1.37559	2.236025
779	0.05	46.127	779	0.42	22.67717	0.69712	0
841	0.08	76.169	841	0.431	21.00719	0.79479	0

Table 4.8: Results (Part *b*) for Instance 4

	p	C	int. LP	LP time	MRH	MNRH	MAX	RMPObj	RMP time	# itr
	0.3	3	949.5	0.11	929.1	947.8	947.8	947.8	0.23	0
M	**p**	**C**	**int. LP**	**LP time**	**MRH**	**MNRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
5	0.3	3	1006.4	0.461	929.1	947.8	947.8	949.5	22.282	35
10	0.3	3	1061.1	0.471	929.2	955.5	955.5	955.5	12.568	19
15	0.3	3	1062.9	0.461	931	936.6	936.6	950.3	60.127	93
20	0.3	3	1081.4	0.45	884.5	925.4	925.4	1069.4	37.083	53
25	0.3	3	1040.4	0.44	879.2	889	889	1034	10.825	16
30	0.3	3	1126.6	0.451	951.8	973	973	1119.9	49.561	72

Table 4.9: Results (Part *a*) for Instance 5

IMPObj	IMP time	CGtime	int. IP	IP time	% improv.	% D Gap	% O Gap
947.8	0	0.23	949.5	0.11	0	0	0.179042
IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
949.5	0.03	22.312	949.5	0.391	0.179363	5.99211	0
955.5	0.02	12.588	955.5	0.401	0	11.0565	0
950.3	0.06	60.187	950.3	0.42	1.462738	11.8529	0
1045.4	37.865	74.948	1069.4	0.45	12.96737	1.11965	2.244249
1034	0.02	10.845	1034	0.451	16.31046	0.62191	0
1119.9	0.05	49.611	1119.9	0.451	15.09764	0.59686	0

Table 4.10: Results (Part *b*) for Instance 5

	p	**C**	**int. LP**	**LP time**	**MRH**	**MNRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
	0.7	3	406.2	0.1	385.92	406.2	406.2	406.2	0.23	0
M	**p**	**C**	**int. LP**	**LP time**	**MRH**	**MNRH**	**MAX**	**RMPObj**	**RMP time**	**# itr**
5	0.7	3	490.35	0.48	385.92	406.2	406.2	406.2	45.245	72
10	0.7	3	535.05	0.49	398.25	409.5	409.5	409.74	37.814	60
15	0.7	3	541.21	0.451	399.87	403.74	403.74	448.39	162.353	150
20	0.7	3	551.95	0.45	357.66	402.36	402.36	546.36	122.195	150
25	0.7	3	529.79	0.43	357.69	381	381	525	79.004	122
30	0.7	3	570.06	0.441	400.44	419.16	419.16	565.16	96.338	133

Table 4.11: Results (Part *a*) for Instance 6

IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
406.2	0	0.23	406.2	0.11	0	0	0
IMPObj	**IMP time**	**CGtime**	**int. IP**	**IP time**	**% improv.**	**% D Gap**	**% O Gap**
406.2	0.04	45.285	406.2	0.39	0	20.7159	0
409.74	0.04	37.854	409.74	0.401	0.058608	30.5821	0
446.2	19.018	181.371	451.7	0.43	10.51667	19.8171	1.217622
546.36	0.101	122.296	546.36	0.44	35.78885	1.02313	0
525	0.08	79.084	525	0.411	37.79528	0.91228	0
565.16	0.09	96.428	565.16	0.431	34.83157	0.86646	0

Table 4.12: Results (Part *b*) for Instance 6

Chapter 5

Some Non Steady-State Generalizations

Consider a **non-steady state** analysis of the problem, i.e., in which the link failure probabilities are not fixed but follow a distribution. Below, we discuss the formulation of the problem and demonstrate that it reduces to the one discussed previously in Chapter 2.

Consider a bipartite graph G_B (not necessarily complete) with vertex set V and edge set E. Let V_1, V_2 such that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \phi$, be the two layers. Also, consider that each edge, $e_{ij} \in E$ ($i \in V_1$ and $j \in V_2$), has the properties of failure and recovery (but each having different parameters and behaving independently of the others). The link failure (and possible recovery) here model the failure in connectivity between sensors and cluster heads due to uncertain causes such as terrain effects and instrument malfunction. The vertices (corresponding to sensors or cluster heads) are assumed not to fail.

Reliability of vertices ($i \in V_1$) in layer 1 can be considered with respect to connectivity with vertices in layer two ($j \in V_2$). The "connection" (or coverage) for vertex i is said to be reliable at time t if there exists at least one edge with vertex i that is working at time t. Thus, if $\mathbf{r_{ij}}(t)$ is the probability that the edge e_{ij} is working at time t, then the reliability for vertex i (i.e., the probability that vertex i is covered) over a time horizon T is

$$\mathbf{R_i} = \frac{1}{T}\int_0^T \left(1 - \left(\prod_{e_{ij}\,\epsilon\, E} (1 - \mathbf{r_{ij}}(t))\right)\right) dt \tag{5.1}$$

The $\mathbf{r_{ij}}(t)$ can be found for each edge at time t using Continuous Time Markov Chain models as discussed subsequently in Section 4.3. The *general expression* for the above integral tends to blow up even for relatively simple instances, such as, when the maximum degree for every $i \in V_1$ is 2 or 3. The weighed reliability of the network can now be found as

$$\mathbf{R} = \sum_{i \,\in\, V_1} w_i \mathbf{R_i} \tag{5.2}$$

where w_i is the weight ("importance") attached with sensor i. If as further simplification, it is assumed that all the links are identical, then this analysis might be applied (initially) for optimal placement of cluster-heads with respect to maximum sensor coverage over a time-horizon. So $r_{ij}(t)$ is now simply $r(t)$. Also, let $f(t) = 1 - r(t)$, the probability that the link is not working at time t. Thus, the problem of optimally locating cluster heads under non-steady state conditions can be written as

Objective → *Maximize the weighed total reliability of the network.*

Using (1), and remembering $f(t) = 1 - r(t)$, we have

Max. $\sum_{i \in V_1} w_i R_i = \sum_{i \in V_1} w_i \left(\frac{1}{T} \int_0^T \left(1 - (f(t))^{\sum_j x_{jt} r_{ijt}} \right) dt \right)$
S.T. $\qquad \sum_j x_{jt} \leq n \qquad 0 \leq t \leq T$

where x_{jt} is a $0-1$ variable taking a value of 1 if there is a cluster-head at possible cluster-head location j at time instant t, and 0 otherwise. r_{ijt} is a $0-1$ constant which is 1 if cluster-head location j covers sensor i at time t and 0 otherwise.

Cluster-heads are assumed to change positions at discrete time intervals (sensors are mobile but their positions at any time instant is known). To account for this, the time horizon is broken down into T time periods, during each period, the cluster-heads remain "stationary". The movement of cluster-heads from one location to another during successive time periods is assumed to be instantaneous. Changes take place at whole numbered time-points (indexed as k), starting at $t = 0$. Thus, the new (equivalent) objective along with the constraint is given by

$$\Leftrightarrow \text{Max. } \sum_{i\in V_1} w_i \left(\frac{1}{T}\sum_{k=1}^{T}\int_{k-1}^{k}\left(1-(f(t))^{\sum_j x_{j(k-1)}r_{ij(k-1)}}\right)dt\right)$$
$$\text{S.T.} \qquad \sum_j x_{jk} \le n \qquad \forall k$$

