Optimization Based Methods for Solving Some Problems in Telecommunications and the Internet

Long Jia

School of Information Technology
and Mathematical Sciences
University of Ballarat
PO Box 663
University Drive, Mount Helen
Ballarat, Victoria 3353
Australia

This thesis is presented for the degree of Doctor of Philosophy of the University of Ballarat

April 2005

To my parents, Jiaping Chang, Zhiyi Jia,

my wife, Xiangrong Zheng
and my son, Jingqu Jia.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university and is less than 100,000 words in length excluding tables, maps, footnotes, bibliographies and appendices. To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made.

Long Jia

April 2005

Abstract

The purpose of this thesis is to develop some new algorithms based on optimization techniques for solving some problems in some areas of telecommunications and the Internet. There are two main parts to this thesis. In the first part we discuss optimization based stochastic and queueing models in telecommunications network corrective maintenance. In the second part we develop optimization based clustering (OBC) algorithms for network evolution and multicast routing.

The most typical scenario encountered during mathematical optimization modelling in telecommunications, for example, is to minimize the cost of establishment and maintenance of the networks subject to the performance constraints of the networks and the reliability constraints of the networks as well.

Most of these optimization problems are global optimization, that is, they have many local minima and most of these local minima do not provide any useful information for solving these problems. Therefore, the development of effective methods for solving such global optimization problems is important.

To run the telecommunications networks with cost-effective network maintenance, we need to establish a practical maintenance model and optimize it. In the first part of the thesis, we solve a known stochastic programming maintenance optimization model with a direct method and then develop some new models. After that we introduce queue programming models in telecommunications network maintenance optimization. The ideas of profit, loss, and penalty will help telecommunications companies have a good view of their maintenance policies and help them improve their service.

In the second part of this thesis we propose the use of optimization based cluster-

ing (OBC) algorithms to determine level-constrained hierarchical trees for network evolution and multicast routing. This problem is formulated as an optimization problem with a non-smooth, non-convex objective function. Different algorithms are examined for solving this problem. Results of numerical experiments using some artificial and real-world databases are reported.

Acknowledgement

The work on this thesis would never have been possible without the support of my principle supervisor, Professor Alex Rubinov. I have been very fortunate in having him as my supervisor. I can find no words to thank him on the excellent job he did in guiding and assisting me through my PhD study.

My sincere thanks to my associated supervisors, Dr. Adil Bagirov, Dr. Iradj Ouveysi for giving me a great deal of help. I could not have done this without their help. They gave their time and knowledge so generously during our numerous insightful discussions. I have found to work with them over the last more than two years - a very instructive experience that has given much strength to my future career. Dr. Gill Waters offered her published and unpublished papers and some useful discussion through emails, she deserves considerable credit. I would like also to thank Professor A. Wirth and H. S. Gan for their helpful discussion of their paper.

Thanks also to the head of our school, Professor Sidney A. Morris. He kindly offered the English class for the International students (Dr. Jack Harvey took the challenge to fight against all sorts of "our English ") and other help like thesis English checkers. The school has been developed so quickly under his supervision during my PhD study. So we can have more and more staff, visitor from all over the world and lot seminars; all these resources nourish my study.

My deepest thanks go to my wife and my son, for having put up with me for the past few years in my efforts to gain my doctorate. I know I have been demanding and you have always been there for me; I can never thank you enough. I hope that I have been worth all the pain! Filial piety is a very important part of our culture. My parents always support me to fulfil my ambition and forgive me not taking care

of them. Their love is infinite!

Sincere gratitude must be expressed to my colleagues without whom this thesis would not be possible: Zari Dzalilov, Julien Ugon, Nadejda Soukhoroukova, Jiapu Zhang, Musa Mammadov, Gary Sanders, Simon Barty, Glenn Auld, Michelle O'Brien, and Siarhei Dymkou. Among them Zari Dzalilov deserves my best regards: she gave me motherlike care, both in my study and life. Further thanks in this respect go to Associated Professor John Yearwood for acting as my previous associated supervisor in my first year study.

I would like to thank Professor Moshe Zukerman. He was my supervisor when I worked with him as a visiting scholar in the University of Melbourne. The work with him sharped my knowledge of optimization and telecommunications which helped me to get the scholarship for my PhD study.

I would be unfair if I did not mention my thesis's English checker, Ms. Rosemary Hay. To write my thesis in good English is my dream. Ms. Rosemary Hay helps me to make it true.

Finally I would like to acknowledge the support I received from the University of Ballarat, the School of Information Technology and Mathematical Science (ITMS), Center for Informatics and Applied Optimization and the Office of Research. My thanks to all those in ITMS who made it such a great place to work, and to my many friends who kept me going and encouraged me to finish. You all know who you are.

This research was supported by the University of Ballarat Postgraduate Research Scholarship.

List of publications

A. Bagirov, L. Jia, I. Ouveysi, "Nonsmooth optimization based heuristic algorithm for multicast group problem", submitted in 2004.

L. Jia, A. Bagirov, I. Ouveysi, A. M. Rubinov, "Optimization based clustering algorithms in multicast group hierarchies, " Australian Telecommunications, Networks and Applications Conference (ATNAC), Melbourne, Australia, 2003 (published on CD, ISNB 0-646-42229-4).

L. Jia, I. Ouveysi, and A. M. Rubinov, "Optimization in Telecommunications Network Maintenance", L. Caccetta, V. Rehbock, Industrial Optimisation Volume 1, pages 175-184, Proceedings of Symposium on Industrial Optimisation and the 9th Australian Optimisation Day, Curtin University, Perth, Australia, 2003.

Julien Ugon, L. Jia, and I. Ouveysi, "Queueing programming models in Telecommunications Network Maintenance", L. Caccetta, V. Rehbock, Industrial Optimisation Volume 1, pages 336-345, Proceedings of Symposium on Industrial Optimisation and the 9th Australian Optimisation Day, Curtin University, Perth, Australia, 2003.

Long Jia, I. Ouveysi, and A. M. Rubinov, "Optimization in Telecommunication Network Maintenance", Research Report 02/08,University of Ballarat, 2002 http://www.ballarat.edu.au/ard/itms/ publications/researchPapers/papers2002.shtml

L. Jia, I. Ouveysi, A. M. Rubinov, "A comparison of optimization methods in Multicast group Hierarchies", The 5th International Congress on Industrial and Applied Mathematics, ICIAM 2003, Sydney, Australia. (Presentation of the conference)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this study, we investigate three major issues of telecommunications and the Internet: networks maintenance, networks evolution and multicast routing. We applied optimization based methods to these problems. First we study optimization based stochastic and queueing models in telecommunication network corrective maintenance. Then we discuss optimization based clustering (OBC) algorithms for network evolution and multicast routing.

Our proposed optimization based stochastic and queueing algorithms will help telecommunication companies to run networks with a cost-effective network maintenance.

Our proposed optimization based clustering (OBC) algorithms offered a efficient solution with a novel approach to these networks evolution and multicast routing problems.

1.1 Background

The word communication derives from the Latin word communicare: to impart, participate. The science of "communication" is the study of all information transfer processes. Telecommunications entails disciplines, means, and methodologies to communicate over distances, in effect, to transmit voice, facsimile, and computer data. Telecommunications networks consist of three general categories of equipment:

termination equipment, transmission equipment, and switching equipment. Each of these three categories, in turn, comprises a number of subcategories or technologies.

1.1.1 Telecommunications and economic development

Today nearly all businesses are becoming information-technology companies through their use of or dependence on telecommunications products and services. From a business operations perspective, advanced telecommunications infrastructure is an increasingly important consideration in determining where to locate a business. Most research suggests that after workforce availability, the quality of the telecommunications infrastructure is generally considered to be among the top site location criteria. This is especially true for companies that base site and operational decisions on global economic and business factors.

As companies face cost pressures, advanced telecommunications infrastructure offers an unparalleled degree of freedom to disperse operations to low-cost areas, sometimes to previously unimaginable locations. The technological advances are making firms less inclined to set up centralized locations and more inclined to establish smaller 'satellite' facilities located in relatively remote locales. Telecommunications helps these decentralized operations to have closer connections than today's centralized facilities. For example, private Intranets for business communications are one way corporations are using electronic links and telecommunications to maintain close contact with worldwide operations and suppliers. All in all, the telecommunications as the new infrastructure foundation and an advanced system is essential for any economy (or company) to successfully compete in the modern global economy.

1.1.2 The Internet and telecommunications

The Internet is the latest in a long succession of communication technologies. Originally designed to link together a small group of researchers, the Internet is now used by many millions of people, and the number of users continue to grow at astonishing rates.

The unprecedented growth of the Internet over the last few years, and the ex-

pectation of an even faster increase in the numbers of users and networked systems, has resulted in the Internet assuming its position as a mass communication medium. At the same time, the emergence of an increasingly large number of application areas and the evolution of the networking technology suggest that in the near future the Internet may become the single integrated communication infrastructure. Even though the Internet is still small compared to the telephone and the cable TV networks in terms of the number of users and the quantity of capital invested, it has clearly joined them as a significant aspect of our telecommunications infrastructure. That is why when we deal with the optimization in telecommunications, we must take Internet into account. In this thesis we will try to employ optimization methods to solve some Internet problems include the network evolution and multicast routing.

1.2 Motivation

Because the telecommunications and Internet play a key role in the modern global economy, Telecommunications network availability has become an important criterion for mission-critical network-based services. This is particularly important as high speed and high capacity technologies are deployed. The impact of a single failure in this case can be catastrophic and a large number of services might be affected. For example, the nine hour breakdown of AT&T's long-distance telephone network in January 1990 resulted in a $60 million to $75 million loss in AT&T's revenues [8].

It is expected, therefore, that maintenance cost will become a significant percentage of the total cost especially as the price of bandwidth declines. The challenge of creating cost-effective network maintenance policy is often complicated by how to establish a practical model. Finding the optimal trade off between the cost of deployed capital and ongoing maintenance is therefore a very significant problem.

The maintenance of telecommunications and the Internet networks is one of the major expenses of telecommunications service providers. There are two principal ways to reduce downtime. One of them is to increase the number of service staff (operational expenditure), the other is to use more perfect and hence, more expensive equipment or duplication of existing equipment (capital expenditure). Due to competition, providers need to find the cheapest way to maintain the network. This

way can be found by methods of Operations Research, which lead to setting and solving some complicated problems of mathematical optimization. The most typical scenario encountered during a mathematical optimization modelling is:

minimize	$cost$
subject to	$performance\ constraint(s)$
	$reliability\ constraint(s)$

In fact there are many parameters to minimize in telecommunications and the Internet, for example, the delay. In this thesis we deal mainly with cost-based optimizations.

1.2.1 Optimization in telecommunications maintenance

Over the past few decades, optimization has become a powerful tool for solving problems arising in various areas of human activity. Many problems in engineering, economic and science can be formulated as optimization problems, i.e. problems in which an objective function, that depends on a number of variables, has to be optimized subject to a set of constraints.

Optimization has also been widely applied in telecommunications design, analysis, and production. Network optimization has always been a core problem domain in operations research. The field of network optimization is most commonly associated with the minimum cost flow problem and with several of its classical specializations: the shortest path problem, the maximum flow problem and the transportation problem.

However, less work has been done in the telecommunications maintenance optimization. The first part of my thesis will deal with this problem. We set models of telecommunications maintenance and apply optimization based algorithms to solve them.

1.2.2 Optimization in telecommunications and the Internet networks evolution

The success of the Internet is due to best-efforts delivery which allows for easy expansion, coupled with a congestion-adaptive reliable transport protocol (TCP), which serves well for delay insensitive traffic such as Web browsing or file transfers. There are several new proposals for handling Quality of Service (QoS) for real-time and multimedia traffic, but their introduction is slow. The scale of the Internet is one of the impediments to successful QoS provision. Hierarchical structure is the key to scaling problems, but it must be provided in a way that helps evolution too.

In order to do this, we set models for networks evolution with hierarchical structure. And several optimization based clustering (OBC) methods and their combinations with the k-means method are used to solve this problem.

1.2.3 Optimization in multicast routing

Instead of sending a separate copy of the data to each individual group member, a multicast source sends a single copy to all the members. An underlying multicast routing algorithm determines, with respect to certain optimization objectives, a multicast tree connecting the source(s) and group members. Data generated by the source flows through the multicast tree, traversing each tree edge exactly once. As a result, multicast is more resource-efficient, and is well suited to applications such as video distribution.

The problem of providing QoS in multicast routing is difficult due to a number of factors. First, distributed continuous media applications such as teleconference, video on demand, Internet telephony, and Web-based applications have very diverse requirements for delay, delay jitter, bandwidth, and packet loss probability. Multiple constraints often make the multicast routing problem intractable. Second, there are many practical issues that have to be taken into account when a routing algorithm is incorporated as part of a multicast routing protocol (e.g., state collection and update, handling of dynamic topology and membership changes, tree maintenance, and scalability). Adding in QoS further complicates the protocol design process.

Moreover, one has to consider how to collect/maintain QoS-related state at minimal cost, how to construct a QoS-satisfying route/tree in the presence of aggregated imprecise state information, and how to maintain QoS across routing domains.

Computing optimal Steiner Trees in graphs was used as an approach to the multicast routing ([31, 40]). The cost function can incorporate the QoS constraints. There are many papers that use heuristic methods to solve this problem([36], [103],[42]). Alternatively, hierarchical multicast trees offer cost-effectiveness, much more flexibility and scalability, as local trees are built and maintained independently in each cluster.

1.3 Outcomes

In this PhD study we have developed some new algorithms based on optimization techniques for solving some problems in telecommunications and the Internet.

There are two main achievements to this thesis. In the first part we discussed optimization based stochastic and queueing models in telecommunications network corrective maintenance. We solved a known stochastic programming maintenance optimization model with a direct method and then developed some new models. After that we introduced queue programming models in telecommunications network maintenance optimization. The ideas of profit, loss, and penalty will help telecommunications companies have a good view of their maintenance policies and help them improve their service.

In the second part we developed optimization based clustering (OBC) algorithms for network evolution and multicast routing. This problem is formulated as an optimization problem with a non-smooth, non-convex objective function. Different algorithms are examined for solving this problem. Results of numerical experiments using some artificial and real-world databases are reported.

1.4 Thesis outline

In chapter 2 we outline the relevant theory related to the thesis. Particularly we describe those algorithms developed by Dr. Bagirov and Professor Rubinov. We used these algorithms to solve problems in this thesis.

In chapter 3 we study stochastic programming models for maintenance. We describe a stochastic programming based maintenance model and solve it with a direct method and develop some new ones based on binomial and Poisson distributions.

In chapter 4 we introduce queue programming models in telecommunications networks maintenance. These models are formulated as global optimization problems. The latter problems have been solved by using the cutting angle method which was developed by Professor Alex Rubinov and his collaborators.

In chapter 5 we propose the use of optimization based clustering algorithms (OBC) to help in evolving a network. We compare several optimization based clustering methods and their combinations with the k-means method. The results lead to some useful conclusions and promising directions for further study.

In chapter 6 we propose non-smooth optimization based multi-level clustering algorithm to determine multi-level hierarchical multicast trees. The latter problem is formulated as an optimization problem with a non-smooth, non-convex objective function. Results of numerical experiments using some artificial and real-world databases are reported which show the high effectiveness of the proposed algorithm.

Finally, an appendix contains a number of miscellaneous ideas about possible optimization based applications in telecommunications and the Internet. These ideas are intended to offer future research directions in this area.

Chapter 2

Relevant theory

In order to help readers better understand this thesis, we outline the relevant theory related to the thesis. For the details, if needed, readers can read the references and books mentioned in this thesis.

2.1 Introduction

Optimization problems are made up of three basic ingredients:

A set of unknowns or variables which represent parameters for which we want to find the best set of values which satisfy our problems. In manufacturing problems, the variables might include the amounts of different resources used or the time spent on each activity. In fitting-the-data problem, the unknowns are the parameters that define the model. In the panel design problem, the variables used define the shape and dimensions of the panel.

An objective function which we want to minimize or maximize. Objective function is a mathematical function to evaluate the quality of the values of the variables. For instance, in a manufacturing process, we might want to maximize the profit or minimize the cost. In fitting experimental data to a user-defined model, we might minimize the total deviation of observed data from predictions based on the model. In designing an automobile panel, we might want to maximize the strength.

A set of constraints that allow the unknowns to take on certain values but exclude others. For the manufacturing problem, it does not make sense to spend a negative amount of time on any activity, so we constrain all the "time" variables to be non-negative. In the panel design problem, we would probably want to limit the weight of the product and to constrain its shape. The optimization problem is then:

Find values of the variables that minimize or maximize the objective function while satisfying the constraints. It can be formulated as:

$$\min(or\ \ max)\ \ f(x)$$

$$\text{subject to} \quad x \in X,\ \ g_i(x) \leq 0, i = 1, ...m,\ \ and\ h_j(x) = 0, j = 1, ...k.$$

Here $f(x)$ is the objective function, x is the variables.

According to different kinds of objective functions, variables, and constraints, there are many branches of optimization. Readers can get more details from the NEOS [15] and other books ([14][9]). In the following sections we will briefly describe several optimization problems which are important or related to my thesis.