This objective can be linearized as follows

$$\Leftrightarrow \text{Max. } \sum_{i\in V_1} w_i \left(\frac{1}{T}\sum_{k=1}^{T}\int_{k-1}^{k}\left(\sum_{p=1}^{m}(1-f(t))\left(((f(t))^{p-1}y_{pi(k-1)})\right)dt\right)$$
$$\text{S.T.} \qquad \sum_{p=1}^{m} y_{pi(k-1)} - \sum_j x_{j(k-1)}r_{ij(k-1)} \le 0 \qquad \forall i,k$$
$$\& \qquad \sum_j x_{jk} \le n \qquad \forall k$$

where $y_{pi(k-1)}$ is a $0-1$ variable taking a value of 1 if sensor i is covered by atleast p cluster-heads during time period starting at time-point $k-1$ and 0 otherwise. m is the total number of cluster-heads. Interchanging the order of integration and summation, we have

$$\Leftrightarrow \text{Max. } \sum_{i\in V_1} w_i \left(\frac{1}{T}\sum_{k=1}^{T}\sum_{p=1}^{m}\left(\int_{k-1}^{k}(1-f(t))(f(t))^{p-1}dt\right)y_{pi(k-1)}\right)$$
$$\text{S.T.} \qquad \sum_{p=1}^{m} y_{pi(k-1)} - \sum_j x_{j(k-1)}r_{ij(k-1)} \le 0 \qquad \forall i,k$$
$$\& \qquad \sum_j x_{jk} \le n \qquad \forall k$$

the integral $\left(\int_{k-1}^{k}(1-f(t))((f(t))^{p-1}dt\right)$ might be pre-computed and essentially reduces to a constant for given limits of integration and given p. Thus, the problem of optimally locating cluster-heads under non-steady state conditions for maximum coverage (reliability of connection) of mobile sensors essentially reduces to the steady state version and can be formulated as a *Mixed Integer Linear Program* (MILP).

5.1 Jamming

We add the feature of jamming and consider it under various possible scenarios. Jamming of a link or a node (sensor or cluster head) might be due to adversarial action and is considered

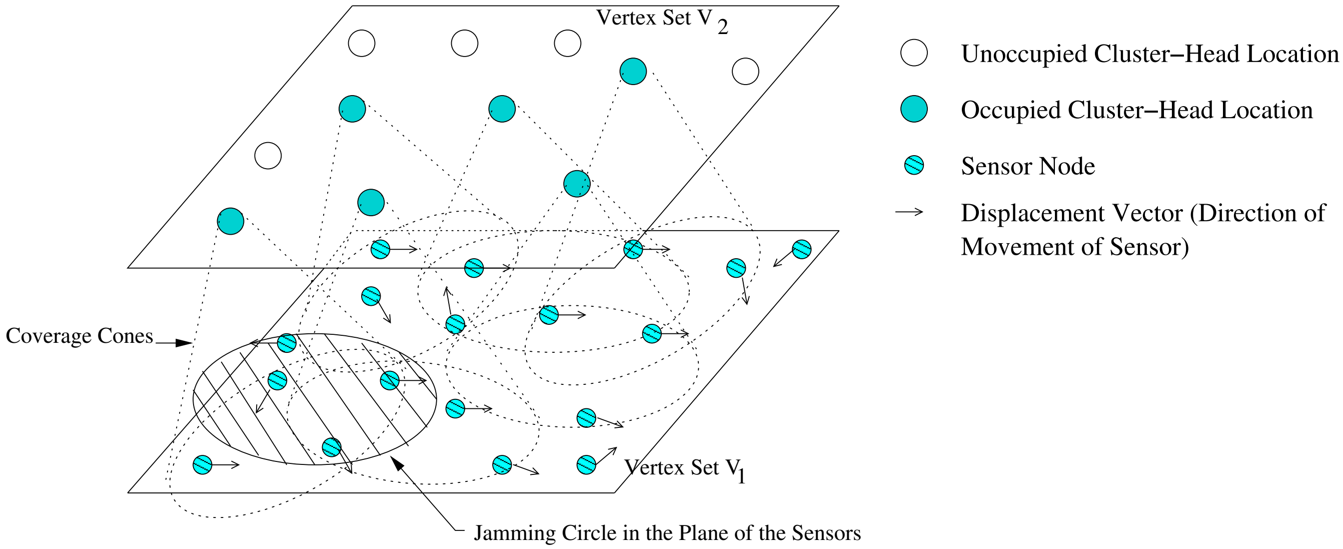

Vertex Set V 2
Unoccupied Cluster−Head Location
Occupied Cluster−Head Location
Sensor Node
Displacement Vector (Direction of Movement of Sensor)
Coverage Cones
Vertex Set V1
Jamming Circle in the Plane of the Sensors

Figure 5.1: An Example of jamming of sensors.

under some different cases as follows. In all the cases, however, jamming is assumed to behave in such a fashion that a jammed link or node is no longer capable of functioning once it is jammed (something like a 0 or 1 behaviour).

Case 1

1. A set of possible enemy scenarios is assumed, each scenario with some steady state probability (c_s, s for the scenario) of occurrence. A scenario is characterized by (possible) enemy jamming device locations and their range.

2. The probability of the scenarios do not change in different time periods (time indices).

3. "Jamming" of a sensor takes place if the sensor is within a known radius (jamming radius) of an enemy jamming device (center of hemisphere), as shown in Fig. 2. Thus, since the sensor locations are known for every time period, jamming of a sensor (under a given scenario) can be represented by a $0-1$ constant.

4. Links and cluster-heads are not jammed, rather sensors are jammed. So, a jammed sensor cannot communicate with *any* of the cluster-heads.

<u>*Formulation for Case 1*</u>: **Objective** → *Maximize the weighed total reliability of the network.*

$\Leftrightarrow$ Max. $\sum_{i\in V_1} w_i \sum_s c_s R_{is} = \sum_{i\in V_1} w_i \sum_s c_s \left\{\frac{1}{T}\int_0^T \left(1-(f(t))^{\sum_j x_{jt} r_{ijst}}\right) dt\right\}$

S.T. $\qquad \sum_j x_{jt} \le n \qquad 0 \le t \le T$

(Where R_{is} is the sensor connection reliability under scenario s and r_{ijst} is the $0-1$ constant which is 1 if cluster-head location j covers sensor i under scenario s at time t and 0 otherwise.)

$\Leftrightarrow$ Max. $\frac{1}{T}\sum_{i\in V_1} w_i \sum_s c_s \left\{\sum_{k=1}^T \int_{k-1}^k \left(1-(f(t))^{\sum_j x_{j(k-1)} r_{ijs(k-1)}}\right) dt\right\}$

S.T. $\qquad \sum_j x_{jk} \le n \qquad \forall k$

Note that $r_{ijs(k-1)} = 0 \quad \forall j$, when sensor i is jammed under scenario s. This is because given a scenario, and since the position of sensors is known, we know whether the sensor is jammed under the scenario or not. And since a jammed sensor is jammed for all possible cluster-head location, hence the condition. Thus, linearizing this as before,

$\Leftrightarrow$ Max. $\frac{1}{T}\sum_{i\in V_1} w_i \sum_s c_s \left\{\sum_{k=1}^T \int_{k-1}^k \left(\sum_{p=1}^m (1-f(t))\left((f(t))^{p-1} y_{pis(k-1)}\right)\right) dt\right\}$

S.T. $\qquad \sum_{p=1}^m y_{pis(k-1)} - \sum_j x_{j(k-1)} r_{ijs(k-1)} \le 0 \qquad \forall i,k,s$

& $\qquad \sum_j x_{jk} \le n \qquad \forall k$

where $y_{pis(k-1)}$ is again a $0-1$ variable which takes a value 1 if sensor i is covered by atleast p cluster-heads under scenario s at time index $k-1$. Thus, $y_{pis(k-1)}$ is a "virtual" variable without any real significance.

$\Leftrightarrow$ Max. $\frac{1}{T}\sum_{i\in V_1} w_i \sum_s c_s \left\{\sum_{k=1}^T \sum_{p=1}^m \left(\int_{k-1}^k (1-f(t))(f(t))^{p-1} dt\right) y_{pis(k-1)}\right\}$

S.T. $\qquad \sum_{p=1}^m y_{pis(k-1)} - \sum_j x_{j(k-1)} r_{ijs(k-1)} \le 0 \qquad \forall i,k,s$

& $\qquad \sum_j x_{jk} \le n \qquad \forall k$

the integral $\left(\int_{k-1}^k (1-f(t))((f(t))^{p-1} dt\right)$ can again be pre-computed and so the formulation is again an **MILP** as for the steady state location of cluster heads (henceforth referred to as the Dynamic Maximum Expected Coverage Problem, or *MEXCLP).*

__**Case** 2__

1. Things are the same except that the enemy scenarios can change in different time-periods.
2. So the scenario probabilities (the $c's$) now have two subscripts.

__Formulation for Case 2__: **Objective** → *Maximize the weighed total reliability of the network.*

$\Leftrightarrow$ Max. $\sum_{i\in V_1} w_i \left\{\frac{1}{T}\int_0^T \sum_s c_{st}\left(1-(f(t))^{\sum_j x_{jt} r_{ijst}}\right) dt\right\}$

S.T. $\quad \sum_j x_{jt} \le n \qquad 0 \le t \le T$

$\Leftrightarrow$ Max. $\frac{1}{T}\sum_{i\in V_1} w_i \left\{\sum_{k=1}^T \sum_s c_{s(k-1)} \int_{k-1}^k \left(1-(f(t))^{\sum_j x_{j(k-1)} r_{ijs(k-1)}}\right) dt\right\}$

S.T. $\quad \sum_j x_{jk} \le n \qquad \forall k$

(Note that $r_{ijs(k-1)} = 0 \qquad \forall j$, when sensor i is jammed under scenario s.)

$\Leftrightarrow$ Max. $\frac{1}{T}\sum_{i\in V_1} w_i \left\{\sum_{k=1}^T \sum_s c_{s(k-1)} \int_{k-1}^k \left(\sum_{p=1}^m (1-f(t))\left((f(t))^{p-1} y_{pis(k-1)}\right)\right) dt\right\}$

S.T. $\quad \sum_{p=1}^m y_{pis(k-1)} - \sum_j x_{j(k-1)} r_{ijs(k-1)} \le 0 \qquad \forall i, k, s$

& $\quad \sum_j x_{jk} \le n \qquad \forall k$

$\Leftrightarrow$ Max. $\frac{1}{T}\sum_{i\in V_1} w_i \left\{\sum_{k=1}^T \sum_s c_{s(k-1)} \sum_{p=1}^m \left(\int_{k-1}^k (1-f(t))(f(t))^{p-1} dt\right) y_{pis(k-1)}\right\}$

S.T. $\quad \sum_{p=1}^m y_{pis(k-1)} - \sum_j x_{j(k-1)} r_{ijs(k-1)} \le 0 \qquad \forall i, k, s$

& $\quad \sum_j x_{jk} \le n \qquad \forall k$

the integral $\left(\int_{k-1}^k (1-f(t))((f(t))^{p-1} dt\right)$ can again be pre-computed.

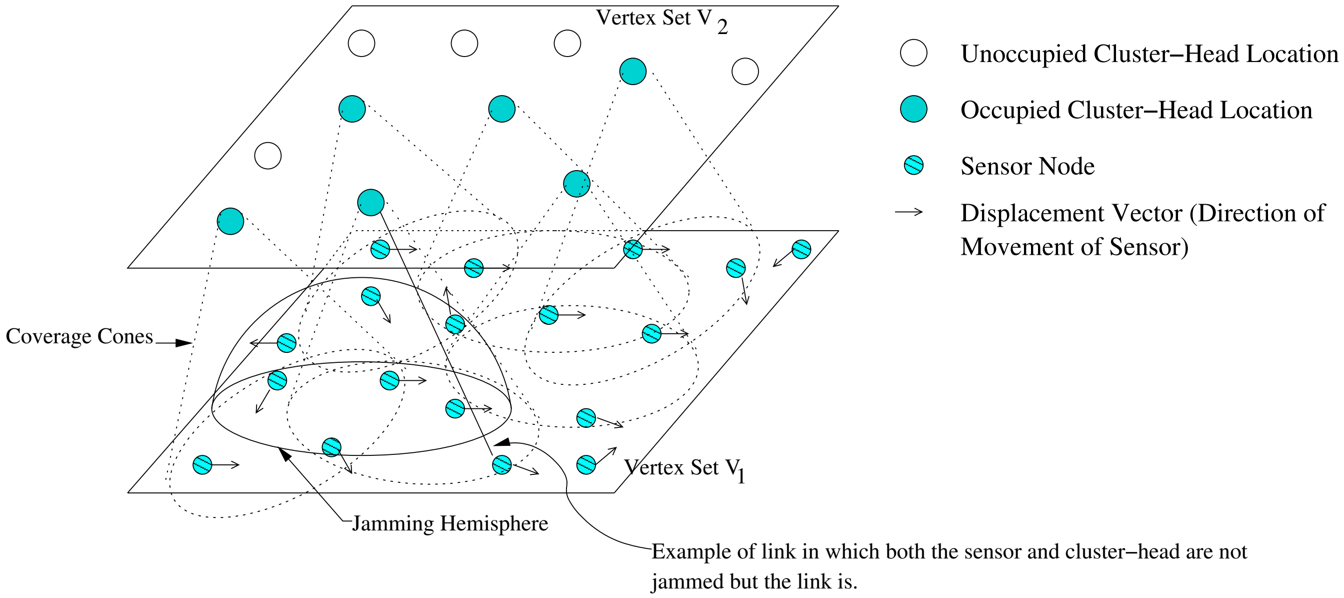

Vertex Set V 2
Unoccupied Cluster–Head Location
Occupied Cluster–Head Location
Sensor Node
Displacement Vector (Direction of Movement of Sensor)
Coverage Cones
Vertex Set V1
Jamming Hemisphere
Example of link in which both the sensor and cluster–head are not jammed but the link is.

Figure 5.2: An Example of jamming of links.

Case 3

1. If we assume that links are jammed instead of the sensors being jammed, the change is not much. $r_{ijs(k-1)} = 0$ only for specific (pre-determined) i, j pairs, instead for all j as we had earlier (note that i, j denote sensor and possible cluster head location.

2. Note that this case somewhat takes care of the situation when cluster-heads might also be jammed.

3. If enemy scenarios can change in different periods, the formulation for this case remains the same as the one for the previous case, with only the difference mentioned above in 1.

(Final) Formulation for Case 3:

Max. $\frac{1}{T}\sum_{i\in V_1} w_i \left\{\sum_{k=1}^{T} \sum_s c_{s(k-1)} \sum_{p=1}^{m} \left(\int_{k-1}^{k} (1-f(t))(f(t))^{p-1} dt\right) y_{pis(k-1)}\right\}$

S.T. $\sum_{p=1}^{m} y_{pis(k-1)} - \sum_j x_{j(k-1)} r_{ijs(k-1)} \leq 0 \qquad \forall i, k, s$

& $\sum_j x_{jk} \leq n \qquad \forall k$

$r_{ijs(k-1)} = 0$ if edge e_{ij} intersects the hemisphere determined by an enemy jamming location as center and jamming distance as radius (thus $r_{ijs(k-1)}$ is known) under scenario s and time-period $k-1$.

Case 4

- If the location of the enemy jamming centers/devices and the jamming radius is known, then the case is simpler for we just need to drop the $s-$subscript (since there are no more scenarios) and make the specific (pre-determined) $r_{ij(k-1)}$ terms zero.

Formulation for Case 4:

$\Leftrightarrow$ Max. $\sum_{i\in V_1} w_i R_i = \sum_{i\in V_1} w_i \left(\frac{1}{T}\int_0^T \left(1-(f(t))^{\sum_j x_{jt}r_{ijt}}\right) dt\right)$

S.T. $\sum_j x_{jt} \leq n \qquad 0 \leq t \leq T$

$\Leftrightarrow$ Max. $\sum_{i\in V_1} w_i \left(\frac{1}{T}\sum_{k=1}^{T}\sum_{p=1}^{m}\left(\int_{k-1}^{k}(1-f(t))(f(t))^{p-1}dt\right) y_{pi(k-1)}\right)$

S.T. $\sum_{p=1}^{m} y_{pi(k-1)} - \sum_j x_{j(k-1)}r_{ij(k-1)} \leq 0 \qquad \forall i, k$

& $\sum_j x_{jk} \leq n \qquad \forall k$

$r_{ij(k-1)} = 0$ if edge e_{ij} intersects the hemisphere determined by an enemy jamming location as center and jamming distance as radius (thus $r_{ij(k-1)}$ is known) at time-period $(k-1)$.