2.1.1 Integer programming

In many applications, the solution of an optimization problem makes sense only if certain of the unknowns are integers. Integer linear programming problems have the general form

$$\min\{C^T x : Ax = b, x \geq 0, x \in Z^n, \}$$

where Z^n is the set of n-dimensional integer vectors. In mixed-integer linear programs, some components of x are allowed to be real. We restrict ourselves to the pure integer case, bearing in mind that the software can also handle mixed problems with little additional complication of the underlying algorithm.

Integer programming problems, such as the fixed-charge network flow problem and the famous travelling salesman problem, are often expressed in terms of binary

variables. The fixed-charge network problem modifies the minimum-cost network flow paradigm by adding a term $f_{ij}y_{ij}$ to the cost, where the binary variable f_{ij} is set to 1 if arc (i, j) carries a nonzero flow x_{ij}; it is set to zero otherwise.

In other words, there is a fixed overhead cost for using the arc at all. In the travelling salesman problem, we need to find a tour of a number of cities that are connected by directed arcs, so that each city is visited once and the time required to complete the tour is minimized. One binary variable is assigned to each directed arc; a variable x_{ij} is set to 1 if city i immediately follows city j on the tour, and to zero otherwise. In most of our study in this thesis, the number of repair persons, hubs etc. are all integers.

2.1.2 Knapsack problem

Suppose a hitch-hiker has to fill up his knapsack by selecting from among various possible objects those which will give him maximum comfort. This knapsack problem can be mathematically formulated by numbering the objects from 1 to n and introducing a vector of binary variables $x_j(\ j = 1, 2, ...n)$ having the following meaning:

$$x_j = \begin{cases} 1 & \text{if object } j \text{ is selected;} \\ 0 & \text{otherwise.} \end{cases}$$

Then, if p_j is a measure of the comfort given by object (j), (w_j) its size, and the size of the knapsack c, our problem will be to select, from among all binary vectors x satisfying the constraint

$$\sum_{j=1}^{n} w_j x_j \leq c,$$

the one which maximizes the objective function

$$\sum_{j=1}^{n} p_j x_j.$$

A naive approach would be to program a computer to examine all possible binary vectors x, selecting the best of those which satisfy the constraint. Unfortunately, the number of such vectors is 2^n. So the knapsack problems are NP-hard problems. For more details, see [16].

The above example is known as the $0-1$ Knapsack problem and can be formulated as:

$$\text{minimize} \quad \sum_{j=1}^{n} p_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \leq c,$$

$$x_j = 0 \text{ or } 1,\ j = 1, ..., n,$$

2.1.3 Global optimization

Optimization is a powerful tool for solving problems arising in various areas of human activity. Many problems in engineering, economics and science can be formulated as optimization problems. We shall consider only minimization problems because maximization problems can be reduced to the former by replacing the objective function f by the function $-f$.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous real valued objective function and let $X \subset \mathbb{R}^n$ be a bounded set. We shall consider the following general optimization problem:

$$\inf \ f(x) \quad \text{subject to } x \in X. \tag{2.1.1}$$

A point x^* is called *a local minimum* of the function f on X if there exists a neighborhood $B_\varepsilon(x^*) = \{x \in \mathbb{R}^n \mid \|x - x^*\| < \varepsilon\},\ \varepsilon > 0$ of x^* such that

$$f(x^*) \leq f(x) \ \ \forall x \in B_\varepsilon(x^*) \bigcap X.$$

In general several local minima may exist and the corresponding function values may differ substantially. Most of the nonlinear programming methods aim for a

local minimum or maximum. The problem of designing algorithms that find the best possible minimum among these local ones is known as the global optimization problem.

A point x^* is called *a global minimum* of the function f over X if

$$f(x^*) \leq f(x) \quad \forall x \in X.$$

In the last few decades the problems of global optimization have been intensively studied by many authors. A number of textbooks exist which also provide more or less detailed surveys, related to certain areas of global optimization (see, for example, [23][24][190]).

A considerable variety of specifications of the global optimization problems exist. These may substantially differ in their underlying analytical assumptions, related to the structure of the set X and the function f. Here we can mention Lipschitz programming, concave minimization and D.C. (Difference of convex functions) programming problems.

2.2 Clustering

Clustering is the *unsupervised* classification of patterns. Cluster analysis deals with the problems of organization of a collection of patterns into clusters based on similarity. It has found many applications, including information retrieval, document extraction, image segmentation etc.

In cluster analysis we assume that we have been given a finite set X of points of d-dimensional space $\mathbb{R}^d$, that is

$$X = \{x^1, \ldots, x^n\}, \text{ where } x^i \in \mathbb{R}^d, \ i = 1, \ldots, n.$$

The subject of cluster analysis is the partition of the set X into a given number q of overlapping or disjoint subsets $C_i, \ i = 1, \ldots, q$ with respect to predefined criteria such that

$$X = \bigcup_{i=1}^{q} C_i.$$

The sets $C_i,\ i = 1, \ldots, q$ are called clusters. The clustering problem is said to be *hard clustering* if every data point belongs to one and only one cluster. Unlike hard clustering in the *fuzzy clustering* problem the clusters are allowed to overlap and instances have degrees of appearance in each cluster. We will exclusively consider the hard unconstrained clustering problem, that is we additionally assume that

$$C_i \bigcap C_k = \emptyset, \quad \forall i, k = 1, \ldots, q,\ i \neq k.$$

and no constraints are imposed on the clusters $C_i,\ i = 1, \ldots, q$. Thus every point $x \in X$ is contained in exactly one and only one set C_i.

Each cluster C_i can be identified by its center (or centroid). Then the clustering problem can be reduced to the following optimization problem (see [110, 165, 194]):

$$\text{minimize} \quad \varphi(C, a) = \frac{1}{n} \sum_{i=1}^{q} \sum_{x \in C_i} \|a^i - x\|^2 \tag{2.2.1}$$

$$\text{subject to} \quad C \in \overline{C},\ a = (a^1, \ldots, a^q) \in \mathbb{R}^{d \times q}$$

where $\|\cdot\|$ denotes the Euclidean norm, $C = \{C_1, \ldots, C_q\}$ is a set of clusters, $\overline{C}$ is a set of all possible q-partitions of the set X, a^i is the center of the cluster $C_i,\ i = 1, \ldots, q$:

$$a^i = \frac{1}{|C_i|} \sum_{x \in C_i} x,$$

and $|C_i|$ is a cardinality of the set $C_i,\ \ i = 1, \ldots, q$. The problem (2.2.1) is also known as the minimum sum-of-squares clustering. The combinatorial formulation (2.2.1) of the minimum sum-of-squares clustering is not suitable for direct application of mathematical programming techniques. The problem (2.2.1) can be rewritten as the following mathematical programming problem:

$$\text{minimize} \quad \psi(a, w) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{q} w_{ij} \|a^j - x^i\|^2 \tag{2.2.2}$$

$$\text{subject to} \quad \sum_{j=1}^{q} w_{ij} = 1,\ i = 1, \ldots, n,$$

and

$$w_{ij} \in \{0, 1\},\ i = 1, \ldots, n,\ j = 1, \ldots, q.$$

Here

$$a^j = \frac{\sum_{i=1}^{n} w_{ij} x^j}{\sum_{i=1}^{n} w_{ij}}, \quad j = 1, \ldots, q$$

and w_{ij} is the association weight of pattern x^i with cluster j (to be found), given by

$$w_{ij} = \begin{cases} 1 & \text{if pattern } i \text{ is allocated to cluster j } \forall\ i = 1, \ldots, n,\ j = 1, \ldots, q, \\ 0 & \text{otherwise.} \end{cases}$$

$w = (w_{ij})$ is an $n \times q$ matrix.

There exist different approaches to clustering including agglomerative and divisive hierarchical clustering algorithms as well as algorithms based on mathematical programming techniques. Descriptions of many of these algorithms can be found, for example, in [170, 179, 194]. An excellent up-to-date survey of existing approaches is provided in [180] and a comprehensive list of literature on clustering algorithms is available in this paper.

Problem (2.2.2) is a global optimization problem. Therefore different algorithms of mathematical programming can be applied to solve this problem. Some review of these algorithms can be found in [174] with dynamic programming, branch and bound, cutting planes, k-means algorithms being among them. Dynamic programming approach can be effectively applied to the clustering problem when the number of instances $n \leq 20$, which means that this method is not effective to solve real-world problems (see [181]). However, when $q = 1$ the minimum sum-of-squares clustering problem can be solved exactly by dynamic programming, in polynomial time [194].

Branch and bound algorithms are effective when the database contain only hundreds of records and the number of clusters is not large (less than 5) (see [169, 173, 183, 174]). For these methods the solution of clustering problems with $n \geq 1000$ and $q \geq 10$ is out of reach. Different heuristics can be used for solving large clustering problems and k-means is one of such algorithms. Different versions of this algorithm have been studied by many authors (see [194]). This is a very fast algorithm and it is suitable for solving clustering problems in large data sets. k-means gives good results when there are few clusters but deteriorates when there are many [174]. This algorithm achieves a local minimum of problem (2.2.1)(see [192]), however results of numerical experiments presented, for example, in [177] show that the best clustering found with k-means may be more than 50 % worse than the best known one.

Much better results have been obtained with metaheuristics, such as simulated annealing, tabu search and genetic algorithms [188]. The simulated annealing approaches to clustering have been studied, for example, in [166, 193, 195]. Application of tabu search methods for solving clustering problem is studied in [156]. Genetic algorithms for clustering have been described in [188]. The results of numerical experiments, presented in paper [157] show that even for small problems of cluster analysis when the number of entities $n \leq 100$ and the number of clusters $q \leq 5$ these algorithms take 500-700 (sometimes several thousands) times more CPU time than the k-means algorithms. For relatively large databases one can expect that this difference will increase. This makes metaheuristic algorithms of global optimization ineffective for solving many clustering problems. However, these algorithms can be applied to large clustering problems if combined with decomposition (see [176]).

An approach to cluster analysis problems based on bilinear programming techniques has been described in [185]. The paper [163] describes the global optimization approach to clustering and demonstrates how the supervised data classification problem can be solved via clustering. The objective function in this problem is both nonsmooth and nonconvex and this function has a large number of local minimizers. Problems of this type are quite challenging for general-purpose global optimization techniques. Due to the large number of variables and the complexity of the objective function these techniques, as a rule, fail to solve such problems.

In [171] an interior point method for minimum sum-of-squares clustering problem is developed. The papers [176, 186] develops variable neighborhood search algorithm and the paper [175] presents j-means algorithm which extends k-means by adding a jump move. The global k-means heuristic, which is an incremental approach to minimum sum-of-squares clustering problem, is developed in [184]. The incremental approach is also studied in the paper [177]. Results of numerical experiments presented show the high effectiveness of these algorithms for many clustering problems.

As mentioned above the problem (2.2.2) is the global optimization problem and the objective function in this problem has many local minima. However, global optimization techniques are highly time-consuming for solving many clustering problems. It is very important, therefore, to develop clustering algorithms based on optimization techniques that compute "deep" local minimizers of the objective function. The

clustering algorithm proposed and studied in this chapter is of this type and is based on nonsmooth optimization techniques. The algorithm provides the capability of calculating clusters step-by-step, gradually increasing the number of data clusters until termination conditions are met, that is it allows one to calculate as many cluster as a data set contains with respect to some tolerance.

2.2.1 The nonsmooth optimization approach to minimum sum-of-squares clustering

In this section we present a formulation of the clustering problem in terms of nonsmooth, nonconvex optimization.

The problems (2.2.1) and (2.2.2) can be reformulated as the following mathematical programming problem (see [163, 27, 110, 165])

$$\text{minimize} \quad f(a^1, \ldots, a^q) \quad \text{subject to } a = (a^1, \ldots, a^q) \in \mathbb{R}^{d \times q}, \tag{2.2.3}$$

where

$$f(a^1, \ldots, a^q) = \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\ldots,q} \|a^j - x^i\|^2. \tag{2.2.4}$$

It is shown in [110] that problems (2.2.1), (2.2.2) and (2.2.3) are equivalent. The number of variables in problem (2.2.2) is $(n + d) \times q$ whereas in problem (2.2.3) this number is only $d \times q$ and the number of variables does not depend on the number of instances. It should be noted that in many real-world databases the number of instances n is substantially greater than the number of attributes d. On the other hand in the hard clustering problems the coefficients w_{ij} are integer, that is the problem (2.2.2) contains both integer and continuous variables. In the nonsmooth optimization formulation of the clustering problem we have only continuous variables. All these circumstances can be considered as advantages of the nonsmooth optimization formulation (2.2.3).

If $q > 1$, the objective function (2.2.4) in problem (2.2.3) is nonconvex and nonsmooth. If the number q of clusters and the number d of attributes are large, we have a large-scale global optimization problem. Moreover, the form of the objective function in this problem is complex enough not to be amenable to the direct

application of general purpose global optimization methods. Therefore, in order to ensure the practicality of the nonsmooth optimization approach to clustering, proper identification and use of local optimization methods is very important. Clearly, such an approach does not guarantee a globally optimal solution to problem (2.2.3). On the other hand, this approach provides a "deep" minimum of the objective function that, in turn, provides a good enough clustering description of the data set under consideration.

2.3 Discrete gradient method

In this section we will briefly describe the discrete gradient method. We start with the definition of the discrete gradient.

Definition of the discrete gradient

Definition of Lipschitz continuous function: Φ is called Lipschitz at a point x if there exist a constant $L > 0$, such that

$$|f(y) - f(z)| \leq L||y - z||$$

if

$$||y - z|| < \epsilon$$

and

$$||z - x|| < \epsilon.$$

Let Φ be a locally Lipschitz continuous function defined on $\mathbb{R}^p$. Let

$$S_1 = \{g \in \mathbb{R}^p : \|g\| = 1\}, \quad G = \{e \in \mathbb{R}^p : e = (e_1, \ldots, e_n), |e_j| = 1, \ j = 1, \ldots, p\},$$

$$P = \{z(\lambda) : z(\lambda) \in \mathbb{R}^1, \ z(\lambda) > 0, \ \lambda > 0, \ \lambda^{-1} z(\lambda) \to 0, \ \lambda \to 0\},$$

$$I(g, \alpha) = \{i \in \{1, \ldots, p\} : \ |g_i| \geq \alpha\},$$

where $\alpha \in (0, p^{-1/2}]$ is a fixed number.

Here S_1 is the unit sphere, G is the set of vertices of the unit hypercube in $\mathbb{R}^p$ and P is the set of univariate positive infinitesimal functions.

We define operators $H_i^j : \mathbb{R}^p \to \mathbb{R}^p$ for $i = 1, \ldots, p,\ j = 0, \ldots, p$ by the formula

$$H_i^j g = \begin{cases} (g_1, \ldots, g_j, 0, \ldots, 0) & if\ j < i, \\ (g_1, \ldots, g_{i-1}, 0, g_{i+1}, \ldots, g_j, 0, \ldots, 0) & if\ j \geq i. \end{cases}$$

We can see that

$$H_i^j g - H_i^{j-1} g = \begin{cases} (0, \ldots, 0, g_j, 0, \ldots, 0) & if\ \text{j=1,...,p, j} \neq i, \\ 0 & if\ \text{j=i.} \end{cases} \tag{2.3.1}$$

Let $e(\beta) = (\beta e_1, \beta^2 e_2, \ldots, \beta^p e_p)$, where $\beta \in (0, 1]$. For $y \in \mathbb{R}^p$ we consider vectors

$$y_i^j \equiv y_i^j(g, e, z, \lambda, \beta) = y + \lambda g - z(\lambda) H_i^j e(\beta),\ i = 1, \ldots, p,\ j = 0, \ldots, p.$$

It follows from (2.3.1) that

$$y_i^{j-1} - y_i^j = \begin{cases} (0, \ldots, 0, z(\lambda) e_j(\beta), 0, \ldots, 0) & if\ \text{j=1,...,d, j} \neq i, \\ 0 & if\ \text{j=i.} \end{cases} \tag{2.3.2}$$

It is clear that $H_i^0 g = 0$ and $y_i^0(g, e, z, \lambda, \beta) = y + \lambda g$ for all $i \in I(g, \alpha)$.

Definition 1 *(see [107]) The discrete gradient of the function Φ at the point $x \in \mathbb{R}^p$ is the vector $\Gamma^i(x, g, e, z, \lambda, \beta) = (\Gamma_1^i, \ldots, \Gamma_d^i) \in \mathbb{R}^p, g \in S_1, i \in I(g, \alpha)$, with the following coordinates:*

$$\Gamma_j^i = [z(\lambda) e_j(\beta)]^{-1} \left[\Phi(y_i^{j-1}(g, e, z, \lambda, \beta)) - \Phi(y_i^j(g, e, z, \lambda, \beta))\right],\quad j = 1, \ldots, p, j \neq i,$$

$$\Gamma_i^i = (\lambda g_i)^{-1} \left[\Phi(y_i^p(g, e, z, \lambda, \beta)) - \Phi(y) - \sum_{j=1, j \neq i}^{p} \Gamma_j^i (\lambda g_j - z(\lambda) e_j(\beta))\right].$$

A more detailed description of the discrete gradient and its examples can be found in [107].