Case 5

- If the jamming centers are distributed in a way that is completely unknown to us, we can assume that (corresponding to **Case 1**) each sensor has a finite known probability of being jammed.

- Even in this case the formulation does not change much since the reliability of connection of the sensor i is now found using a conditioning argument as $R_i = \left\{\frac{1}{T}\int_0^T \left(1-(f(t))^{\sum_j x_{jt}r_{ijt}}\right)(1-a)dt\right\} + 0 \times a$, where a is the probability that the sensor i is jammed.

Formulation for Case 5:

Max. $\sum_{i\in V_1} w_i \left(\frac{1}{T}\sum_{k=1}^{T}\sum_{p=1}^{m}\left(\int_{k-1}^{k}(1-f(t))(f(t))^{p-1}dt\right) y_{pi(k-1)}\right)(1-a)$

S.T. $\sum_{p=1}^{m} y_{pi(k-1)} - \sum_j x_{j(k-1)} r_{ij(k-1)} \leq 0 \qquad \forall i, k$

& $\sum_j x_{jk} \leq n \qquad \forall k$

Determination of possible value(s) of a is an aspect that might have to be looked into in this case (since this would require estimation or probabilistic knowledge of enemy jamming device locations).

5.2 Charaterization of Sensor and Cluster Head Behaviour

Till now, we found that non-steady state analysis and the inclusion of jamming in the model (under certain conditions) does not essentially complicate the model and that if certain computations are done offline, they reduce to the Dynamic MEXCLP model. However, we have been completely oblivious to the information transmission and fusion in the network. We now further generalise the model by incorporating these aspects. In order to achieve this, we need to make certain assumptions regarding the sensor and cluster head behaviour and about information processing in general. These charaterizations are considered in the following different cases

Case 1

1. *Information :* The total (quantity of) information obtained is to be maximized. The usefulness of information is not based on its entirety, i.e., even parts of information might be useful.)

2. *Transmission of Information :* Assumed to be any known general function of time (can be, for instance, alternate states of transmission and no transmission, each state lasting for an amount of time distributed exponentially with some mean).

3. *Processing of Information :* Instantaneous (corresponding to just the receipt of information. The actual processing might be done separately.)

4. *Capacity of cluster heads :* Assumed to be satisfied over discrete time intervals and not instantaneously.

Objective → *Maximize the total information obtained.*

Max. $\sum_{i\in V_1}\int_0^T d_i(t)\left(1-(f(t))^{\sum_j x_{jt}r_{ijt}}\right)dt$

S.T. $\quad \sum_j x_{jt} \le 0 \qquad 0\le t\le T$

& $\quad \sum_i x_{jt}r_{ijt}\int_0^T d_i(t)dt \le c_t \qquad 0\le t\le T$

$\Leftrightarrow$ Max. $\sum_{i\in V_1}\sum_{k=1}^{T}\int_{k-1}^{k} d_i(t)\left(1-(f(t))^{\sum_j x_{j(k-1)}r_{ij(k-1)}}\right)dt$

S.T. $\quad \sum_j x_{jk} \le 0 \qquad \forall k$

& $\quad \sum_i x_{j(k-1)}r_{ij(k-1)}\int_{k-1)}^{k} d_i(t)dt \le c_{(k-1)} \qquad \forall k$

$\Leftrightarrow$ Max. $\sum_{i\in V_1}\sum_{k=1}^{T}\int_{k-1}^{k}\left(\sum_{p=1}^{m} d_i(t)(1-f(t))\left((f(t))^{p-1}y_{pi(k-1)}\right)\right)dt$

S.T. $\quad \sum_{p=1}^{m} y_{pi(k-1)} - \sum_j x_{j(k-1)}r_{ij(k-1)} \le 0 \qquad \forall i,k$

& $\quad \sum_j x_{jk} \le 0 \qquad \forall k$

& $\quad \sum_i x_{j(k-1)}r_{ij(k-1)}\int_{k-1)}^{k} d_i(t)dt \le c_{(k-1)} \qquad \forall k$

$\Leftrightarrow$ Max. $\sum_{i\in V_1}\sum_{k=1}^{T}\sum_{p=1}^{m}\left(\int_{k-1}^{k} d_i(t)(1-f(t))(f(t))^{p-1}dt\right)y_{pi(k-1)}$

S.T. $\quad \sum_{p=1}^{m} y_{pi(k-1)} - \sum_j x_{j(k-1)}r_{ij(k-1)} \le 0 \qquad \forall i,k$

& $\quad \sum_j x_{jk} \le 0 \qquad \forall k$

& $\quad \sum_i x_{j(k-1)}r_{ij(k-1)}\int_{k-1)}^{k} d_i(t)dt \le c_{(k-1)} \qquad \forall k$

the integral $\left(\int_{k-1}^{k} d_i(t)(1-f(t))(f(t))^{p-1}dt\right)$ might be pre-computed (if required, numerically) for known functions $d_i(t)$ and essentially reduces to a constant for given limits of integration and given p.

Case 2

1. *Information :* The total (quantity of) information obtained is to be maximized. The usefulness of information is not based on its entirety, i.e., even parts of information might be useful.)

2. *Transmission of Information :* Assumed to be any known general function of time (can be, for instance, alternate states of transmission and no transmission, each state lasting for an amount of time distributed exponentially with some mean).

3. *Processing of Information :* Assumed to take some time. Can have a general distribution.

4. *Capacity of cluster heads :* Assumed to be satisfied over discrete time intervals and not instantaneously.

Case 3

1. *Information :* Information is assumed to be in multiples of a basic unit. Information is useful only if obtained in its entirety. Thus if k units of information is being transmitted then the information is useful only if all k units are obtained.)

2. *Transmission of Information :* Poisson (batch) arrival process.

3. *Processing of Information :* Assumed to take some time. Can have a general distribution.

4. *Capacity of cluster heads :* Assumed to be satisfied instantaneously.

Such a system can be modelled using the (transient case) of the $M^{[X]}/G/1/k$ queueing system. We can minimize the loss for the system subject to capacity constraints. It should be noted here that link failures and jamming are not considered.

The last two cases will involve queueing type formulations and will have to be looked into.

5.3 Determination of $r_{ij}(t)$

We discuss the evaluation of $r_{ij}(t)$ here. Note again that this is not the same as the constant r_{ij}, signifying coverage in the MILP formulation.

Consider a single link that is susceptible to failure, with the mean time for which the link is working being distributed exponentially. The link can recover from the failure in time which might be a constant or also follows an exponential distribution. Let $\mathbf{r}(t)$ be the instantaneous reliability of the link (mathamatically, the probability that the link is working at time instant t). The reliability of the link (over a finite time horizon T) is then defined to be

$$\mathbf{R} = \frac{1}{T}\int_0^T \mathbf{r}(t)dt \tag{5.3}$$

When the time for which the link is not working also has an exponential distribution, the analysis is quite straight-forward using a Continuous Time Markov Chain Model and this is discussed first.

Symbols used,

- λ = rate at which failures arrive. Thus, time between the instants when the link starts working and the next failure arrives is distributed exponentially with parameter λ.
- Δ = time for which the link is not functioning (time for recovery). Thus, time between the instants when the link stops working and starts working again.
- T = time horizon under consideration.

5.3.1 When $\Delta \sim exp(\mu)$

The system can now be modelled as a Continuous Time Markov Chain(CTMC), with the state space consisting of 2 discrete states:

- **State 0**: Link is functioning

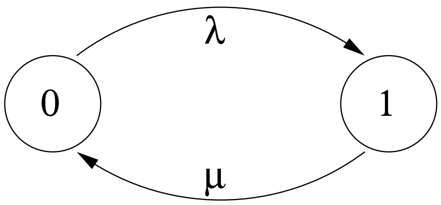


Figure 5.3: The transition diagram for the case when the link can be in 2 states with times distributed exponentially

- **State 1**: Link is not functioning due to "temporary" failure

The link remains in states 0 or 1 for times that are distributed exponentially with parameters λ and μ respectively [Figure 1.1].