It follows from Definition 1 that for the calculation of the discrete gradient $\Gamma^i(x, g, e, z, \lambda, \beta), i \in I(g, \alpha)$ we define a sequence of points

$$y_i^0, \ldots, y_i^{i-1}, y_i^{i+1}, \ldots, y_i^p.$$

For the calculation of the discrete gradient it is sufficient to evaluate the function Φ at each point of this sequence. It follows from (2.3.2) that the points y_i^{j-1} and y_i^j differ by one coordinate only $(j = 0, \ldots, p,\ j \neq i)$.

The discrete gradient is defined with respect to a given direction $g \in S_1$. We can see that for the calculation of one discrete gradient we have to calculate $(p+1)$ values of the function Φ: at the point y and at the points $y_i^0, \ldots, y_i^{i-1}, y_i^{i+1}, \ldots, y_i^p$. For the calculation of another discrete gradient at the same point with respect to any other direction $g^1 \in S_1$ we have to calculate this function p times, because we have already calculated Φ at the point y.

2.3.1 The method

We consider the following unconstrained minimization problem:

$$\text{minimize} \ \ \Phi(y) \ \ \text{subject to} \ \ y \in \mathbb{R}^p \tag{2.3.3}$$

where the function Φ is assumed to be locally Lipcshitz continuous. An important step in the discrete gradient method is the calculation of a descent direction of the objective function Φ. Therefore, we first describe an algorithm for the computation of this direction.

Let $z \in P, \lambda > 0, \beta \in (0, 1]$, the number $c \in (0, 1)$ and a small enough number $\delta > 0$ be given.

Algorithm 1 *(Bagirov' Algorithm 1)* An algorithm for the computation of the descent direction.

Step 1. Choose any $g^1 \in S_1, e \in G, i \in I(g^1, \alpha)$ and compute a discrete gradient $v^1 = \Gamma^i(y, g^1, e, z, \lambda, \beta)$. Set $\overline{D}_1(y) = \{v^1\}$ and $k = 1$.

Step 2. Calculate the vector $\|w^k\| = \min\{\|w\| : w \in \overline{D}_k(y)\}$. If

$$\|w^k\| \leq \delta, \tag{2.3.4}$$

then stop. Otherwise go to Step 3.

Step 3. Calculate the search direction by $g^{k+1} = -\|w^k\|^{-1} w^k$.

Step 4. If

$$\Phi(y + \lambda g^{k+1}) - \Phi(y) \leq -c\lambda\|w^k\|, \tag{2.3.5}$$

then stop. Otherwise go to Step 5.

Step 5. Calculate a discrete gradient

$$v^{k+1} = \Gamma^i(y, g^{k+1}, e, z, \lambda, \beta),\ i \in I(g^{k+1}, \alpha),$$

construct the set $\overline{D}_{k+1}(x) = \text{co}\{\overline{D}_k(x) \bigcup \{v^{k+1}\}\}$, set $k = k + 1$ and go to Step 2.

This algorithm was suggested by Bagirov A.M. ([106], [107], [108]). Algorithm 1 contains some steps which deserve some explanations. In Step 1 we calculate the first discrete gradient with respect to an initial direction $g^1 \in \mathbb{R}^p$. The distance between the convex hull $\overline{D}_k$ of all calculated discrete gradients and the origin is calculated in Step 2. If this distance is less than the tolerance $\delta > 0$ then we accept the point y as an approximate stationary point (Step 2), otherwise we calculate another search direction in Step 3. In Step 4 we check whether this direction is a descent direction. If it is we stop and the descent direction has been calculated, otherwise we calculate another discrete gradient with respect to this direction in Step 5 and update the set $\overline{D}_k$. At each iteration k we improve the approximation $\overline{D}_k$ of the subdifferential of the function Φ.

The discrete gradient contains some information about the behavior of the function Φ in some regions around the point y. This algorithm allows one to find descent directions in stationary points which are not local minima (descent directions in such stationary point always exist). Therefore, this method can escape from the stationary points which are not local minima and even sometimes shallow local minima

It is proved that Algorithm 1 is terminating (see [107]).

Now we can describe the discrete gradient method. Let sequences $\delta_k > 0,\ z_k \in P,\ \lambda_k > 0,\ \beta_k \in (0, 1],\ \delta_k \to +0,\ z_k \to +0,\ \lambda_k \to +0,\ \beta_k \to +0,\ k \to +\infty$ and numbers $c_1 \in (0, 1), c_2 \in (0, c_1]$ be given.

Algorithm 2 *(Bagirov' Algorithm 2)* Discrete gradient method

Step 1. Choose any starting point $y^0 \in \mathbb{R}^p$ and set $k = 0$.

Step 2. Set $s = 0$ and $y_s^k = y^k$.

Step 3. Apply Algorithm 1 for the calculation of the descent direction at $y = y_s^k, \delta = \delta_k, z = z_k, \lambda = \lambda_k, \beta = \beta_k, c = c_1$. This algorithm terminates after a finite number of iterations $l > 0$. As a result we get the set $\overline{D}_l(y_s^k)$ and an element v_s^k such that

$$\|v_s^k\| = \min\{\|v\| : v \in \overline{D}_l(y_s^k)\}.$$

Furthermore either $\|v_s^k\| \leq \delta_k$ or for the search direction $g_s^k = -\|v_s^k\|^{-1} v_s^k$

$$\Phi(y_s^k + \lambda_k g_s^k) - \Phi(y_s^k) \leq -c_1 \lambda_k \|v_s^k\|. \tag{2.3.6}$$

Step 4. If

$$\|v_s^k\| \leq \delta_k \tag{2.3.7}$$

then set $y^{k+1} = y_s^k, k = k + 1$ and go to Step 2. Otherwise go to Step 5.

Step 5. Construct the following iteration $y_{s+1}^k = y_s^k + \sigma_s g_s^k$, where σ_s is defined as follows

$$\sigma_s = \{\sigma \geq 0 : \ \Phi(y_s^k + \sigma g_s^k) - \Phi(y_s^k) \leq -c_2 \sigma \|v_s^k\|\}.$$

Step 6. Set $s = s + 1$ and go to Step 3.

The discrete gradient method starts from any initial point (Step 1) with initial values of parameters $z \in P, \ \lambda > 0, \ \beta > 0$. Then we calculate the descent direction at this point using Algorithm 1 or we find that this is an approximate stationary point (Steps 2 and 3). If this point is the approximate stationary point we update the values of the parameters to get better approximation to the subdifferential of the function Φ (Step 4). Otherwise we carry out line search in Step 5 and we continue with the same values of the parameters. The convergence of the discrete gradient method is studied in [107, 160].

2.4 Cutting angle method

In this section we consider the following problem of global optimization:

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in S \tag{2.4.1}$$

where the objective function f is an increasing positively homogeneous (IPH) of degree one and the set S is the unit simplex in $\mathbb{R}^n$:

$$S = \left\{ x \in \mathbb{R}^n_+ : \sum_{i=1}^{n} x_i = 1 \right\}.$$

Here $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x_i \geq 0, i = 1, \ldots, n\}$.

Recall that a function f defined on $\mathbb{R}^n_+$ is called increasing if $x \geq y$ implies $f(x) \geq f(y)$; the function f is positively homogeneous of degree one if $f(\lambda x) = \lambda f(x)$ for all $x \in \mathbb{R}^n_+$ and $\lambda > 0$.

For a given vector $l \in \mathbb{R}^n_+, l \neq 0$ we put $I(l) = \{i = 1, \ldots, n : l_i > 0\}$. We use the following notation for $c \in \mathbb{R}$ and $l \in \mathbb{R}^n_+$:

$$(c/l)_i = \begin{cases} c/l_i & \text{if } i \in I(l), \\ 0 & \text{if } i \notin I(l). \end{cases}$$

Note that an IPH function is nonnegative on $\mathbb{R}^n_+$. We assume that $f(x) > 0$ for all $x \in S$. It follows from positiveness of f that $I(l) = I(x)$ for all $x \in S$ and $l = f(x)/x$. Let e^k be the k-th orthant vector, $k = 1, \ldots, n$. Now we describe the cutting angle method for solving problem (2.4.1).

Algorithm 3 *(Bagirov and Rubinov's Algorithm)* see [162]

Step 0. (Initialization) Take points $x^k \in S$, $k = 1, \ldots m$, where $m \geq n$, $x^k = e^k$ for $k = 1, \ldots, n$ and $x^k_j > 0$ for $k = n+1, \ldots, m$, $j = 1, \ldots, n$. Let $l^k = f(x^k)/x^k$, $k = 1, \ldots, m$. Define the function h_m :

$$h_m(x) = \max_{k \leq m} \min_{i \in I(l^k)} l^k_i x_i = \max \left\{ \max_{k \leq n} l^k_k x_k, \max_{n+1 \leq k \leq m} \min_{i \in I(l^k)} l^k_i x_i \right\} \tag{2.4.2}$$

and set $j = m$.

Step 1. Find a solution x^* of the problem

$$\text{minimize} \quad h_j(x) \quad \text{subject to} \quad x \in S. \tag{2.4.3}$$

Step 2. Set $j = j + 1$ and $x^j = x^*$.

Step 3. Compute $l^j = f(x^j)/x^j$, define the function

$$h_j(x) = \max\{h_{j-1}(x), \min_{i \in I(l^j)} l_i^j x_i\} \equiv \max_{k \le j} \min_{i \in I(l^k)} l_i^k x_i \tag{2.4.4}$$

and go to Step 1.

More detailed description of Algorithm 3 with necessary explanations can be found in [161, 162, 190]. This algorithm can be considered as a version of the cutting angle method ([158, 159]). The cutting angle method provides a sequence of lower estimates for the global minimum f_* of (2.4.1) with an IPH objective function, which converges to f_*. Theoretically this sequence can be used for establishment of a stopping criterion (see [190] for details). Let

$$\lambda_j = \min_{x \in S} h_j(x) = h_j(x^{j+1}) \tag{2.4.5}$$

be the value of the problem (2.4.3). λ_j is a lower estimate of the global minimum f_*. It is known (see, for example, [190]), that λ_j is an increasing sequence and $\lambda_j \to f_*$ as $j \to +\infty$.

The cutting angle method constructs the sequence $\{f(x^j)\}$, which is not necessarily decreasing: it is possible that $f(x^{j+1}) > f(x^j)$ for some j.

The most difficult and time-consuming part of the cutting angle method is solving the auxiliary problem (2.4.3). An algorithm for the solution of this problem was proposed in [161]. Some modifications of this algorithm (and corresponding modifications of the cutting angle method) are discussed in [162] and [164].

Only one value of the objective function is used at each iteration of the cutting angle method. Some modifications of this method require to evaluate a few values of the objective function at each iteration.

Global minimization of Lipschitz functions

Now we consider the following problem of global optimization:

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in S \tag{2.4.6}$$

where the function f is Lipschitz continuous on S. This problem can be reduced to the global minimization of a certain IPH function over S. The following theorem has been established in [191] (see [190]).

Theorem 1 *Let $f : S \to \mathbb{R}$ be a Lipschitz function and let*

$$L = \sup_{x,y \in S, x \neq y} \frac{|f(x) - f(y)|}{\|x - y\|_1} \tag{2.4.7}$$

be the least Lipschitz constant of f in $\|\cdot\|_1$-norm, where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. Consider a positively homogeneous function

$$\varphi(x) = \begin{cases} \|x\| f(\frac{x}{\|x\|}) & if \ x \neq 0, \\ 0 & if \ x=0 \end{cases}$$

defined on $\mathbb{R}^n_+$. If $\min_{x \in S} f(x) \geq 2L$ then φ is an IPH function and $\varphi(x) = f(x)$ for all $x \in S$.

Let

$$c \geq 2L - \min_{x \in S} f(x), \tag{2.4.8}$$

where L is defined by (2.4.7). Let $f_1(x) = f(x) + c$. Theorem 1 implies that the function f_1 can be extended to an IPH function φ. Since $f_1(x) = \varphi(x)$ for all $x \in S$ the global minimization of φ over S is equivalent to the following global optimization problem:

$$\text{minimize} \quad f_1(x) \quad \text{subject to} \quad x \in S \tag{2.4.9}$$

and consequently the cutting angle method can be applied to solve this problem. On the other hand the functions f and f_1 have the same minimizers on the simplex S and if the constant c is known the problem (2.4.6) can be solved by the cutting angle method. In order to estimate c we need to know an upper estimation of the Lipschitz constant L and a lower estimation of the desired global minimum of the function f. We will assume that c is a sufficiently large number. However it should be noted that for increasing values of c the cutting angle method works less efficiently.

Chapter 3

Stochastic programming models —Corrective maintenance

In this chapter we will study stochastic programming models for maintenance. We will introduce a known stochastic programming based maintenance model and solve it with a direct method and develop some new ones based on this model.

3.1 Network management

Several studies have shown that network management can take as much as 40% of the total communication budget. As an important part of network management, maintenance service, exists to both keep equipment in running order and to reduce the number of breakdowns. Breakdowns can be very expensive. For example, the nine hour breakdown of AT&T's long-distance telephone network in January 1990 resulted in a $60 million to $75 million loss in AT&T's revenues [8]. The challenge of creating a cost-effective network maintenance policy is often complicated by the difficulty of establishing a practical model. To ensure that the network management methodology, and the tools that may be selected by an organization, provide for efficient network operations, we need to establish some models to optimize the network management and maintenance.

This chapter contains some models which reduce the problem under consideration

to a certain optimization problem. We start with the so-called simplified model (SM model) [1] in this section. Although this model is simple, it gives the main idea of optimization in telecommunications networks maintenance. Normal distribution was used in paper [1]. Our numerical experiments demonstrate however, that this distribution is not always adequate. We introduce a new version of the SM model, which is based on consideration of binomial and Poisson distributions. Using this way we conducted a number of numerical experiments, which show that we can get some basic idea about how to decide the number of service persons and the number of hub duplicates. It also shows that the SM model with a different distribution of the number of hub failures may result in a different maintenance policy.

3.2 What is maintenance?

The objective of maintenance is to bring whatever is being maintained towards a state of failure-free operation. This, however, is not to be understood as implying 'at whatever cost'. Rather, the aim is to find the best situation, taking into account the increasing costs of increasingly sophisticated maintenance, as well as costs resulting from increasingly high failure rates [11]. This illustrated in Figure 3.1.

There are two main types of maintenance:

Preventive maintenance aims to reduce the probability of failure; this breaks into two sub-types:

- systematic or scheduled maintenance, in which specified components are replaced (usually at regular intervals) when they are becoming worn;
- condition-based maintenance, in which the decision to replace or not to replace is made according to the outcome of a diagnostic study.

Corrective maintenance is used only after a failure. This does not necessarily mean that such action has not been foreseen; in fact, with the aid of a maintenance tree, methods for quick recovery from failure can be developed. This structure can be shown diagrammatically as in Figure 3.2.

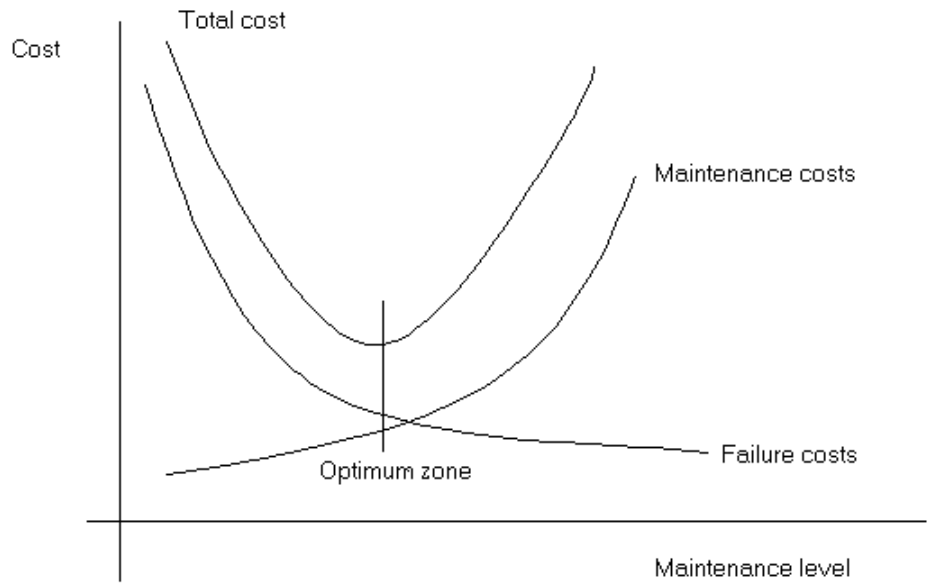

Figure 3.1: Maintenance and failure cost

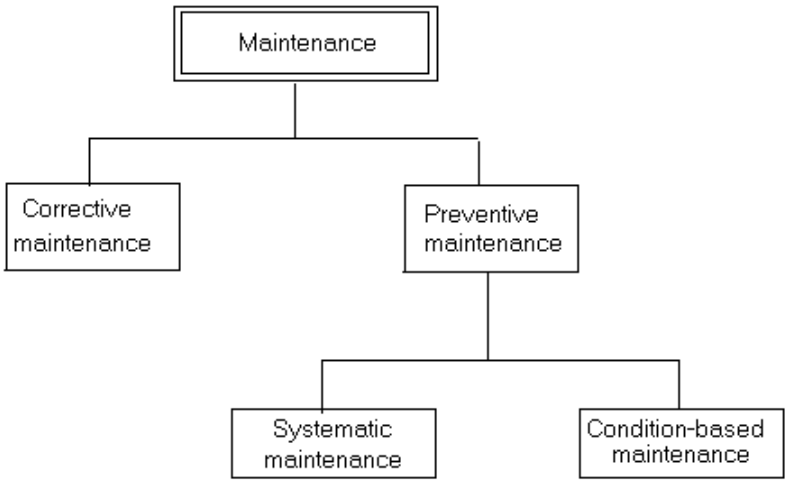

Figure 3.2: Maintenance policies

Breakdowns

A maintenance service exists to keep equipments in running order and also to reduce the number of breakdowns. Breakdowns can be expensive, giving rise to costs for a number of reasons:

- The need for intervention in the processes of data, voice, and some other important communications.
- Investigation and repair;
- Lowered quality of the service;
- Indirect costs such as: failure to meet fixed costs;
- variable costs not otherwise allowed for;
- extra costs incurred in compensating for loss of service;
- reduced profit margin.