If,

- $\mathbf{P}(\mathrm{t})$: $[P_{ij}(t)]$.
- $P_{ij}(t)$: State transition probabilities at time t.
- $\mathbf{I}$: the identity matrix.
- $\mathbf{Q}$: the infinitesimal generator matrix of the CTMC.
- $\mathbf{S}(\mathrm{t})$: $[S_j(t)]$, a row vector.
- $S_j(t)$: $P\{\mathbf{X}(t) = j\}$.
- $\mathbf{X}(\mathrm{t})$: random variable representing the state at time t.

Now, $\mathbf{S}(t) = \mathbf{S}(0) \cdot \mathbf{P}(t)$. Thus, to find the probability that the system is in some state, we just need to find $\mathbf{P}(\mathrm{t})$.

The infinitesimal generator for this case is given by

$$\mathbf{Q} = \left[\begin{array}{cc} -\lambda & \lambda \\ \mu & -\mu \end{array} \right] \tag{5.4}$$

The matrix **Laplace Transform** is

$$\mathbf{P}^e(s) = [s\mathbf{I} - \mathbf{Q}]^{-1} = \left[\begin{array}{cc} s+\lambda & -\lambda \\ -\mu & s+\mu \end{array} \right]^{-1} = \frac{1}{s(s+\lambda+\mu)} \left[\begin{array}{cc} s+\mu & \lambda \\ \mu & s+\lambda \end{array} \right] \tag{5.5}$$

Partial fraction expansion gives

$$\mathbf{P}^e(s) = \frac{1}{s} \left[\begin{array}{cc} \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \\ \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \end{array} \right] + \frac{1}{(s+\lambda+\mu)} \left[\begin{array}{cc} \frac{\lambda}{\lambda+\mu} & -\frac{\lambda}{\lambda+\mu} \\ -\frac{\mu}{\lambda+\mu} & \frac{\mu}{\lambda+\mu} \end{array} \right] \tag{5.6}$$

Inverting the transform, we find the answer in a closed form

$$\mathbf{P}(t) = \left[\begin{array}{cc} \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \\ \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \end{array} \right] + e^{-(\lambda+\mu)t} \left[\begin{array}{cc} \frac{\lambda}{\lambda+\mu} & -\frac{\lambda}{\lambda+\mu} \\ -\frac{\mu}{\lambda+\mu} & \frac{\mu}{\lambda+\mu} \end{array} \right] \tag{5.7}$$

Therefore, if $\mathbf{S}(0) = [p \quad (1-p)]$, the state at time t can be found as

$$\mathbf{S}(t) = [p \quad (1-p)] \left[\begin{array}{cc} \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \\ \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \end{array} \right] + e^{-(\lambda+\mu)t} [p \quad (1-p)] \left[\begin{array}{cc} \frac{\lambda}{\lambda+\mu} & -\frac{\lambda}{\lambda+\mu} \\ -\frac{\mu}{\lambda+\mu} & \frac{\mu}{\lambda+\mu} \end{array} \right] \tag{5.8}$$

But, $\mathbf{r}(\mathrm{t})$ = probability that link is working at time $t = P\{\mathbf{X}(t) = 0\}$. Thus,

$$\mathbf{r}(t) = p(\frac{\mu}{\lambda+\mu} + e^{-(\lambda+\mu)t}\frac{\lambda}{\lambda+\mu}) + (1-p)(\frac{\mu}{\lambda+\mu} - e^{-(\lambda+\mu)t}\frac{\mu}{\lambda+\mu}) \tag{5.9}$$

which gives

$$\mathbf{R} = \frac{1}{T}\int_0^T \mathbf{r}(t)dt = \frac{\mu}{\lambda+\mu} + \frac{(1-e^{-(\lambda+\mu)T})}{(\lambda+\mu)^2 T}((\lambda+\mu)p - \mu) \tag{5.10}$$

Thus, as $T \longrightarrow \infty$, $\mathbf{R} \longrightarrow \frac{\mu}{\lambda+\mu}$ and as $T \longrightarrow 0$, $\mathbf{R} \longrightarrow p$. An illustrative plot of $\mathbf{R}$ against p and T is shown at the end of Section 1.4.

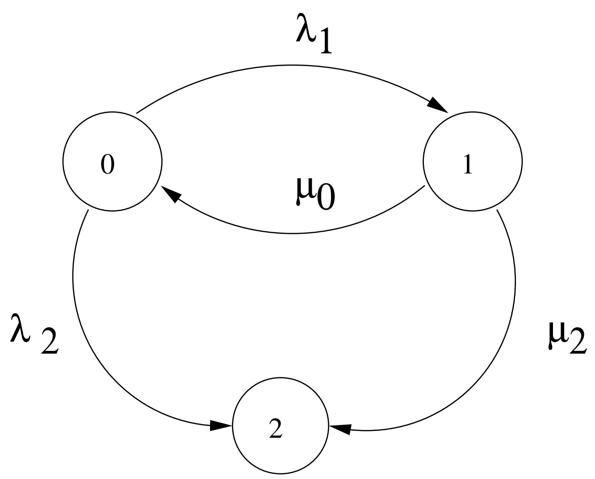


Figure 5.4: The transition diagram for the case when the link can be in 3 states

5.3.2 Consideration of catastrophic failure

- **State 0**: Link is functioning
- **State 1**: Link is not functioning due to "temporary" failure
- **State 2**: Link is not functioning due to "catastrophic or permanent" failure

Now, at every transition from state 0, the link can go to states 1 or 2 with known probabilities c_1 and c_2. From state 1, again, it could go to either states 0 or 2 (with corresponding known probabilities d_0 and d_2), and the link remains in state 2 once it reaches this state (corresponding to catastrophic failure) [Figure 1.2]. This system might be modelled as a 3-state CTMC with state 2 being the "absorbing" state. The link remains (analogous to the previous case) in states 0, 1 and 2 for times which are distributed exponentially with parameters λ, μ and 0 respectively. The infinitesimal generator for this system is given by

$$\mathbf{Q} = \begin{bmatrix} -\lambda & \lambda_1 & \lambda_2 \\ \mu_0 & -\mu & \mu_2 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.11}$$

where $\lambda_i = c_i\lambda$ $(i = 1, 2)$ and $\mu_j = d_j\mu$ $(j = 0, 2)$. Proceeding in a similar manner as in the previous section,

$$\mathbf{P}^e(s) = [s\mathbf{I} - \mathbf{Q}]^{-1} = \begin{bmatrix} s+\lambda & \text{-}\lambda_1 & \text{-}\lambda_2 \\ -\mu_0 & s+\mu & \text{-}\mu_2 \\ 0 & 0 & \text{s} \end{bmatrix}^{-1} \tag{5.12}$$

which gives,

$$\mathbf{P}^e(s) = \frac{1}{s(s^2+(\lambda+\mu)s+\lambda\mu-\lambda_1\mu_0)} \begin{bmatrix} s(s+\mu) & s\lambda_1 & s\lambda_2+\mu\lambda_2+\lambda_1\mu_2 \\ s\mu_0 & s(s+\lambda) & \lambda_2\mu_0+s\mu_2+\lambda\mu_2 \\ 0 & 0 & s^2+(\lambda+\mu)s+\lambda\mu-\lambda_1\mu_0 \end{bmatrix} \tag{5.13}$$