In telecommunications the breakdowns are very expensive. Telecommunications network should provide the whole-hour (24-hours per day, 7-days per week) service. Any breakdown can be disastrous. To minimize the breakdowns is one of the most important factors to improve the service level.

Reliability

The reliability is the fraction of time the system is working. This is also sometimes referred to as availability. This takes into account the mean time between failures (MTBF) and the mean time to repair (MTTR). Thus, the reliability (R) of a system is

$$R = 1 - \frac{MTTR}{MTBF}. \tag{3.2.1}$$

3.3 Stochastic programming

Stochastic programming approaches modelling optimization problems that involve uncertainty. Whereas deterministic optimization problems are formulated with known parameters, real world problems almost invariably include some unknown parameters. When the parameters are known only within certain bounds, one approach to tackling such problems is called robust optimization. Here the goal is to find a solution which is feasible for all such data and optimal in some sense. Stochastic programming models are similar in style, but take advantage of the fact that probability distributions governing the data are known or can be estimated. The goal here is to find some policy that is feasible for all (or almost all) the possible data instances and maximizes the expectation of some function of the decisions and the random variables. More generally, such models are formulated, solved analytically or numerically, and analyzed in order to provide useful information to a decision-maker.

In general, the problem of stochastic programming can be formulated as follows:

$$\text{minimize} \quad f(x, \theta)$$

$$\text{subject to}$$

$$x \in X, \quad g_i(x, \theta) \leq 0, \quad i = 1, \ldots, m.$$

Here $X \subset \mathbb{R}^n$ is a set with simple structure and $\theta \in \mathbb{R}^m$ is a random vector with a certain distribution. This vector describes uncertainty.

Stochastic programming was introduced in 1955 by Dantzig [26]. Since then stochastic programming models have been developed for a variety of applications, including electric power generation (Murphy et al [148]), financial planning (Carino et al [145]), telecommunications network planning (Sen et al [150]), and supply chain management (Fisher et al [146]), to mention a few. The widespread applicability of stochastic programming models has attracted considerable attention from the OR/MS community, resulting in several recent books (Kall and Wallace [147], Birge and Louveaux [144], Prekopa [149]) and survey articles (Birge [143], Sen and Higle [151]). Nevertheless, stochastic programming models remain one of the more challenging optimization problems.

3.4 A brief review of relevant previous research

The concept of maintainability first appeared in 1954 in the US army. Since then a variety of techniques and methodologies have been developed and implemented to meet the challenge of the new needs.

In W. Feller's paper [18](1957), Feller first published the analytic solutions of the $M|S|R$ machine-repair problem, which consists of of M operating machines with S spares, and R repair persons under steady-state conditions. George H. Weiss was the first to consider a single-unit system with inspection [19] (1962). This research was supported by the United States Air Force. Later, many papers about the cost analysis were published [20][21][22].

The authors of paper [4] studied the $M|M|R$ machine repair problem (instead of hub repair problem). Unlike the hub repair problem, where each duplicate (or spare) is related to one particular hub, the fleet of S spares in this $M|M|R$ machine repair problem are considered to be cold-standby, or warm-standby, or hot-standby, and can replace any failed operating machines. The operating machines have the same failure rates. The spares have different failure rates according to their types: the failure rate of cold standby spares is zero, and the failure rate of warm-standby spares is lower than the failure rate of the operating machine. A model minimizing the sum of costs (loss of profit, idling cost, standby cost and repair cost) subject to maximum system availability. System availability is expressed in terms of steady state probabilities of at least M machines operating. Wang [5] then expanded this model to determine the optimal number of repair persons, warm standbys, and cold standbys. A model to determine the optimal number of repair persons and cold standbys in a system where machines have K failure scenario was subsequently studied by Wang and Lee [6].

Unfortunately the research on maintenance optimization has been absent from many telecommunications networks optimization papers. A paper [1] by H. S. Gan and A. Wirth presented some efficient models. This paper contains some models which reduce the problem under consideration to a certain optimization problem. This is based on the paper [3] by Soo and Truong, who used a queuing approach to model this problem.

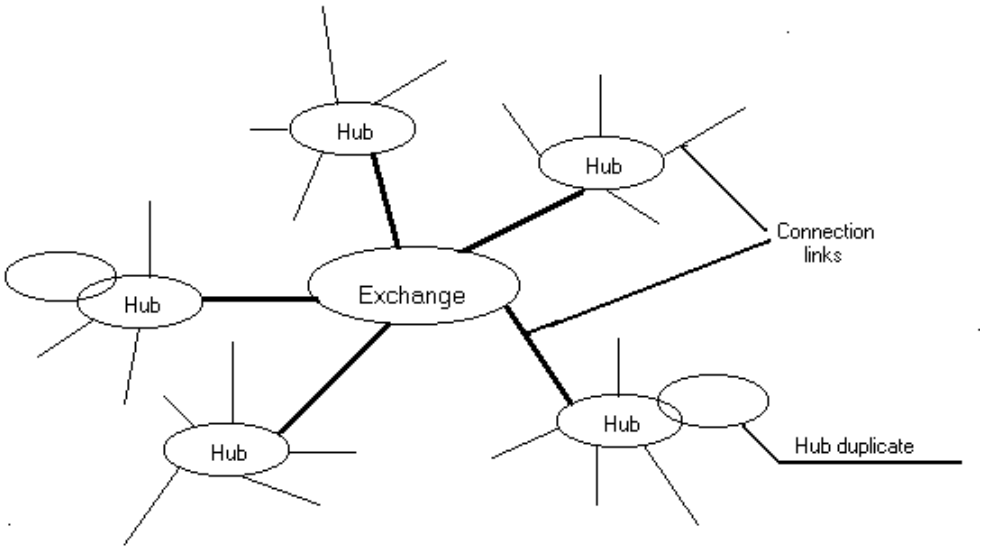

Figure 3.3: A star network configuration

Lastly we want to emphasize the key difference between machine repair problems and hub repair problems, which lies in the different distribution of hubs and machines. Hubs are geographically distributed but the machines are generally installed in the same place. Another difference is hubs should work in whole hours, days and years.

3.5 Some elements of telecommunications network

Telecommunications networks exist to provide communication channels between points, which may be geographically close or, may be separated by worldwide distance. A telecommunications network is generally comprised of the following major components: exchanges, cables, power supplies, switches, routers, hubs, ports, and lots of software. In our work, for the sake of simplicity, initially we consider a star topology for network configuration Figure 3.3.

Exchange: A telecommunications switching center that supplies telecommunications services to a specific geographic area.

Hub: A remote concentration unit, which connects the users to the exchange. This

hub will connect a small proportion of population to the exchange to reduce the capital cost of the network.

Original Hub: An original hub is the hub in service.

Hub duplicate: A hub duplicate is a stand-by hub. When an original hub fails, a hub duplicate will be a substitute for the original hub whilst it is being serviced or repaired.

Repair person: A repair person services the malfunctioning hubs.

Connection links: Means of connection, which link up the exchange to the hubs and hubs to users.

We would like to emphasize that the hub in our thesis can represent other telecommunications facilities, such as routers and bridges.

3.6 SM model

We now present the SM model from [1]. Assume there are n hubs in the network and let one time unit be the time for one repair person to repair one malfunctioning hub.

Our objective is to construct a cost-effective telecommunications network maintenance policy, which consists of a crew of repair persons and a fleet of hub duplicates. The cost structure will include the salary of the repair persons and the cost of duplication. We define the following parameters:

α — salary of one repair person per unit time;

c_i — cost of one duplicate hub i in per unit time;

p_i — probability of failure of original hub i per unit time;

p_{di} — probability of failure of hub duplicate i per unit time.

c_l — the criterion of the level of service.

X — the number of malfunctioning hubs.

We shall use the following variables

s — total number of repair persons employed; s is the nonnegative integer;

y_i— indicates the duplication of the i-th hub:

$$y_i = \begin{cases} 1 & \text{if hub } i \text{ is not duplicated,} \\ 0 & \text{otherwise.} \end{cases}$$

The total number of duplicated and non-duplicated hubs will be, respectively:

$$y^d = n - y, \qquad y = \sum_{i=1}^{n} y_i.$$

It is assumed that the network must be maintained.

The total cost (TC) function to maintain the network (per unit time), can be defined as follows:

$$TC(s, y_1, y_2, , , y_n) = \alpha s + \sum_{i=1}^{n} c_i(1 - y_i).$$

If we consider the hubs and duplication costs are all the same, that is $c_i = c, \forall i$. Then we can simplify the above formula as:

$$TC(s, y) = \alpha s + c(n - y).$$

The first term here is the total salary of the crew of service persons and the second term gives the total hub duplication cost. It should be noted that the terms duplicate (or hub duplicate) and duplicated hub both have very distinct meanings. The term duplicate (or hub duplicate) refers to a standby hub, whereas the term duplicated hub refers to a hub where, at its corresponding location, a standby hub (hub duplicate) is installed.

The following assumptions have been made in [1] (explicitly or implicitly):

- The exchange is
  a. too expensive to be duplicated, and therefore will not be duplicated
  b. ideal, i.e. has zero probability of failure, $p_e = 0$.

- All original hubs

  (a.) will have the same amount of repair time.

  (b.) have the same probability of failure per unit time: $p_i = p$.

  (c.) are located equidistant from the exchange.

- A simple, but very general model is the stochastic binary system. Each hub can take on either of two states: operative or failed.

- The hub duplicates are ideal with $p_{di} = 0$, which guarantee the hub duplicate will work instead of the failed original hub. Each hub can only have one duplicate.

- The duplication costs are the same for every hub, i.e. $c_i = c$.

- The crew of repair persons is stationed at the exchange.

- There will be no absenteeism in the crew of repair persons and their skills are homogenous, i.e. the total service time taken by any repair person servicing a malfunctioning hub is equal.

- Only one repair person is needed to service one malfunctioning hub, regardless of the type of failure.

- In a queue of malfunctioning duplicated and non-duplicated hubs, the non-duplicated hubs will be given the highest service priority. Here we assume that there is no switch over time from malfunctioning hub to hub duplicate.

Following the assumptions above, an example of the network architecture for SM is illustrated in Figure 3.4.

And the SM model can be defined as:

$$\text{minimize } TC(s, y) \equiv \alpha s + c(n - y) \tag{3.6.1}$$

subject to

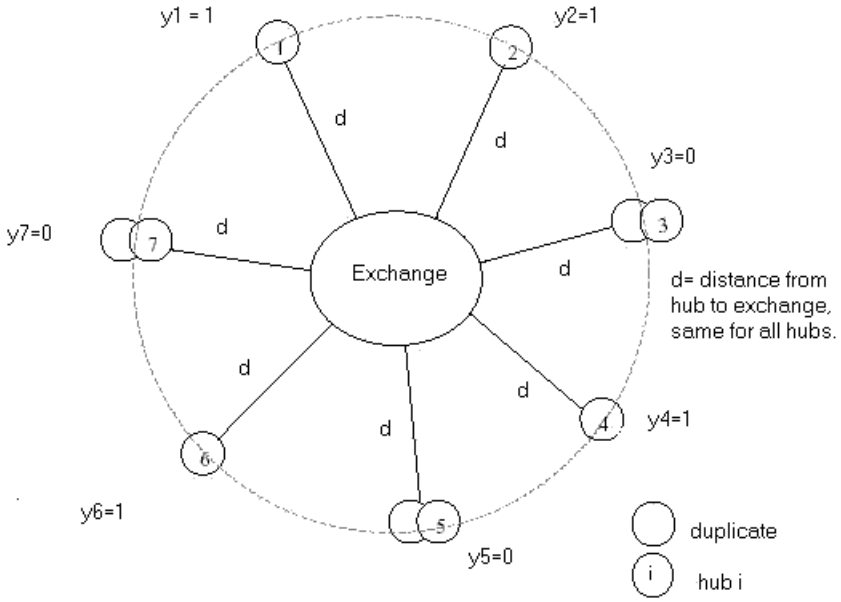

Figure 3.4: Network architecture for the simplified model (SM)

$$P(X > s) \leq c_l. \tag{3.6.2}$$

$P(X > s)$ is the probability of not enough repair men to fix the malfunctioning hubs. The main idea of the SM model is to minimize the cost of maintenance (3.6.1) under the constraint (3.6.2): the probability of not enough repair persons to repair malfunctioning hubs should be less or equal to c_l. Here X is the number of malfunctioning hubs, and s is the number of service persons. The level of service can be measured by different c_l. To finish a setting of the model we need to consider a certain distribution of failures of both non-duplicated and duplicated hubs per unit time.

3.7 SM model with normal distribution (SMN)

The following assumption has been made in [1]. The number of failures of non-duplicated hubs per unit time is normally distributed with mean $\mu(y)$ and standard deviation $\sigma(y)$. (See Figure 3.5).

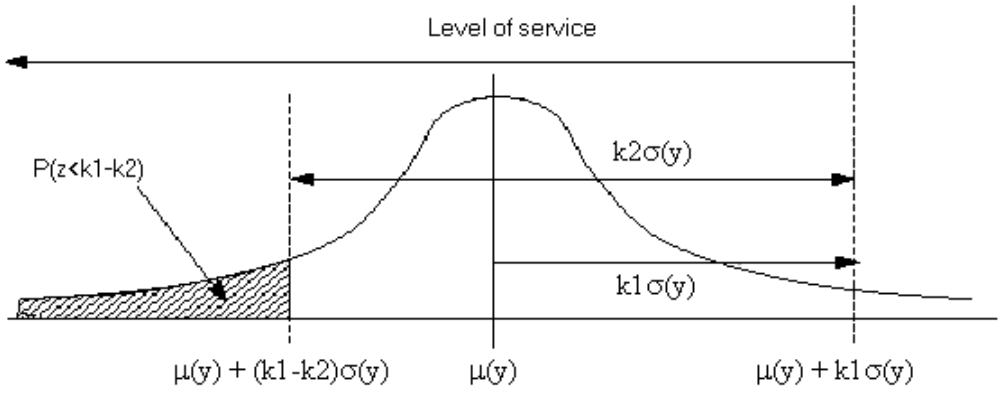

Figure 3.5: The normal distribution of the number of failures

As illustrated in Figure 3.5, the level of service provided to repair malfunctioning non-duplicated hubs is $\mu(y) + k_1\sigma(y)$ and the malfunctioning duplicated hubs will only be repaired when there are $k_2\sigma(y)$ repair persons available.

In Figure 3.5 we assume that the number of failures (random variable x) of non-duplicated hubs per unit time is normally distributed with the mean $\mu(y)$, and the standard deviation $\sigma(y)$. We denote that z is the standard score for variable x, that is

$$z = \frac{(x - \mu(y))}{\sigma(y)}.$$

It presents the number of $\sigma(y)$ away from the mean. The normal curve has been analyzed in terms of the area of the curve that is defined by various values of z. For example, about 68% of the area under a normal curve lies within one standard deviation of the mean, that is between $z = \pm 1$. And about 95% of the area lies within two standard deviations of the mean-actually between $z = \pm 1.96$. In other words, the probability that variable x has a value between $\mu(y) \pm 1.96\sigma(y)$ is 0.95. So in Figure 3.5, $x = \mu(y) + k_1\sigma(y)$ can be described as $z = k_1$; and $x = \mu(y) + (k_1 - k_2)\sigma(y)$ is equivalent to $z = (k_1 - k_2)$. In this case k_1 can be used to measure the level of service provided to repair malfunctioning non-duplicated hubs, as is illustrated in Figure 3.5. If the probability of not enough repair persons to repair malfunctioning non-duplicated hubs is less than 0.00001, we find the value of k_1 from the z table to be as 5.9605. In other words, from formula (3.6.2) if we choose $c_l = 0.00001$, the

following expressions:

$$P(X > s) < c_l$$

and

$$\mu(y) + k_1\sigma(y) \leq s$$

are the same thing when we choose $c_l = 0.00001$ and $k_1 = 5.9605$.