Partial fraction expansion gives

$$\begin{aligned} \mathbf{P}^e(s) &= \frac{1}{s}\begin{bmatrix} 0 & 0 & \frac{\lambda_2\mu+\lambda_1\mu_2}{\lambda\mu-\lambda_1\mu_0} \\ 0 & 0 & \frac{\lambda_2\mu_0+\lambda\mu_2}{\lambda\mu-\lambda_1\mu_0} \\ 0 & 0 & 1 \end{bmatrix} \\ &+ \frac{1}{(s^2+(\lambda+\mu)s+\lambda\mu-\lambda_1\mu_0)}\begin{bmatrix} s+\mu & \lambda_1 & -\frac{(\lambda_2\mu+\lambda_1\mu_2)s+\mu^2\lambda_2+\lambda_1\lambda_2\mu_0+\lambda\lambda_1\mu_2+\mu\lambda_1\mu_2}{\lambda\mu-\lambda_1\mu_0} \\ \mu_0 & s+\lambda & -\frac{(\lambda_2\mu_0+\lambda\mu_2)s+\lambda\lambda_2\mu_0+\mu\lambda_2\mu_0+\lambda^2\mu_2+\lambda_1\mu_0\mu_2}{\lambda\mu-\lambda_1\mu_0} \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{5.14}$$

Since we are interested only in state 0, we need only invert the 1^{st} column in the two matrices. Thus,

$$\mathbf{P}(t) = \begin{bmatrix} 0 & 0 & \frac{\lambda_2\mu+\lambda_1\mu_2}{\lambda\mu-\lambda_1\mu_0} \\ 0 & 0 & \frac{\lambda_2\mu_0+\lambda\mu_2}{\lambda\mu-\lambda_1\mu_0} \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \boxed{\mathbf{A}_1} & \boxed{\mathbf{A}_2} & \boxed{\mathbf{A}_3} \end{bmatrix} \tag{5.15}$$

where the $\mathbf{A}_i's$ are 3×1 column vectors. We are interested in $\mathbf{A}_1$, and it is given by

$$\mathbf{A}_1 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{5.16}$$

the $a_i's$ being,

$$a_1 = \frac{\left(e^{-\frac{1}{2}t(\lambda+\mu+K)}\right)\left(\lambda - \mu + (\mu-1)e^{tK} + (1+e^{tK})K\right)}{2K} \tag{5.17}$$

$$a_2 = \frac{e^{-\frac{1}{2}t(\lambda+\mu+K)}\left(-1+e^{tK}\right)\mu_0}{K} \tag{5.18}$$

$$a_3 = 0 \tag{5.19}$$

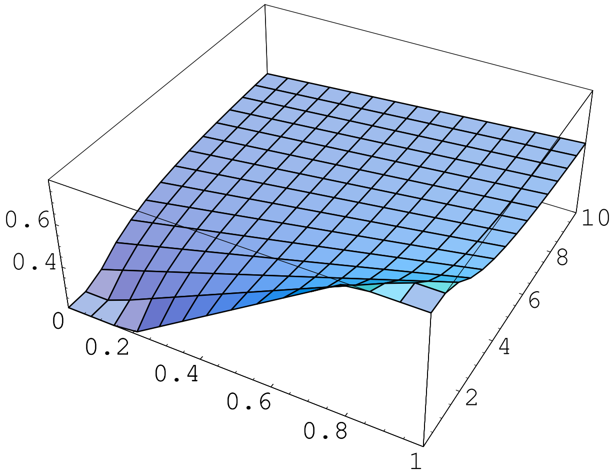
where $K = ((\lambda-\mu)^2 + 4\lambda_1\mu_0)^{\frac{1}{2}})$. Now, starting in state 2 would be meaningless since it would imply that the link is not functioning throughout the horizon. Thus, consider the starting state probability to be given by $\mathbf{S}(0) = [p \;\; (1-p) \;\; 0]$, the reliability is then found as

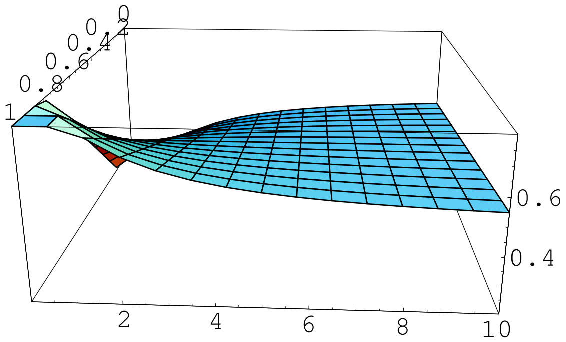
$$\begin{aligned} \mathbf{R} &= \frac{1}{T}\int_0^T \mathbf{r}(t)dt \\ or\ \mathbf{R} &= \frac{\left((1-p)\mu_0\left(\frac{K}{\lambda\mu-\lambda_1\mu_0} + \frac{\left(2e^{-\frac{1}{2}T(\lambda+\mu+K)}\left((\lambda+\mu)(e^{TK}-1)+K(1+e^{TK})\right)\right)}{4(\lambda\mu-\lambda_1\mu_0)}\right)\right)}{TK} \\ &+ \frac{\left(p\left(\frac{2\mu K}{\lambda\mu-\lambda_1\mu_0} + \frac{\left(4e^{-\frac{1}{2}T(\lambda+\mu+K)}\left((\lambda\mu-\mu^2-2\lambda_1\mu_0)(1-e^{TK})+\mu K(1+e^{TK})\right)\right)}{4(\lambda\mu-\lambda_1\mu_0)}\right)\right)}{2TK} \end{aligned} \tag{5.20}$$

Thus, as $T \longrightarrow \infty$, $\mathbf{R} \longrightarrow 0$ (since it would get "absorbed" in state 2).

When the time of failure has some other distribution, the analysis is not slightly more cumbersome. For instance, we discuss below the case when the link failure time is known (deterministic).

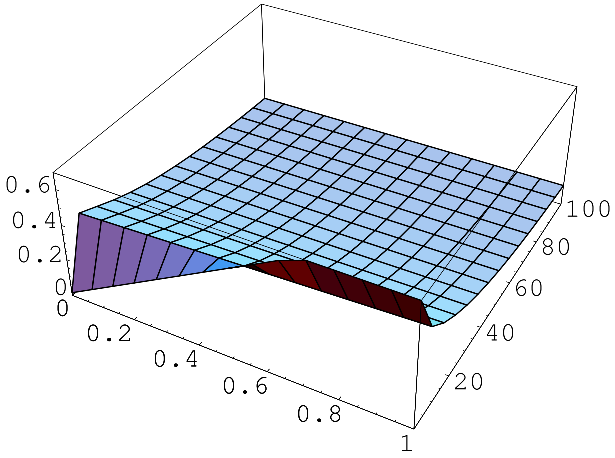
R = 1 / 2 + (1 - Exp[-T]) (p - 0.5) / T

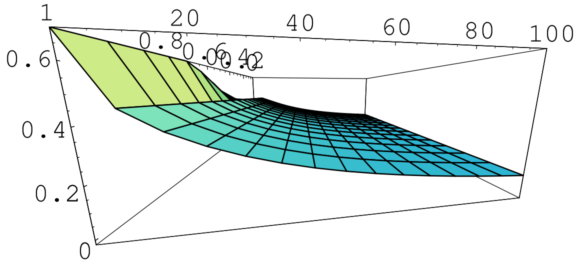




For the case when there are only two states. $\lambda = \mu = 0.5$

$$\mathbf{R} = \frac{1}{\mathbf{T}}(0.49999999999999994`\,(19.487179487179468` - 5.128205128205123`\,e^{-1.95`\,\mathbf{T}}\,(-0.10000000000000009` + 3.9`\,e^{1.9`\,\mathbf{T}}))\,(1-\mathbf{p})) + \frac{1}{\mathbf{T}}(0.2631578947368421`\,(38.974358974358935` - 10.256410256410247`\,e^{-1.95`\,\mathbf{T}}\,(0.09499999999999997` + 3.705`\,e^{1.9`\,\mathbf{T}}))\,\mathbf{p})$$





$\lambda = 1$
$\lambda_1 = 0.95$
$\lambda_2 = 0.05$
$\mu_0 = 0.95$
$\mu = 1$
$\mu_2 = 0.05$