Although the hub duplicate can start work automatically when the duplicated hub fails to work, we also need to repair it and make it a hub duplicate in turn. To make it visual and easy to calculate, we denote the number of repair persons for malfunctioning duplicated hubs as $k_2\sigma(y)$. Figure 3.5 also shows the relationship between k_1 and k_2.

Assuming the independency of the probabilities of hub failures, then $\mu(y)$ and $\sigma(y)$ can be expressed as follows,

$$\mu(y) = \sum_{i=1}^{n} y_i p_i, \tag{3.7.1}$$

$$\sigma(y) = \sqrt{\sum_{i=1}^{n} y_i p_i (1 - p_i)}. \tag{3.7.2}$$

Since $p_i = p$ and $\sum_{i=1}^{n} y_i = y$, so we get

$$\mu(y) = yp, \tag{3.7.3}$$

$$\sigma(y) = \sqrt{yp(1-p)}. \tag{3.7.4}$$

The repair of malfunctioning duplicated hubs will only be performed when there are $k_2\sigma(y)$ repair persons available. The proportion of time when at least $k_2\sigma(y)$ repair persons available is $P(z < k_1 - k_2)$. Hence, the level of service for the duplicated hubs constraint can be formulated as follows,

$$P(z < k_1 - k_2)k_2\sigma(y) \geq p(n - y).$$

Here $p(n - y)$ is the mean of the number of failures of duplicated hubs per unit of time, it is also assumed to be normally distributed. In other words, this

constraint specifies that we will only provide repair service for the "average" number of malfunctioning duplicated hubs.

So, the SM with normal distribution (SMN) is formulated below:

$$\text{minimize } TC(s,y) \equiv \alpha s + c(n-y) \tag{3.7.5}$$

subject to

$$\mu(y) + k_1\sigma(y) \leq s, \tag{3.7.6}$$

$$P(z < k_1 - k_2)k_2\sigma(y) \geq p(n-y), \tag{3.7.7}$$

$$k_2\sigma(y) \leq s, \tag{3.7.8}$$

$$y \leq n, \tag{3.7.9}$$

$$s, y \geq 0, \tag{3.7.10}$$

$$s, y \text{ are integers.} \tag{3.7.11}$$

With normal distribution, constraint (3.6.2) can be expressed as (3.7.6)—(3.7.8). Constraint (3.7.6) ensures the level of service for non-duplicated hubs is met by the allocation of a sufficient number of repair persons. Constraints (3.7.7) and (3.7.8) ensure that the level of service of duplicated hubs is fulfilled and constraint (3.7.9) specifies that the total number of duplications must not exceed the total number of hubs in the network. The non-negativity constraints are in (3.7.10,3.7.11).

3.8 Direct method to solve the SMN

3.8.1 The KKT method

In paper [1] the dual approach to the solution of the problem (3.7.5)– (3.7.10) based on the Karush-Kuhn-Tucker conditions was suggested.

By transforming (3.7.5) into a maximization problem and rearranging equations (3.7.6–3.7.10) to a "less than or equal to" constraints, and letting $y = w^2$ and $k_3 = P(z < k_1 - k_2)k_2$, we obtain the following,

$$\text{maximize } TC \equiv (cw^2 - \alpha s - cn) \tag{3.8.1}$$

subject to

$$pw^2 + k_1\sqrt{p(1-p)}w - s \leq 0, \tag{3.8.2}$$

$$-k_3\sqrt{p(1-p)}w - pw^2 \leq -np, \tag{3.8.3}$$

$$w \leq \sqrt{n}, \tag{3.8.4}$$

$$-w \leq 0, \tag{3.8.5}$$

$$k_2\sqrt{p(1-p)}w - s \leq 0. \tag{3.8.6}$$

The KKT conditions for (3.8.1–3.8.6) are expressed as follows:

$$2cw - \lambda_1(2pw + k_1\sqrt{p(1-p)}) - \lambda_2(-k_3\sqrt{p(1-p)} - 2pw) - \lambda_3 - \lambda_4 - \lambda_5(k_2\sqrt{p(1-p)}) = 0, \tag{3.8.7}$$

$$-\alpha + \lambda_1 + \lambda_5 = 0, \tag{3.8.8}$$

$$\lambda_1(-pw^2 - k_1\sqrt{p(1-p)}w + s) = 0, \tag{3.8.9}$$

$$\lambda_2(-np + k_3\sqrt{p(1-p)}w + pw^2) = 0, \tag{3.8.10}$$

$$\lambda_3(\sqrt{n} - w) = 0, \tag{3.8.11}$$

$$\lambda_4 w = 0, \tag{3.8.12}$$

$$\lambda_5(s - k_2\sqrt{p(1-p)}w) = 0. \tag{3.8.13}$$

H. S. Gan and A. Wirth investigated all the possible combinations of λ_1, λ_2, λ_3, λ_4, and λ_5. This approach can not explicitly take into account that s and y are integers. We suggest a direct approach to solving this problem.

3.8.2 The direct method

Studying the SM with the normal distribution, we find out the following procedure to solve the SMN.

Let

$$P(z < k_1 - k_2)k_2 = k_3. \tag{3.8.14}$$

Then (3.7.7) can be represented as

$$k_3\sqrt{yp(1-p)} \geq p(n-y). \tag{3.8.15}$$

1. We can solve this inequality explicitly in the following way:

from (3.8.15) we get

$$k_3{}^2yp(1-p) \geq p^2(n-y)^2.$$

Since $p > 0$ this inequality is equivalent to:

$$py^2 - [2np + {k_3}^2(1-p)]y + pn^2 \leq 0.$$

The solution set of this inequality is a segment, we denote this segment by $[a, b]$. Let u_1, u_2 be roots of equation:

$$py^2 - [2np + {k_3}^2(1-p)]y + pn^2 = 0. \tag{3.8.16}$$

We have:

$$u_1 = \frac{(h-m)}{2p},$$

and

$$u_2 = \frac{(h+m)}{2p},$$

where

$$h = 2np + {k_3}^2(1-p),$$

and

$$m = \sqrt{(2np + {k_3}^2(1-p))^2 - 4p^2n^2}.$$

It is easy to see that $u_2 > u_1 > 0$, so we can choose $a = u_1, b = u_2$.

2. Since the system of simultaneous inequalities (3.7.6) and (3.7.8) is equivalent to:

$$\max(\mu(y) + k_1\sigma(y), k_2\sigma(y)) \leq s,$$

then for each y, we consider the function $s(y)$ defined by

$$s(y) = \max[yp + k_1\sqrt{yp(1-p)}, k_2\sqrt{yp(1-p)}] \tag{3.8.17}$$

where $y \in [a, b]$.

3. Let

$$f(y) = \alpha s(y) + c(n-y),$$

Then the problem (3.7.5) -(3.7.10) is equivalent to the following problem:

$$\text{minimize} f(y)$$

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
TC	968	1753	2446	3042	4234	5428	6424	7519	8317	9213	9909
s	9	17	24	30	42	54	64	75	83	92	99
y^d	68	53	46	42	34	28	24	19	17	13	9

$\frac{c}{\alpha} = 0.01$, n=100, normal distribution.

Table 3.1: Result one from direct method

subject to

$$y \in [a, b], y \text{ is integer.} \tag{3.8.18}$$

To solve this problem we need to find all integers y between numbers a and b, calculate values $f(y)$ and find the least of them.

The code was written in C. Numerical experiments were carried out on PC Pentium II with 400 MHZ main processor. In this code we considered different values of k_2, $\frac{c}{\alpha}$, and p. We made some calculations and compared some of our results with those from [1]. We obtained the same results that confirm that both the dual approach from [1] and the direct approach from the current paper are correct.

Tables 3.1-3.2 describe the results obtained by using the direct method. We use $k_2 - k_1 = 0.15$, $n = 100$, $k_1 = 5.9605$, $\alpha = 100$, and $\frac{c}{\alpha} = 0.01$, 1.0, respectively. In these tables, p is the probability of failure of hub per unit time. TC is the total cost of the salary of the repair persons and the cost of duplication in per unit time, and s is the total number of repair persons needed. Here $y^d = n - y$ represents the total number of hub duplicates. Table 3.3 presents the results of different value of n in this model. Comparing these three tables, we get the trend: TC increases as p increases, so does the $\frac{c}{\alpha}$. It is also evident from the formulations: it costs more to maintain a more unreliable network; when the salary of repair person is the same, the greater the cost of duplication the greater the TC. Of course, when the cost of duplication is too much, reducing or eliminating duplication will be the right choice (see Table 3.2).

The choice of k_2 is also very important in this model. Tables 3.4-3.5 summarize the influence of different choices of k_2. In Table 3.4, $\frac{c}{\alpha} = 0.1$, $n = 100$, $k_1 = 5.9605$,

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
TC	1800	2800	3700	4400	5800	7000	8000	9000	9800	10400	10800
s	18	28	37	44	57	69	80	87	94	103	99
y^d	0	0	0	0	1	1	0	3	4	1	9

$\frac{c}{\alpha}$ =1.0, n=100, normal distribution.

Table 3.2: Result two from direct method

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
TC	7205	13020	18580	23957	34621	44997	55081	64967	74655	84142	93128
s	69	128	184	238	345	449	550	649	746	841	931
y^d	305	220	180	157	121	97	81	67	55	42	28

$\frac{c}{\alpha}$ = 0.01, n=1000, normal distribution.

Table 3.3: Result with different n

$k_2 - k_1$	-1.05	-0.75	-0.45	-0.15	0.15	0.45	0.75	1.05	1.50	1.95	3.00
TC	2000	2060	2060	2180	2230	2340	2450	2550	2700	2750	2800
s	13	14	14	16	17	19	21	23	26	27	28
y^d	70	66	66	58	53	46	35	25	10	5	1

$\frac{c}{\alpha}$ = 0.1, p= 0.1, n=100, different k_2.

Table 3.4: Result with different k_2, $p = 0.1$

$k_2 - k_1$	-1.05	-0.75	-0.45	-0.15	0.15	0.45	0.75	1.05	1.50	1.95	3.00
TC	8890	8890	9070	9150	9330	9600	9780	9960	10230	10310	10310
s	87	87	91	89	90	95	97	99	102	103	103
y^d	19	19	17	15	14	10	8	6	3	1	1

$\frac{c}{\alpha}$ = 0.1, p= 0.8, n=100, different k_2.

Table 3.5: Result with different k_2, p= 0.8

$p = 0.1$ and $\alpha = 100$. In Table 3.5, $\frac{c}{\alpha} = 0.1$, $n = 100$, $k_1 = 5.9605$, $p = 0.8$ and $\alpha = 100$. So we get the conclusions: first, as k_2 increases TC increases. Second, if k_2 and p are large enough, then s will be decided by $k_2\sigma(y)$ according to formulation (3.7.7) (when $k_2 > k_1 + \frac{yp}{\sqrt{yp(1-p}}$). In this case it contradicts the assumption that the non-duplicated hubs will be given the highest service priority, and it also leads to an unreasonable answer: 103 service persons are needed to service one hub duplicate and 100 hubs (see Table 3.2). The same result can be seen in Table 3.5. In fact we can simply choose no duplication and 100 service persons and get TC= 10000. This is the weak point in this model: when the value of p is too large and n is not big enough, the normal distribution does not approach the real distribution (binomial distribution) and $\mu(y) + k_1\sigma(y)$ may be more than n. When $n > 500$ and $p < 0.95$ this does not happen (see Table 3.3). We have some doubts about SMN with a small value of p because of its application. We also need to clarify the limitation for normal distribution, because it does not always work. That is why we tried the following approach with different distributions.

3.9 Model with Poisson and binomial distribution (SMP and SMB)

We now consider the SM model with binomial (SMB) and Poisson (SMP) distributions respectively. In fact the binomial distribution is the real distribution in stochastic binary, but in telecommunications we usually use Poisson distribution, since it describes queueing systems.

Some definitions of new variables and parameters in SMB and SMP:

s_1 — the number of service persons needed to fix the malfunctioning non-duplicated hubs.

s_2 — the number of service persons needed to fix the malfunctioning duplicated hubs.

s_3 — the number of service persons needed to fix the failures of duplicated hubs and duplicate hubs happen simultaneously.

s — the total number of service persons to fix all kinds of above failures.

We introduce following parameters:

λ_1 — the mean of Poisson random variable related to non-duplicated hubs.

λ_2 — the mean of Poisson random variable related to duplicated hubs.

c_{l1} — the criterion of the level of service for the malfunctioning non-duplicated hubs.

c_{l2} — the criterion of the level of service for the malfunctioning duplicated hubs.

c_{l3} — to adjust the criterion of the level of service for the malfunctioning duplicated hubs.

c_{l4} — to produce different p_{di} $(=c_{l4}p)$.

For Poisson distribution we have (SMP)

$$\text{minimize } TC(y,s) \equiv \alpha s + c(n-y), \tag{3.9.1}$$

subject to

$$P(X > s_1) = 1 - \sum_{k=0}^{s_1} \frac{\lambda_1^k e^{-\lambda_1}}{k!} < c_{l1}, \tag{3.9.2}$$

$$s_1 \geq \mu_1, \tag{3.9.3}$$

$$P(X > s_2) = 1 - \sum_{k=0}^{s_2} \frac{\lambda_2^k e^{-\lambda_2}}{k!} < c_{l2}, \tag{3.9.4}$$

$$s_2 \geq \mu_2, \tag{3.9.5}$$

$$s_3 = \lceil (n-y)ppc_{l4} \rceil, \tag{3.9.6}$$

$$s = s_1 + c_{l3}s_2 + s_3, \tag{3.9.7}$$

$$y, s \leq n, \tag{3.9.8}$$

$$s, y \in N.$$

N is the set of all nonnegative integers.

(3.9.9)

Here we use $\mu_1 = y$ and $\mu_2 = (n - y)p_{di}$. The parameters c_{l1} to c_{l3} are used as a measure of the service levels in our model. In constraint (3.9.2) the probability of not enough service persons to repair the malfunctioning non-duplicated hubs should be less than c_{l1}, similar to constraint (3.9.4) where we also assume that the number of failures of duplicated hubs is Poisson distributed. In constraint (3.9.3) we use the mean of the number of failures of non-duplicated hubs as the lowest boundary of the service level. In other words, there are at least enough service persons to fix at least the mean malfunctioning non-duplicated hubs in unit time. The same is the constraint (3.9.5). From the formulation of the Poisson distribution we have

$$\lambda_1 = yp,$$

and

$$\lambda_2 = (n - y)p_{di}.$$

In this model p_{di} is the probability of the number of failures of hub duplicates per unit time. In general $p_{di} < p$. To simplify the calculation, we can assume $p_{di} = p$. Since one can take more time to repair the malfunctioning duplicated hubs (we also assume that the switch over time is zero), sometimes one service person can fix the malfunctioning duplicated hubs after fixing the malfunctioning non-duplicated hubs. In equation (3.9.7) we use c_{l3} to adjust the number of service persons in this case. To make the model more reliable, we also consider the case of the failure of duplicated hubs and duplicate hubs happen simultaneously. In equation (3.9.6), we use c_{l4} to produce different $p_{di}(=c_{l4}p)$, so, the variable s_3 is the ceiling of the mean of the failure number of hubs and duplicate hubs happening simultaneously.

For binomial distribution model (SMB)

$$\text{minimize } TC(y, s) \equiv \alpha s + c(n - y), \tag{3.9.10}$$

subject to

$$P(X > s_1) = 1 - \sum_{k=0}^{s_1} \begin{pmatrix} y \\ k \end{pmatrix} p^k (1-p)^{y-k} < c_{l1}, \tag{3.9.11}$$

$$s_1 \geq \mu_1 (= yp), \tag{3.9.12}$$

$$P(X > s_2) = 1 - \sum_{k=0}^{s_2} \begin{pmatrix} n-y \\ k \end{pmatrix} (p_{di})^k (1-(p_{di}))^{n-y-k} < c_{l2}, \tag{3.9.13}$$

$$s_2 \geq \mu_2 (= (n-y)p_{di}), \tag{3.9.14}$$

$$s_3 = \lceil (n-y)ppc_{l4} \rceil, \tag{3.9.15}$$

$$s = s_1 + c_{l3} s_2 + s_3, \tag{3.9.16}$$

$$y, s \leq n. \tag{3.9.17}$$

All the meanings of constraints are similar to SMP. The codes for SMP and SMB were also written in C. Numerical experiments were carried out on PC Pentium II with 400 MHZ main processor as well. To solve this problem we needed to try all integers y between the numbers of 0 to n, then calculate the values of s_1, s_2, s_3 and s respectively, and finally calculate values of $TC(y, s)$ and find the least of them.

3.9.1 The result and comparison

In the model of SMP, we chose $c_{l1}, c_{l2} = 0.009$ and $c_{l3}, c_{l4} = 0.3$ and $\frac{c}{\alpha} = 0.01$, we obtained the result shown in Table 3.6. We only changed the value of $\frac{c}{\alpha}$ from 0.01 to 1.0 and obtained Table 3.7.