5.3.3 When $\Delta \ll \frac{1}{\lambda}$, $\Delta \ll T$ and Δ is a constant

Based on the above assumptions which define this case, the process can be approximated as a Poisson process (since the probability of arrival of failure when the link is not functioning, which is of no consequence here but is essential in a Poisson process, is very small). Thus, the *probability of k failures during the time horizon* $= (\frac{e^{-\lambda T}(\lambda T)^k}{k!})$. So, the reliability is given by

$$\mathbf{R} = \frac{1}{T}[\sum_{k=0}^{\infty}(\frac{e^{-\lambda T}(\lambda T)^k}{k!})(T - k\Delta)] = \frac{1}{T}[T - \Delta\sum_{k=0}^{\infty}(\frac{e^{-\lambda T}(\lambda T)^k}{k!})(k)] \tag{5.21}$$

$\mathbf{R}$ can thus be simplified to

$$\mathbf{R} = \frac{1}{T}[T - \Delta\lambda T e^{-\lambda T}\sum_{k=0}^{\infty}(\frac{(\lambda T)^{k-1}}{(k-1)!})] = \frac{1}{T}[T - \Delta\lambda T] = [1 - \lambda\Delta] \approx e^{-\lambda\Delta} \tag{5.22}$$

5.3.4 When Δ is comparable with $\frac{1}{\lambda}$ and Δ is a constant

Let $\mathbf{X}_k$ be the time for which the link is working between the $(k-1)^{th}$ and the k^{th} failures. Thus $\mathbf{X}_k$ is distributed exponentially with parameter λ. Also, let $\mathbf{N}$ (a random variable) be the number of failures (completely contained) within the time horizon.

Now, since $\mathbf{X}_k \sim exp(\lambda) \implies \sum_{i=1}^{k}\mathbf{X}_i \sim gamma(k, \lambda)$. Thus, for constant α_1,

$$\mathbf{P}\left\{\sum_{i=1}^{k}\mathbf{X}_i = T - k\Delta - \alpha_1\right\} \approx \frac{\lambda e^{-\lambda(T-k\Delta-\alpha_1)}(\lambda(T-k\Delta-\alpha_1))^k}{k!}d\alpha_1 \tag{5.23}$$

Strictly speaking, it should have been $\mathbf{P}\left\{T - k\Delta - \alpha_1 - \frac{d\alpha_1}{2} \leq \sum_{i=1}^{k}\mathbf{X}_i \leq T - k\Delta - \alpha_1 + \frac{d\alpha_1}{2}\right\}$.

Consider initially that at times $t = 0$ and $t = T$, the link is functioning. Then the probability that there were k failures in time T is given by

$$\mathbf{P}\{\mathbf{N}=k\}=\int_{T-k\Delta}^{0}\left(\mathbf{P}\left\{\sum_{i=1}^{k}\mathbf{X}_i=T-k\Delta-\alpha_1\right\}\right)(\mathbf{P}\{\mathbf{X}_{k+1}\geq\alpha_1\})d\alpha_1 \tag{5.24}$$

Using the above equations (and recalling that $\mathbf{X}_k \sim exp(\lambda)$), we have

$$\mathbf{P}\{\mathbf{N}=k\}=\int_{T-k\Delta}^{0}\frac{\lambda e^{-\lambda(T-k\Delta-\alpha_1)}(\lambda(T-k\Delta-\alpha_1))^k}{k!}e^{-\lambda\alpha_1}d\alpha_1 \tag{5.25}$$

The reliability of the link is thus given by,

$$\mathbf{R}=\frac{1}{T}\sum_{k=0}^{\lfloor\frac{T}{\Delta}\rfloor}[\mathbf{P}\{\mathbf{N}=k\}][T-k\Delta] \tag{5.26}$$

Thus,

$$\mathbf{R}=\frac{1}{T}\left[Te^{-\lambda T}+\sum_{k=1}^{\lfloor\frac{T}{\Delta}\rfloor}\left(\int_{T-k\Delta}^{0}\frac{\lambda e^{-\lambda(T-k\Delta-\alpha_1)}(\lambda(T-k\Delta-\alpha_1)^k}{k!}e^{-\lambda\alpha_1}d\alpha_1\right)(T-k\Delta)\right] \tag{5.27}$$

To deal with the initial assumptions,

- If $\Delta \ll T$, then the initial assumptions would have little effect on the accuracy of the solution, and we could work with this approximation (as with $\lfloor\frac{T}{\Delta}\rfloor \approx \infty$).
- Let $\mathbf{R}=\mathbf{Rel}(T)$. Then, if δ was the time for which the link was functioning before $t=0$, the reliability is given by

$$\mathbf{R}_{(m)}=\int_{0}^{\Delta}f_\delta(\delta)\mathbf{Rel}(T-(\Delta-\delta))d\delta \tag{5.28}$$

where, $f_\delta(\delta)$=probability density function of δ.

- Consider the case when the link is not working at time $t = T$. The probability that we had $(k-1)$ failures during the time horizon is given by,

$$\mathbf{P}\left\{\sum_{i=1}^{k}\mathbf{X}_i = T-(k-1)\Delta-\alpha_2\right\} \approx \frac{\lambda e^{-\lambda(T-(k-1)\Delta-\alpha_2)}(\lambda(T-(k-1)\Delta-\alpha_2))^{k-1}}{(k-1)!}d\alpha_2 \tag{5.29}$$

Thus the modified $\mathbf{Rel}(T)$ is given by

$$\begin{aligned}\mathbf{Rel}_{(m)}(T) &= e^{-\lambda T} + \frac{1}{T}\left[\sum_{k=1}^{\lfloor\frac{T}{\Delta}\rfloor}\left(\int_{T-k\Delta}^{0}\frac{\lambda e^{-\lambda(T-k\Delta-\alpha_1)}(\lambda(T-k\Delta-\alpha_1)^{k}}{k!}e^{-\lambda\alpha_1}d\alpha_1\right)(T-k\Delta)\right]\\ &+ \frac{1}{T}\left[\sum_{k=1}^{\lfloor\frac{T}{\Delta}\rfloor}\left(\int_{\Delta}^{0}\frac{\lambda e^{-\lambda(T-(k-1)\Delta-\alpha_2)}(\lambda(T-(k-1)\Delta-\alpha_2)^{k-1}}{(k-1)!}(T-(k-1)\Delta-\alpha_2)d\alpha_2\right)\right]\end{aligned} \tag{5.30}$$

Thus the general expression for reliability ($\mathbf{R}_{(G)}$) of a single link when Δ is a constant is given by

$$\mathbf{R}_{(G)} = \int_0^{\Delta} f_\delta(\delta)\mathbf{Rel}_{(m)}(T-(\Delta-\delta))d\delta \tag{5.31}$$

As is evident, the theoretical evaluation of $r_{ij}(t)$ in all except the most trivial of cases, is quite impractical and one would need to resort to numerical approaches to evaluate it.

5.4 Chapter Summary

This chapter discussed some non-steady state generalizations of the Dynamic MEXCLP model. It was found that under suitable assumptions, the formulations for this case can be reduced to the steady state MILP formulations discussed earlier. Several cases involving the issue of Jamming were investigated. It was again shown that most of these cases can be reduced to the MILP formulations of the Dynamic MEXCLP model. Theoretical computation of the link failure probabilities were discussed and it was shown that these were not very practical to compute theoretically except for the most trivial of cases.

Chapter 6

Conclusions and Possible Future Work

Conclusions

The Capacitated Dynamic MEXCLP model was proposed and investigated. The model tried to capture the capacity constraint over a time horizon for the distributed sensing networks in which sensors and cluster heads can be mobile. An MILP formulation was proposed and solution procedures to solve the formulation were suggested. The work is a generalization of the Dynamic MEXCLP model proposed earlier. It can be also viewed as a new clustering approach for the capacitated *ad hoc* networks. The non-steady state generalizations proposed laid the ground work for further development in terms of generalizations to capture more realistic scenarios in which (link) failures need not obey steady state conditions. The specific conclusions that could be drawn are listed below.

- The addition of the capacity constraint was found to increase the complexity of the problem significantly. This is quite evident from the increase of solution times with respect to similar problems in the Dynamic MEXCLP model.

- The greedy heuristics proposed were found to provide reasonably good solutions quickly. These solutions could then be used as initial BFS for further refinement using Column Generation technique.

- The solution procedure involving CG could be used to solve large sized problems, representative of the real world. These problems (in fact problems which were *medium-sized* for the Dynamic MEXCLP model) could not be solved using CPLEX.

- However, CG was not found to be efficient with respect to computational time, in particular for small scale problems. Thus, for very small problems (which have correspondingly limited applicability) CPLEX might be preferred.

- It was proved that under some assumptions, the model for the case when link failure probabilities are not constant can be reduced to the MILP formulations for the steady state case. Thus, real-time implementation is possible if link reliabilities are precomputed.

- Various scenarios were investigated under which Jamming could occur and it was shown that most of them could be reduced to the basic dynamic expected coverage models.

Possible Future Work

In light of the conclusions above, some immediate future work that might be possible in this topic are discussed below:

- The *Non-steady state analysis* might be investigated further and link failure probabilities might be simulated to achieve a better, more reliable network design.

- *Variable relocation costs* might be considered to capture more realistic cluster head movements.

- *Energy* and *power* are important considerations that have been overlooked in this work. Models might be developed that incorporate these aspects.

- *Queueing models* with respect to capacity might be investigated to identify transmission problems, buffer capacities and information loss in the system.

- *Time Delays* and *Synchronization of clocks* in the system might be investigated.

Bibliography

[1] I. Averbakh and O. Berman. Minimax regret median location on a network under uncertainity. *INFORMS Journal on Computing*, 12(2):104–110, Sep. 2000.

[2] 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.

[3] M. O. Ball, J. S. Provan, and D. R. Shier. Reliability covering problems. *Networks*, 21:345–357, 1991.

[4] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(1):316–329, 1998.

[5] S. Basagni. Distributed clustering for ad hoc networks. *International Symposium on Parallel Architectures, Algorithms, and Networks*, 1999.

[6] S. Basagni, I. Chlamtac, and A. Farago. A generalized clustering algorithm for peer-to-peer networks. *ICALP '97*, 1997.

[7] P. Basu, N. Khan, and T. D. C. Little. A mobility based metric for clustering in mobile ad hoc networks. *IEEE*, 2001.

[8] R. Batta. Comment on the dynamics of plant layout. *Management Science*, 33(8):1065–1065, Aug. 1987.

[9] R. Batta, J. M. Dolan, and N. N. Krishnamurthy. The maximal expected covering location problem: Revisited. *Transportation Science*, 23(4):277–287, Nov. 1989.

[10] 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.

[11] 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.

[12] O. Berman and D. Krass. The generalized maximal covering location problem. *Computers and Operations Research*, 29:563–581, 2002.

[13] O. Berman and B. LeBlanc. Location-relocation of mobile facilities on a stochastic network. *Transportation Science*, 18(4):315–330, Nov. 1984.

[14] O. Berman and A. R. Odoni. Locating mobile servers on a network with markovian properties. *Networks*, 12:73–86, 1982.

[15] O. Berman and M. R. Rahnama. Optimal location-relocation decisions on stochastic networks. *Transportation Science*, 19(3):203–221, Aug. 1985.

[16] 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.

[17] S. C. Chapman and J. A. White. Probabilistic formulations of emergency service facilities location problems. *ORSA/TIMS Conference, San Juan, Puerto Rico*, 1974.

[18] 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.

[19] 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.

[20] W. Chen, N. Jain, and S. Singh. Anmp: Ad hoc network management protocol. *IEEE on Selected Areas in Communications*, 1999.

[21] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. *Proceedings of IEEE SICON'97, Singapore*, 1997.

[22] 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.

[23] R. Church and C. ReVelle. The maximal covering location problem. *Papers Regional Science Association*, 32:101–118, 1974.

[24] 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.

[25] M. S. Daskin. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, Feb. 1983.

[26] M. Gerla and J. T.-C. Tsai. Multicluster, mobile, multimedia radio network. *ACM-Baltzer Journal of Wireless Networks*, 1995.

[27] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. *Proceedings of IEEE ICUPC'97*, 1997.

[28] Z. J. Haas and S. Tabrizi. On some challenges and design choices in ad hoc communications. *Proceedings of IEEE MILCOM'98*, 1998.

[29] S. L. Hakimi. Optimum location of switching centers and the absolute centers and the medians of a graph. *Operations Research*, 12:450–459, 1964.

[30] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13:462–475, 1965.

[31] D. L. Hall and J. Llinas, editors. *Handbook of multisensor data fusion.* Boca Raton, FL : CRC Press, 2001.

[32] K. Hogan and C. Revelle. Concepts and applications of backup coverage. *Management Science*, 32(11):1434–1444, Nov. 1986.

[33] D. B. Johnson and D. A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. *Mobile Computing*, 1999.

[34] K. Kasera and R. Ramanathan. A location management protocol for hierarchically organized multihop mobile wireless networks. *Proceedings of IEEE ICUPC'97*, 1997.

[35] 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.

[36] M. Lübbecke and J. Desrosiers. Selected topics in column generation. Les Cahiers du GERAD G-2002-64, HEC Montréal, Canada, 2002. Submitted.

[37] V. Marianov and C. Revelle. The queuing probabilistic location set covering problem and some extensions. *Socio-Economic Planning Sciences*, 28(3):167–178, 1994.

[38] V. Marianov and D. Serra. Probabilistic maximal covering location-allocation models for congested systems. *Journal of Regional Science*, 38(3):401–424, Aug. 1998.

[39] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad hoc sensor networks. *IEEE Infocom 2001*, 3:1380–1387, Apr. 2001.

[40] E. Melachrinoudis and M. E. Helander. A single facility location problem on tree with unreliable edges. *Networks*, 27:219–237, 1996.

[41] P. B. Mirchandani and A. R. Odoni. Location of medians on stochastic networks. *Transportation Science*, 13(2):85–97, May 1979.

[42] 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.

[43] I. D. Moon and S. Chaudhry. An analysis of network location problems with distance constraints. *Management Science*, 30:290–307, 1984.

[44] 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.

[45] NIST. *http : //w3.antd.nist.gov/wahn_ssn.shtml. last updated*, Apr. 2003.

[46] S. H. Owen and M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, Dec. 1998.

[47] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., 1982.

[48] A. K. Parekh. Selecting routers in ad-hoc wireless networks. *Proceedings of the IEEE International Telecommunication Symposium*, 1994.

[49] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *Proceedings of INFOCOM'97*, 1997.

[50] D. J. Patel. *Clustering Sensors in Wireless Ad hoc Networks using a Dynamic Expected Coverage Model.* M.S. Thesis, University at Buffalo, State University of New York, 2003.

[51] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing for mobile computers. *Proceedings of ACM SIGCOMM'94*, 1994.

[52] 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.

[53] R. Ramanathan and M. Steenstrup. Hierarchically organized multihop mobile wireless networks for quality-of-service support. *Mobile Networks and Applications*, 1998.

[54] C. Revelle, J. Schweitzer, and S. Snyder. The maximal conditional covering problem. *INFOR*, 34(2):77–91, 1996.

[55] M. J. Rosenblatt. The dynamics of plant layout. *Management Science*, 32(1):76–86, Jan. 1986.

[56] 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.

[57] D. A. Schilling, V. Jayaraman, and R. Barkhi. A review of covering problems in facility location. *Location Science*, 1(1):25–55, 1993.

[58] 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.

[59] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19:1363–1373, 1971.

[60] G. L. Vairaktarakis and P. Kouvelis. Incorporation dynamic aspects and uncertainity in 1-median location problems. *Naval Research Logistics*, 46(2):147–168, Mar. 1999.

[61] J. A. White and K. E. Case. On covering problems and the central facilities location problem. *Geographical Analysis*, 1974.

[62] 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.