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
TC	600	800	1000	1300	1797	2300	2998	3600	4400	4500	5398
s	5	7	9	12	17	22	29	35	43	44	53
y^d	100	100	100	100	97	100	98	100	100	100	98

$\frac{c}{\alpha} = 0.01$, n=100, Poisson distribution.

Table 3.6: SMP with $\frac{c}{\alpha} = 0.01$

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	90
TC	1100	1800	2500	3100	4400	5600	6700	7900	9100	10000	10000
s	11	18	25	31	44	56	67	79	91	100	100
y^d	0	0	0	0	0	0	0	0	0	0	0

$\frac{c}{\alpha} = 1.0$, n=100, Poisson distribution.

Table 3.7: SMP with $\frac{c}{\alpha} = 1.0$

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
TC	600	800	1000	1200	1700	2200	2700	3400	4000	4800	5500
s	5	7	9	11	16	21	26	33	39	47	54
y^d	100	100	100	100	100	100	100	100	100	100	100

$\frac{c}{\alpha} = 0.01$, n=100, binomial distribution.

Table 3.8: SMB with $\frac{c}{\alpha} = 0.01$

p	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
TC	1100	1800	2400	3000	4100	5200	6200	7100	8000	8900	9600
s	11	18	24	30	41	52	62	71	80	89	96
y^d	0	0	0	0	0	0	0	0	0	0	0

$\frac{c}{\alpha} = 1.0$, n=100, binomial distribution.

Table 3.9: SMB with $\frac{c}{\alpha} = 1.0$

In the model of SMB, we also chose $c_{l1}, c_{l2} = 0.009$ and $c_{l3}, c_{l4} = 0.3$ and $\frac{c}{\alpha} = 0.01$, we obtained the results in Table 3.8. We only changed the value of $\frac{c}{\alpha}$ from 0.01 to 1.0 and obtained the results in Table 3.9.

We achieved nearly the same solution with the Poisson (SMP) and binomial distribution (SMB)(see Tables 3.6-3.9). We found that, in this model, if the value of $\frac{c}{\alpha}$ is lower we can duplicate all the hubs and have no duplicated hub otherwise. Here we need to mention that Poisson distribution approach to binomial distribution only if the value of np is not too large. One can suggest that $p \leq 0.10$ and $n \geq 1000p$ (See [7], p205). The computation inaccuracy also affected the result of our model, so when p is too large the result is not very accurate. In real life we could not accept a hub with a higher than 0.5 probability of failure in per unit time, even the unit time is long.

By comparing SMP and SMB with SMN we find that they gave different results. It means, although we used the same SM, the different distribution of the number of failures of hubs may lead to a different maintenance policy. With our assumptions in this thesis, the binomial distribution is the real distribution. In our experiment we found that the SMN results in a more expensive maintenance policy.

3.9.2 A more practical model (MPM)

Failures have different consequences to different clients. For most clients the intermittent failures are usually not catastrophic. But failures of even a few seconds in some fly-by-wire avionics software may result in the aircraft's destruction. The tradeoff between high level service and cheap cost with lower service level has been positive. We define different clients. From the view of maintenance, the main difference between different clients is the maximum breakdown time. For the highest level clients, the minimum breakdown time should be very small, these clients are willing to pay a substantially higher cost for the best possible service level, then the higher ones, and then the common clients. For common clients breakdowns are not as sensitive. Taking all these mentioned points into account, we propose a more practical model (MPM) in this section.

Telecommunications networks exist in a variety of shapes and sizes. We consider

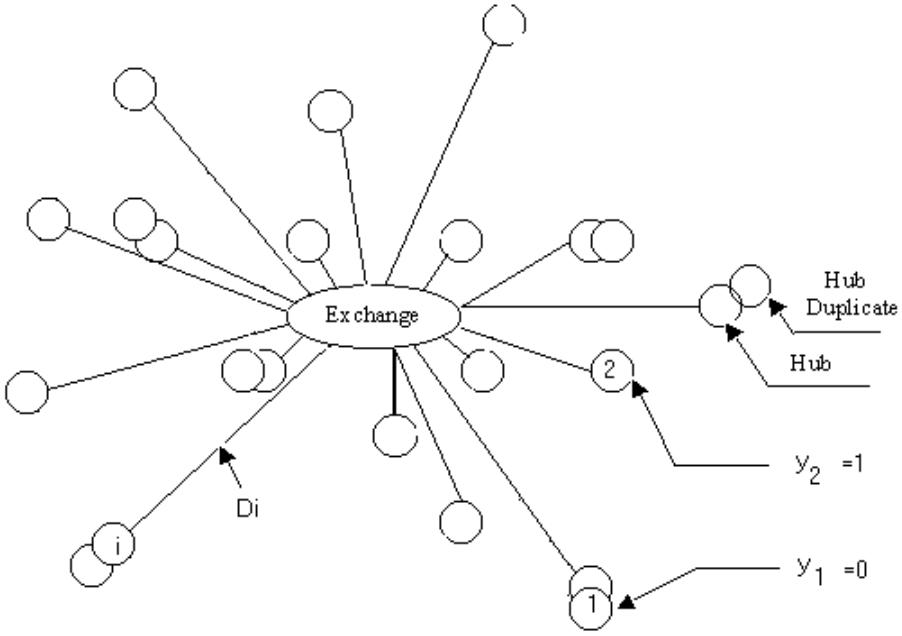

Figure 3.6: Network architecture for the practical model (MPM)

the generalized model, and more practically, we take the variation in travel distances into account. We also store some hubs to replace a failed hub when needed. Finally we introduce the difference of time needed to fix the duplicated hub, non-duplicated hub and replace the malfunctioning hub. Following the assumptions above, an example of the network architecture for MPM is illustrated in Figure 3.6.

Now we need to introduce some parameters and variables:

$t_{di} = D_i/V$;

D_i is the distance from hub i to the exchange, and V is the travel speed of the repairman.

t_0, the time needed to repair the duplicated hub. If the hub duplicate starts work automatically when the original hub fails, $t_0 = 0$.

t_1, the time needed to fix the non-duplicated hub .

t_2, the time needed to replace the non-duplicated hub.

Tm_i, the minimum download time for hub i, which is decided by the clients

served.

c_i, the price for duplicating one hub in unit time, the price for each hub is different.

c_{si}, the price for storing one hub in per unit time, the price is also different.

α — salary of one repair person per unit time;

p_i — probability of failure of original hub i per unit time;

p_{di} — probability of failure of hub duplicate i per unit time.

s — total number of repair persons employed; s is the nonnegative integer;

$$x_i = \begin{cases} 1, & \text{if the repairman wants to replace the failed hub with a stored one,} \\ 0 & \text{otherwise.} \end{cases}$$

We can get the total number of stored hubs by

$$x = \sum_{i=1}^{n} x_i,$$

y_i— indicates the duplication of the i-th hub:

$$y_i = \begin{cases} 1 & \text{if hub } i \text{ is not duplicated,} \\ 0 & \text{otherwise.} \end{cases}$$

The total cost function per unit time, TC can be defined as follows:

$$TC(y_i, x_i, s) = \alpha s + \sum_{i=1}^{n} c_i(1 - y_i) + c_{si}x_i \tag{3.9.18}$$

We can formulate it as a optimization problem:

$$\text{minimize } TC(y_i, x_i, s) \tag{3.9.19}$$

subject to

$$y_i, x_i, s \geq 0. \tag{3.9.20}$$

The first term in TC is the total salary of the crew of service persons and the second term gives the total hub duplication cost. The third term denotes the total hub store cost, which are hubs stored in the center.

We also assume that the number of failures of non-duplicated hubs is normally distributed with mean, $\mu(y)$ and standard deviation, $\sigma(y)$.

$$\mu(y) = \sum_{i=1}^{n} y_i p_i, \tag{3.9.21}$$

$$\sigma(y) = \sqrt{\sum_{i=1}^{n} y_i p_i (1 - p_i)}. \tag{3.9.22}$$

Now let us think about the situation when the duplicated hub fails and the duplicate hub also fails simultaneously. In this case, the probability is $p_i p_{di}$. p_{di} is the probability of the number of failures of duplicate hubs in per unit time. We also assume that the probability of the number of failures of duplicated hubs and duplicate hubs happening simultaneously is normally distributed with mean, $\mu_1(y)$ and standard deviation, $\sigma_1(y)$. So we get:

$$\mu_1(y) = \sum_{i=1}^{n} (1 - y_i) p_i p_{di}. \tag{3.9.23}$$

In this formulation the level of service provided to repair malfunctioning non-duplicated hubs is given as $\mu(y) + k_1\sigma(y)$. $\mu_1(y)$ repair persons are needed to repair malfunctioning duplicated hubs and duplicate hubs together.

Now we can formulate the MPM as below:

$$\text{Minimize} \quad [TC(y_i, x_i, s) = \alpha s + \sum_{i=1}^{n} c_i(1 - y_i) + c_{si} x_i] \tag{3.9.24}$$

subject to

$$\mu(y) + k_1\sigma(y) + \mu_1(y) \leq s, \tag{3.9.25}$$

$$t_{di} + t_0 + y_i t_1(1 - x_i) + y_i x_i t_2 \leq Tm_i, \tag{3.9.26}$$

where, i=1,2,..., n.

$$y, x \leq n, \tag{3.9.27}$$

$$s, y, x \geq 0, \tag{3.9.28}$$

$$s, y, x, \text{are integers.} \tag{3.9.29}$$

3.9.3 The "toy bricks" method

Now we simplify the problem by defining only three kinds of clients: gold, silver, and bronze. And then we assume there are only three kinds of distance: longest, middle, and small and the corresponding travel times are t_{dl}, t_{dm}, and t_{ds} respectively. Finally we assume that all hubs are the same. So $p_i = p$, $p_{di} = p_d$ and $c_i = c$, $c_{si} = c_s$. In this case the network architecture can be illustrated as Figure 3.7.

We add some new definitions:

m_j— is the number of hubs servicing j clients.

Tm_j, the minimum download time for the hub j, which is decided by the clients served.

$$x_{ji} = \begin{cases} 1, & \text{if the repairman wants to replace the failed hub with a stored one,} \\ 0 & \text{otherwise.} \end{cases}$$

We can get the total number of stored hubs by

$$x_j = \sum_{i=1}^{m_j} x_{ji},$$

y_{ji}— indicates the duplication of the i-th hub:

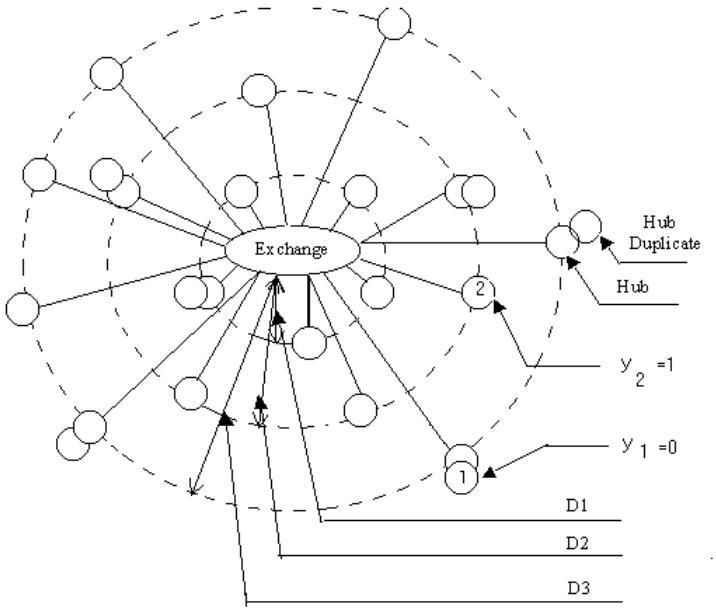

Figure 3.7: Network architecture for the simplified practical model (SPM)

$$y_{ji} = \begin{cases} 1 & \text{if hub } i \text{ is not duplicated,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_j = \sum_{i=1}^{m_j} y_{ji}.$$

After our simplification we have

$$\mu_j(y_j) = y_j p, \tag{3.9.30}$$

$$\sigma(y_j) = \sqrt{y_j p(1-p)}, \tag{3.9.31}$$

$$\mu_{j1}(y_j) = y_j p p_d. \tag{3.9.32}$$

So our problem can be described as

$$\text{Minimize} \quad TC(y_j, s_j, x_j) = \alpha s_j + c(m_j - y_j) + c_s x_j \tag{3.9.33}$$

Subject to

$$\mu_j(y_j) + k_1\sigma(y_j) + \mu_{j1}(y_j) \leq s_j, \tag{3.9.34}$$

$$t_{dl} + t_0 + y_{ji}t_1(1 - x_{ji}) + y_{ji}x_{ji}t_2 \leq Tm_j, \tag{3.9.35}$$

for $i \in Lo$

L_o is the set of hubs which are far away from the center.

$$t_{dm} + t_0 + y_{ji}t_1(1 - x_{ji}) + y_{ji}x_{ji}t_2 \leq Tm_j, \tag{3.9.36}$$

for $i \in Md$

M_d is the set of hubs which have middle distance from center.

$$t_{ds} + t_0 + y_{ji}t_1(1 - x_{ji}) + y_{ji}x_{ji}t_2 \leq Tm_j, \tag{3.9.37}$$

for $i \in St$.

S_t is the set of hubs which have middle distance from center.

Here for gold clients $j = 1$, silver $j = 2$ and bronze $j = 3$. It is obvious that $Tm_3 > Tm_2 > Tm_1$.

Now the algorithm is as follows:

Let $j = 1$ and solve the above problem we get a set of s_1, y_1, x_1. Then let $j = 2$, and then $j = 3$.

Our final solution will be $s = s_1 + s_2 + s_3$, $y = y_1 + y_2 + y_3$ and $x = x_1 + x_2 + x_3$.

We do not have numerical experiments for the MPM model using the "toy bricks" method. It is open for further study.

3.10 Conclusion

Although the SM model in paper [1] is simple, it gives the main idea of optimization in telecommunications network maintenance. The exchanges and hubs are some of the basic equipment in telecommunications. From the SM model we can get some

ideas about how to decide the number of repair persons and duplications of hubs, which are two of the basic factors in network maintenance.

For the SMN, the assumption of the number of failures of non-duplicated hubs per unit time is normally distributed with mean $\mu(y)$ and standard deviation $\sigma(y)$ making the model be more easily solved and visual.The idea of the 'repair of malfunctioning duplicated hubs will only be performed when there are $k_2\mu(y)$ repair persons available' is also novel. We also found some inadequacies in this model. When the value of np is too large, the normal distribution does not approach the real distribution (binomial distribution) and $\mu(y) + k_1\sigma(y)$ may be more than n. Thus, the limitation of applicability of normal distribution should be found. A direct method is introduced in my thesis to solve the SMN. The results prove this method is feasible and simple.

Binomial distribution is the real distribution under our assumptions in this thesis. Poisson distribution is more practical and common in queue systems. We introduce the SM model with binomial (SMB) and Poisson (SMP) distribution and compare them with the SM with normal (SMN) distribution. SMB and SMP need to be improved to make them more practical.

Here we would like to emphasize that there should be a very important prerequisite for the objective function: we must repair the malfunctioning hubs in time or duplicate them. Otherwise the penalty will be excessive. In other words, there is no such solution as $TC = 0$, where $s = 0$, and $y^d = 0$.

It is becoming apparent that there will likely be a demand for several service classes. One service class will provide predictable services for companies that do business on the telecommunications network. Such companies will be willing to pay a certain price to make their services reliable and to give their users a fast feel to their Web sites. This service class may contain a single service. Or, it may contain Gold Service, Silver Service and Bronze Service, with decreasing quality. Our MPM model is based on this idea. We believe that further research on this model might give more practical results.

Chapter 4

Queueing programming models in Telecommunications Network Maintenance—Corrective maintenance

4.1 Introduction

Network availability is an important criterion for mission-critical network-based services. This is particularly important as the high speed and high capacity technologies are deployed. The impact of a single failure in this case can be catastrophic and a large number of services might be affected. It is expected, therefore, that the maintenance cost will become a significant percentage of the total cost especially as the price of bandwidth is declining. Finding the optimal trade off between the cost of deployed capital and ongoing maintenance is therefore a very significant problem.

In this chapter we introduce queue programming models in telecommunications networks maintenance. The ideas of profit, loss, and penalty will help telecommunications companies to have a good view of their maintenance policies and help them improve their service. This new approach allows us to consider the time factor which is hard to take into account in the stochastic approach discussed in previous chapter.

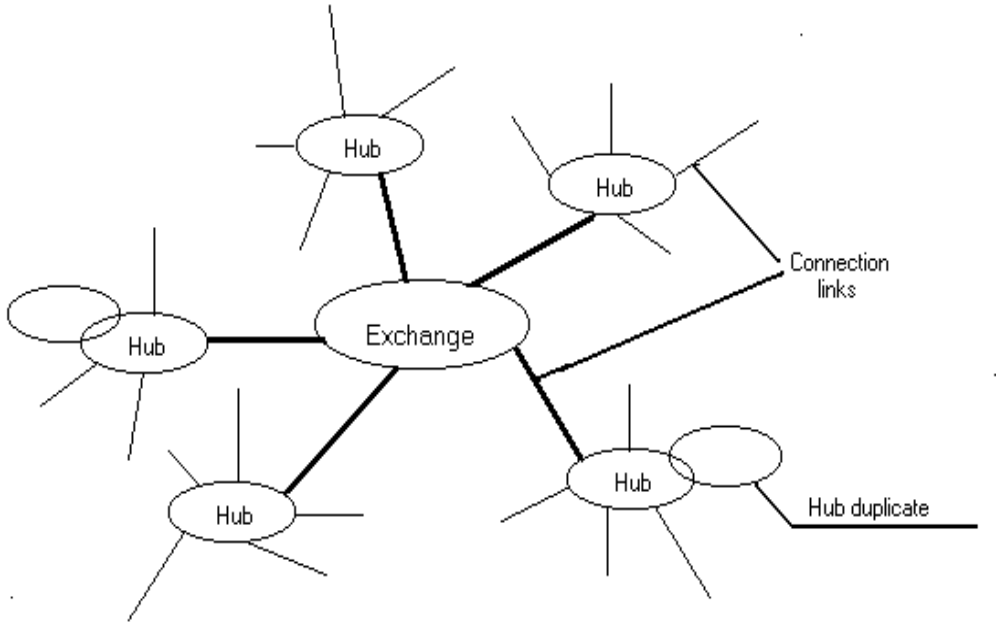

Figure 4.1: A star network configuration

How many hubs will stop working, and for how long will they be broken down (can we fix them)? That is the question that leads us to start our new approach with queuing theory in this chapter.

4.2 Configuration and definitions

Telecommunications networks exist to provide communication channels between points, which may be geographically close or, may be separated by worldwide distances. A telecommunications network generally is comprised of the following major components: exchanges, cables, power supplies, switches, routers, hubs, ports, and lots of software. In our work, for the sake of simplicity, initially we use the same star topology for network configuration as was used in the precious chapter shown in Figure 4.1.

The definitions of some terms such as **Exchange**, **Hub**, **Original Hub** and etc. are the same as those in section 3.5 in chapter 3.

4.3 Some assumptions for a simplified network architecture

For the sake of simplicity we rewrite the assumptions from chapter 3 as follows:

- The exchange is
  a. too expensive to be duplicated, and therefore will not be duplicated
  b. ideal, i.e. has zero probability of failure, $p_e = 0$.
- All original hubs
  (a.) are independent.
  (b.) are located equidistant from the exchange.
- A simple, but very general model is the stochastic binary system. Each hub can take on either of two states: operative or failed.
- The hub duplicates are ideal with $p_{di} = 0$, which guarantees the hub duplicate will work instead of the failed original hub. Each hub can only have one duplicate.
- The duplication costs are the same for every hub.
- The crew of repair persons is stationed at the exchange.
- There will be no absenteeism in the crew of repair persons and their skills are homogenous, i.e. the total service time taken by any repair person servicing a malfunctioning hub is equal.
- Only one repair person is needed to service one malfunctioning hub, regardless of the type of failure.

Following these assumptions, an example of the simplified network architecture is illustrated in Figure 4.2.

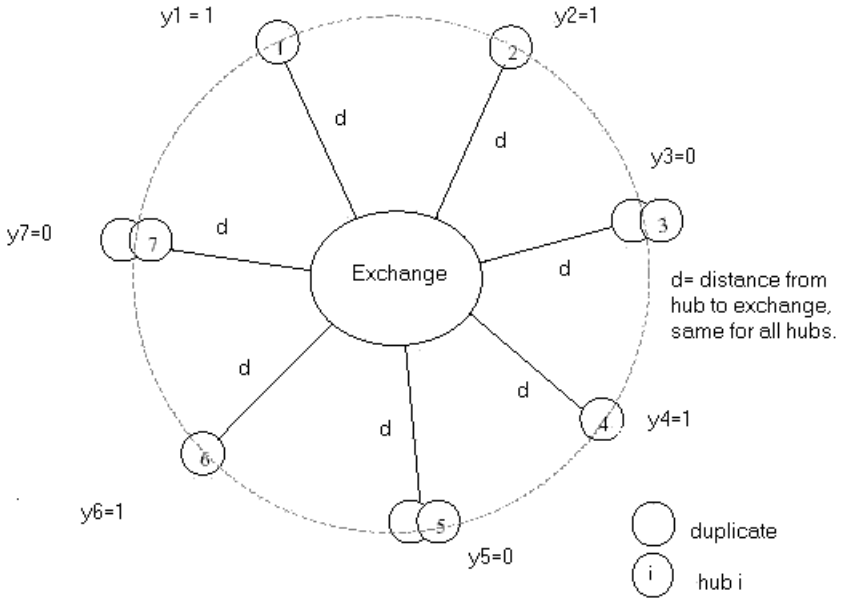

Figure 4.2: The simplified network architecture

4.4 The $M/M/s$ model

A queue can be defined by two elements:

1. the customers forming the queue: they are characterized by their numbers, and their arrival rates;
2. the servers, serving the customers from the queue, characterized by their numbers and the service times.

Queueing theory applies some probabilistic methods to find a deterministic model to queues: from arrival and service time patterns, it is possible to predict the average number of customers in the queue, as well as the average waiting time.

By analogy we can see the network maintenance as a queue: the service provided is the repairing of the malfunction hubs.

- The customers are the hubs, arriving in the line when they break down
- The servers are the repair persons

We assume that the s repair persons each are in charge of m identical hubs, each of which breaks down independently and randomly at an average rate λ in running time. It can be shown that the time between two hubs breaking down has a negative exponential distribution with a mean $\frac{1}{\lambda}$.

It is further assumed that repair times also have a negative exponential distribution with mean $\frac{1}{\mu}$.

The $M/M/s$ model with finite customer population has been discussed by Brian D. Bunday [10]. Its solution can be obtained from a special case of the birth-death equations.

Let λ_n be the rate of arrivals and $\frac{1}{\mu_n}$ the average service time when there are n hubs broken down. Then:

$$\lambda_n = \begin{cases} \lambda(m-n) & n \leq m; \\ 0 & n > m; \end{cases}$$
$$\mu_n = \begin{cases} n\mu & n = 1, 2, ..., s, \\ s\mu & n = s+1, ..., m. \end{cases} \tag{4.4.1}$$

Let $p_n(t)$ be the probability that n hubs are break down at time t. Then it is possible to obtain $p_n(t+\delta t)$ in function of the $p_k(t)$, as:

$$\begin{cases} p_0(t+\delta t) = p_0(t)(1-\lambda_0+\circ(\delta t)) + p_1(t)(1-\lambda_1+\circ(\delta t))(\mu_1\delta t+\circ(\delta t)); \\ \\ p_n(t+\delta t) = p_{n-1}(t)(\lambda_{n-1}\delta t+\circ(\delta t))(1-\mu_{n-1}\delta t+\circ(\delta t)) \\ \qquad\qquad +p_n(t)(1-\lambda_n\delta t+\circ(\delta t))(1-\mu_n\delta t+\circ(\delta t)) \\ \qquad\qquad +p_{n+1}(t)(1-\lambda_{n+1}\delta t+\circ(\delta t))(\mu_{n+1}\delta t+\circ(\delta t)); \\ \\ p_m(t+\delta t) = p_{m-1}(t)(\lambda_{m-1}\delta t+\circ(\delta t))(1-\mu_{m-1}\delta t+\circ(\delta t)) \\ \qquad\qquad +p_m(t)(1-\mu_m\delta t+\circ(\delta t)). \end{cases} \tag{4.4.2}$$

From equation (4.4.2), are deduced some differential equations, which can lead to some steady state equalities:

$$p_n = \begin{cases} \frac{m!}{(m-n)!n!}\rho^n p_0 & n = 1, 2, ..., s, \\ \frac{m!}{(m-n)!s!s^{n-s}}\rho^n p_0 & n = s+1, ..., m, \end{cases}$$

under the condition

$$\rho = \frac{\lambda}{\mu} < 1,$$

that is $\lambda < \mu$, or a repairman can repair a hub faster than its average life duration. From the equality $\sum_{n=0}^{m} p_n = 1$ we then deduce that

$$p_0 = \frac{1}{\sum_{n=0}^{s} \frac{m!}{(m-n)!n!}\rho^n + \sum_{n=s+1}^{m} \frac{m!}{(m-n)!s!s^{n-s}}\rho^n}.$$

We can then easily deduce the average number of down hubs at a given time:

$$L = \sum_{n=0}^{m} n p_n;$$

the average time spent in the queue is then

$$t_w = \frac{L}{\lambda_L},$$

where λ_L is the arrival rate of the hubs in the queue when there are L hubs down. By analogy with equation (4.4.1), we have

$$\lambda_L = \lambda(m - L). \tag{4.4.3}$$

We might also want to know the distribution of the time spent waiting for repair. The event $(t \leq T \leq t + \delta t)$ is the union of the mutually exclusive events

$$E_n = \begin{cases} [p_n(t_0)] & \text{There are already } n \text{ hubs down at the breaking time.} \\ [P_{t,n-s}] & \text{There are } (n-s) \text{ service completions during time } t. \\ [\mu_s \delta t] & \text{There is one service completion in the interval } (t, t+\delta t). \end{cases}$$

If there are less than s hubs down, there are some free repair persons, and there is no waiting time.

Otherwise, we have

$$P_{t,n-s} = \frac{(\mu_s t)^{n-s} e^{-\mu_s t}}{(n-s)!}.$$

Thus the distribution for the time spent waiting for repair by one hub is

$$\begin{cases} P(t=0) = \sum_{n=0}^{s-1} p_n, \\ \Phi(t)\delta t = \sum_{n=s}^{m} p_n \cdot \frac{(\mu_s t)^{n-s} e^{-\mu_s t}}{(n-s)!} \cdot \mu_s \delta t. \end{cases} \tag{4.4.4}$$

Thus the probability of exceeding a given time t_m is

$$P(t \geq t_m) = \begin{cases} 1, & \text{if } t_m{=}0, \\ \int_{t_m}^{\infty} \Phi(t)\delta t = \sum_{n=s}^{m} \left(p_n \cdot e^{-\mu_s t_m} \sum_{i=0}^{n-s} \frac{(\mu_s t_m)^i}{i!} \right), & \text{otherwise.} \end{cases} \tag{4.4.5}$$

Remark 1 $\int_0^{\infty} \Phi(t)\delta t \neq 1$ *because* $P(t=0) \neq 0$. *It is easy to verify that* $\int_0^{\infty} \Phi(t)\delta t = 1 - \sum_{i=0}^{s-1} P_n$.

4.5 The general model (GM)

We start to consider the general queueing model:

$$(M/M/s) : (F/2m/2m);$$

the meanings are given as below

- M means Poisson failure rate distribution with

  λ_1 — the rate of failure of original hub per unit time,

  λ_2 — the rate of failure of duplicated hubs per unit time,

  In our current models for the simplicity we take $\lambda_1 = \lambda_2 = \lambda$.

- Second M Exponentiation distribution fix time with

μ_1 — the rate of service time to fix the failed hub for hubs without hub duplicate,

μ_2 — the rate of service time to fix the failed hub for hubs with hub duplicate.

In our current models for the simplicity we take $\mu_1 = \mu_2 = \mu$.

ρ — the utilization of service, $\rho = \frac{\lambda}{\mu}$.

- Repair persons

  There are s homogeneous repair persons.

- The service discipline of F.

- The maximum number of hubs in the queue:

  there are at most $2m$ failed hubs that wait for repairing.

- The size of source:

  the possible total number of original hubs and hub duplicates in this model is $2m$.

Variables and parameters:

m — The total number of original hubs in this model,

s —total number of repair persons employed,

$$y_i = \begin{cases} 1 & \text{if hub } i \text{ is not duplicated,} \\ 0 & \text{otherwise.} \end{cases}$$

The total number of non-duplicated hubs will be:

$$y = \sum_{i=1}^{n} y_i$$

y^d— the number of original hubs with duplicated hubs (y+y^d=m).

L — the number of failed hubs waiting and being repaired.

L_q — the number of failed hubs in queue waiting for repair.

t_n— breakdown time for non-duplicated hubs.

t_d— breakdown time for duplicated hubs.

t_m— the permitted minimum breakdown time.

t_0— penalty time for non-duplicated hubs.

$$t_0 = t_n - t_m.$$

i_p — indicator of penalty:

$$i_p = \begin{cases} 1 & \text{if the breakdown time exceeds the permitted minimum breakdown time.} \\ 0 & \text{otherwise.} \end{cases}$$

p_n — the steady-state probability of n failed hubs in system parameters.

c_d — cost of duplicate per hub duplicate per unit time.

c_n — the profit loss of one no working hub per unit time.

c_p — cost of penalty when the waiting time exceeds the minimum breakdown time.

c_s — cost of repair person's salary in per unit time.

4.6 GM with a maximum objective function (GMA)

Now we consider the way to maximize the income. Under the assumptions defined before, we add some new definitions and redefine some parameters for this model.

- Among y hubs composing a network, there are two different types of hubs - and thus of customers
  - $y - y^d$ non-duplicated ones, with failure rate λ, and repair time rate μ,
  - y^d duplicated ones, with failure rate λ^d, and repair time rate μ^d.
- Each kind of customer can be assigned a different priority. Once these parameters are known, it is possible to find

  - L the average number of positions where no service is provided due to failed hubs.
  - t_w the average waiting times repairs of non-duplicated hubs.

- The cost also depends of some independent parameters:
  - t_m, the permitted breakdown time,
  - c_d the cost per unit time for the duplication of one hub,
  - c_w the profit of one working hub location per unit time,
  - c_p fixed penalty cost when the waiting time exceeds the permitted breakdown time,
  - c_s one repair person's salary per unit time.

There are s repair-persons working for the company. The cost of employing these persons per unit time is

$$sc_s.$$

As well, the total cost per unit time for the duplicated hubs is

$$c_d y^d.$$

We know there are on average L locations waiting for service in the queue, which means that there are $m - L$ locations working. The profit per unit time made by the company with these hubs is

$$(m - L)c_w.$$

Let i_p be the indicator of the penalty being given (i.e. the probability of the time $t_m - \frac{1}{\mu_L}$ to be exceeded, defined in (4.4.5)):

$$i_p = P(t \geq t_m - \frac{1}{\mu_L}),$$

where $\frac{1}{\mu_L}$ is the average repair time. By analogy with equation 4.4.1, we have

$$\mu_L = \begin{cases} L\mu, & \text{if } L < \mu; \\ s\mu, & \text{if } L \geq \mu. \end{cases}$$

In other terms, for each hub breaking down, i_p is the probability, that the company will have to pay the penalty to the users of this hub. The average penalty cost per hub for the company is $i_p c_p$.

The average number of hubs breaking per unit time is: λ_L, defined in equation 4.4.3 and thus the penalty cost per hour for the company is

$$\lambda_L i_p c_p.$$

Our objective is to maximize total income (TI) of the company, by constructing a network composed of a judicious number of duplicated hubs, and hiring a reasonable crew. The optimization problem to solve is

$$\text{maximize} \qquad TI(s, y^d) \equiv c_n(m - L) - c_s s - c_d y^d - \lambda_L i_p c_p \tag{4.6.1}$$

$$\text{subject to} \quad \begin{cases} 0 \leq y^d \leq m, \\ 0 \leq s \leq 2m, \\ s, y^d \in \mathbb{N}. \end{cases} \tag{4.6.2}$$

A simplified model of GMA (SGMA)

In this model, we consider that the duplicated hubs are too expensive, so there is only one hub in one location. Our problem becomes:

$$\text{maximize} \qquad TI(s) \equiv c_n(m - L) - c_s s - \lambda_L i_p c_p \tag{4.6.3}$$

$$\text{subject to} \quad \begin{cases} 0 \leq s \leq m, \\ s \in \mathbb{N}, \end{cases} \tag{4.6.4}$$

where all the values are defined as in Section 4.5.

4.7 Solution method

The models with one queue discussed before was solved [105]. From a programming point of view, the objective function presents two major problems:

4.7.1 The problem of large numbers and integer-type functions

The size of the problem is extremely limited by the accuracy of the computer. As the objective function contains factorials, a big size m can induce a very small value for p_0. In this case, the computer will approximate these values by 0, and the calculations of p_n will be wrong. Some programming techniques can be applied to overcome this difficulty. The computation time will then be increased considerably. These techniques have not been used here, the tests have been carried out over small size functions (m=100).

Because the parameters of the objective function are integers, the optimization technique has to be carefully chosen. Two different methods have been implemented. For small sized integer problems, the basic method is to sort out all the solutions, and select the best one. Tests having been carried out over small-sized problems. This method has the fastest calculation time for the maximal value for the objective function.

4.8 Numerical results

For reasons of confidentiality, real telecommunications network architecture is rarely published by any company. It is impossible to find any data from real networks and compare them with our theoretical results. In this thesis, we have therefore evaluated the reaction of our model to some tendency and evaluated its performance.

On Figures 4.3 and 4.4, we can see that

- The total income (TI) decreases in a function of ρ (the probability of failure).

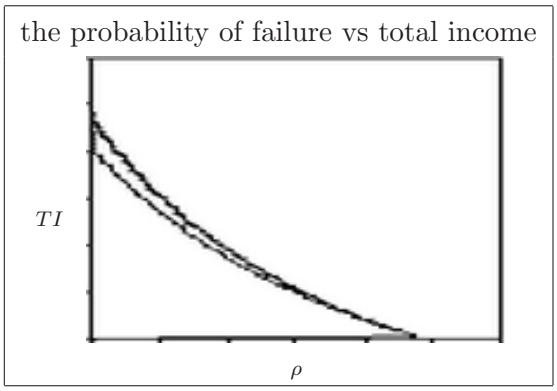

Figure 4.3: The $TI(\rho)$.

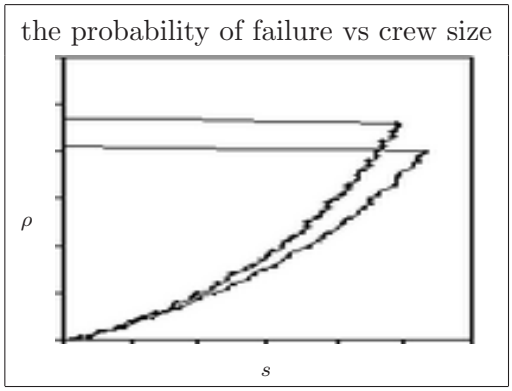

Figure 4.4: The probability of failure vs crew size

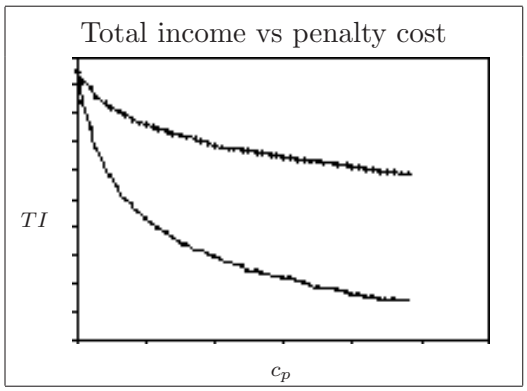

Figure 4.5: The $TI(c_p)$

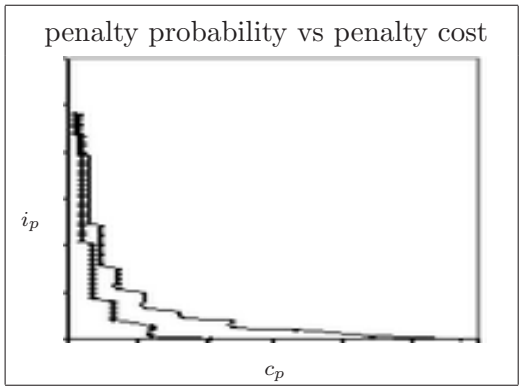

Figure 4.6: The penalty probability vs penalty cost

- The number of employees (s) needed increases when the arrival rate of failed hubs increases.

All these results are logical and the same with the results we got in previous chapter. When the optimal income of the company becomes negative, our model recommends to stop using this company.

From Figures 4.5-4.7, we can reach the conclusion:

- The optimal income decreases when the penalty cost increases.
- When the penalty cost increases, the optimal solution for the probability to pay it decreases.
- We can also observe the size of the queue which decreases as the penalty cost increases.

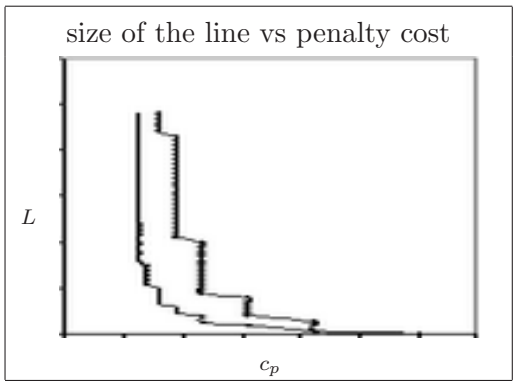

Figure 4.7: The size of the line vs penalty cost

Also we should notice here some results about the relations between the penalty and total income, the size of queue and the penalty cost. We need to have a tradeoff to meet different goals. If we want to offer more reliable service to users then we choose a big penalty in the objective function. Of course we can charge users more. For cheap reasonable reliable service we can choose small value of penalty and longer permitted breakdown time.

4.9 Conclusion

In this chapter we proposed telecommunications maintenance models based on queueing theory. Queueing theory is based on the concept of probability and being used to reduce stochastic model to a deterministic model. Although the queueing models we used are relatively simple and certain practical aspects are ignored, our objective is to gain insight that can guide the design of our maintenance policy. The results obtained in this thesis are very reasonable, although no comparison is possible with real networks.

All the models proposed in the previous chapter and this chapter are aimed to solve the same problem: telecommunications network maintenance. However they approach this problem in different ways. Stochastic programming approach is relatively simple and emphasizing the static state of this problem, so it is good for a decision maker to consider it as a long term monitoring approach. While the queueing approach takes the sort of dynamic states into account and it also

reduces this problem to a certain optimization problem. All these can help the telecommunications companies to put this model into the use in their businesses.

This area is not completed if only a study the corrective maintenance without preventive maintenance is done. So in section 7.3 we will introduce preventive maintenance. We believe that it has the same important as corrective maintenance. Unfortunately we do not have time to do more study in this area.

Queueing theory can also be useful for models in some other fields, and our model can be adapted to these fields with considerable success. In this case, the optimization process described here is quite efficient, and can also be adapted.

Chapter 5

Optimization based clustering algorithms (OBC) for network evolution

5.1 Introduction

As the telephone industry moves into the 21st century it has become necessary for its architects to make some fundamental decisions about the way traffic is transmitted, switched, and managed. The issue that faces all of us, equipment vendors, service providers, and end-users alike, is that legacy networks no longer can handle the multiplicity and diversity of traffic that it increasingly has to support.

The success of the Internet is due to best-efforts delivery which allows for easy expansion, coupled with a congestion-adaptive reliable transport protocol (TCP), which serves well for the delay insensitive traffic of Web browsing or file transfer. There are several new proposals for handling Quality of Service (QoS) for real-time and multimedia traffic, but their introduction is slow. The scale of the Internet is one of the impediments to successful QoS provision. Making fundamental decisions that will change the shape and direction of telecommunications and Internet architecture is not easy. It requires resolve and clear thinking on the part of everyone involved. Hierarchical structure is the key to scaling problems, but it must be provided in a

way that helps evolution.

5.1.1 Model for network design

There are many types of devices used to construct a network, often referred to generically as: **nodes**. Sometimes nodes are distinguished by their functions. Such as

- terminals: simple devices, usually serving a single user;
- Hosts: a large computer serving many users and providing computing capability or access to a database.
- Multiplexors/concentrators: devices which join the traffic on low speed lines into a single stream which can use a higher speed line.
- Local switches: devices which allow attached facilities and devices to communicate directly with one another.
- Routers: devices which are used to connect multiple local area networks.

In this thesis the node refers to different things. Sometimes it refers to city and sometimes refers to routers. This will not lead to any misunderstanding.

The network design problem is usually thought of as one of minimizing cost while satisfying throughput requirements. However, constraints on performance must also be satisfied. The following is a very general formulation:

Given: Node locations, channel capacity options and costs
Minimize: Total communication cost
over: Topology, channel capacities, routing
Subject to: Delay constraint,
reliability constraint,
traffic requirement.

In general, there are $2^{N(N-1)/2}$ possible topologies, where N is the number of nodes. A network with ten nodes, for example, has 45 potential links, each of which

could be either included or excluded from the network. Even if it is assumed there is only one possible link speed, this gives rise to 2^{45} possible solutions, more than 10^{13} possibilities. Furthermore, capacities are available in discrete sizes. In addition, the constraints must be satisfied. This means that an enormous integer optimization problem must be solved.

Existing heuristic design procedures are quite efficient for the design of small to moderate sized networks (25-75 nodes); however, they become very costly and even prohibitive when dealing with large networks [196].

Suppose that a given set of city's locations is to be connected and that the cost of all possible connections is known. We do not care about capacity, reliability, delay, or any other performance objectives. We simply want to connect the cities at minimum cost. The focus in this chapter is on specific problems in topological optimization within the evolution of network [29].

5.1.2 Hierarchical network design

All large scale networks are hierarchical. Network design problems can (roughly) be divided into two categories (I) Access network design and (II) Backbone network design [125]. Access network design assumes a centralized traffic demand and may involve hierarchical structures. These problems are closely related to facility location problems and clustering problems. Backbone network design, on the other hand, assumes a distributed (several sources and destinations) traffic demand and allows arbitrary topologies of the network to be chosen.

In backbone telecommunications networks, switches are arranged in groups and the groups of switches are connected. Very little research has been done on hierarchical backbone networks with a distributed demand pattern. Several reasons are immediately obvious, one being that dividing the network into groups limits the choice of solutions, and hence low cost solutions may be overlooked. Additionally, routing the distributed demands while designing the topology of the network presents substantial difficulties.

5.1.3 Hierarchical routing

Hierarchical routing protocols were introduced in the early 80's [100, 101] with the growth of data communication networks. It was envisaged that the storage and updating costs necessary for dynamic routing algorithms would become prohibitive as the network size increased.The proposed solution was to divide the network into several routing areas. With this scheme, the nodes have complete routing information about their routing area: They have knowledge of all the addresses which are reachable within their area and what is the best path to reach them. Nevertheless, the nodes of this network have only partial information on the outside world (i.e. the other partitions). Information exchanged between two adjacent routing domains is summarized by special nodes in order to achieve this goal. This summary process is known as aggregation and is usually based on address masks. This way, the routing table contains one "entry" (which roughly consists of a destination address and path information necessary to reach it) for subnetworks in the same area, and one entry per set of subnetworks (possibly one per remote area) for destinations beyond the area limits.

Hierarchical routing schemes based on the hierarchical clustering of the network nodes are therefore proposed to reduce the cost of routing. Those studies show a remarkable efficiency of optimally selected hierarchical routing schemes for large networks.

In general, hierarchies are needed to reduce the burden on routers of keeping large amounts of network state information and to reduce the number of signalling messages exchanged. Provided suitable hierarchies are available, full information need only be kept on systems within the same hierarchical level together with summarized information about adjoining levels. The technique is therefore particularly helpful for routing where large tables must currently be maintained. Increasingly, routers and servers on the Internet are being equipped to provide specialized services, especially those associated with QoS provision or multicast. To evolve such services to a large user population, it is important that systems with similar capabilities can be categorized and merged to form larger systems which can then form their own hierarchies.

5.1.4 QoS on the Internet

Todays Internet offers a best effort connectionless service, in which datagrams may be lost, discarded, unpredictably delayed or mis-ordered. Fortunately, the Transmission Control Protocol (TCP) can both recover from errors and reduce injected traffic when there is congestion, so that non-delay sensitive (elastic) traffic requiring reliable delivery (e.g.file transfer, Web page retrieval) is well served. Real time traffic, such as voice or video that have delay and bandwidth constraints (inelastic traffic) are poorly served, but there is increasing demand for them to be carried on the Internet. Such traffic can only be carried if the network allows the negotiation of Quality of service parameters and offers resource reservation, queueing disciplines and routing together with the protocols to support them. A number of proposals have been put forward including the Integrated Services Architecture (Intserv), Differentiated Services (Diffserv) and Multi-Protocol Label Switching (MPLS).

Intserv [120] follows similar conventions to those of ATM networks and supports individual flows with specific QoS requirements. It is intended to support guaranteed, controlled (i.e. within some tolerance limits) and best efforts traffic. To maintain QoS and control congestion, Intserv requires admission control, routing algorithms, queueing disciplines and discard policies. A router must deal with the flows as a whole and maintain look-up tables that allow fast decisions on classification and priority queueing for each packet. Intserv is complex and has therefore been slow to be implemented. One important component, which is now implemented in many routers, is the Resource ReServation Protocol (RSVP) [121] which can handle both unicast and multicast resource provisioning.

Because of the difficulty of introducing Intserv, an alternative technique based on service classes, Diffserv, was introduced [122]. This is intended to aid scalability, to offer an evolutionary approach and to reduce complexity within the network. In Diffserv, the QoS is based on a field in the IP header. It offers differential levels of service for aggregated traffic (normally unidirectional) and uses well-defined building blocks for end-to-end agreement and hop by hop treatment, thus separating policy from forwarding. Diffserv works well for bandwidth-intensive data applications. But it does not give service guarantees per se.

More recently, MPLS has been proposed [116]. In this, packets are forwarded in Forward Equivalence Classes (FEC) indicated by a label. Each packet is assigned to an FEC and labelled as it enters the network (based on analysis of the header). The label is used to determine QoS and forwarding policy. Because of its similarity with lightweight virtual circuits, MPLS offers easy integration with ATM and Frame Relay networks. If the QoS needed is just within a subnetwork or a WAN cloud, these Layer 2 technologies (such as ATM, Frame Relay, and Token Ring), especially ATM, can provide the answer. But ATM or any other Layer 2 technology will never be pervasive enough to be the solution on a much wider scale, such as on the Internet [197].

Because of the variety of schemes and the difficulty of introduction of the services, it is likely that QoS will initially only be offered in small clouds. Most current implementations are on IP-based intranets belonging to specific organizations or consortia. As the techniques begin to be more widely accepted, it is likely that they will be made available to users on the public Internet and the problem of scaling arises. To overcome this, we need to be able to identify routers and servers which offer specific public QoS support and to group them in the most appropriate way. In order to do this, we need information on geographical location, services provided, and whether subscription and charging are supported. Some of this information is already available, but better availability would help Internet evolution regardless of the scalability mechanisms used. Intuitively, we need to group systems together geographically - hence clustering seems a sensible approach [29].

5.1.5 Clustering algorithms

Clustering algorithms are a method of partitioning a set of points based on the characteristics of the points; the aim is to produce clusters of points that are more similar to other points in the same cluster than to points in other clusters.

There are two main types of clustering algorithm: hierarchical and non-hierarchical. See for example [92]. A hierarchical clustering algorithm assigns each node in turn to an appropriate cluster, then repeatedly merges two nearby clusters until all points belong to a single cluster. (Note that the term hierarchical is used in a different

sense when we discuss multicast hierarchies.) Non-hierarchical clustering algorithms typically divide the set of points into a given number, k, of clusters based on a given starting set of cluster-centers and then iterate to optimize the membership of the clusters.

An important example of non-hierarchical clustering is the k-means algorithm, that partitions a data set into k clusters as follows:

- Specify k nodes as initial centers
- Allocate each node to the cluster containing the nearest center to the node
- Re-calculate the center of each cluster.
- Repeat steps 3 and 4 until solutions converge (or for a maximum number of rounds).

For each type of clustering algorithm, we can choose from a number of options, for example, to find the nearest neighbor or to optimize cluster cohesion. One recommended technique is to determine a suitable number of clusters by evaluating the results of a hierarchical technique and then to optimize for this number of clusters by using a non-hierarchical technique. In paper [93], Waters wrote: "We have chosen clustering algorithms firstly, for the flexibility they offer in dividing the receivers in different ways e.g. by the maximum number in a cluster, by the total number of clusters, by limiting the number of sub-clustering steps to control the depth of the tree. Secondly, clustering has worked successfully in previous networking design scenarios. We believe that clustering algorithms can be particularly useful in high level design e.g. for overnight reconfiguration for file or content distribution. Such clustering can also reflect large distributed communities and populations, as occur in Grid Computing."

The easiest way to perform the clustering is to use a Euclidean distance to determine the closeness of the systems considered. In the first set of evaluations, we used geographical co-ordinates as our Euclidean plane. This in fact assumes a plane where each degree of longitude and latitude is treated as identical and does not take account of the spherical nature of the earths surface. If we consider the distribution of Internet users, they are concentrated on major cities and are certainly not

evenly spread around all possible physical locations around the globe. It is therefore appropriate to consider the use of clustering to determine the general location of users. Although the geographical location of all Internet users may not be available, it is probable that the geographical location of the members of a global Intranet community would be known.

If we consider maps of Internet use, this is concentrated on major cities and is certainly not evenly spread around all possible physical locations. By arranging the system into clusters, the number of trunk interconnections can be reduced. Iterative use of clustering algorithms should also help to divide very large numbers of systems into further levels.

Clustering techniques have been applied in many fields, including image analysis and vowel discrimination in speech. Their application to the design of backbone communication networks is described in [91]. One of the first problems of applying this technique to packet networks is to decide what are the optimization criteria? For instance, multicast applications may need a maximum fan-out at each level or may need to minimize the number of levels in the hierarchy. Note that clustering algorithms will always partition the nodes into sets and in many cases a useful grouping will be obtained even if it is not optimal.

5.1.6 Related Work

Hierarchical network design

Papers that discuss a centralized demand structure and hierarchies include [131] which solves an access network design problem and [133] which solves a network design hierarchical star-star problem. The multi-level capacitated minimum spanning tree problem [128] has a centralized demand structure, and a hierarchical structure emanates, because the edges have different levels representing the capacities.

Another type of hierarchical network design problem is defined in [130]. The problem is somewhat different from others: Given two primary nodes and some secondary nodes, the least expensive network connecting the primary nodes with a primary path and connecting secondary nodes not on the primary path to a node on

the primary path via a secondary path is to be found. The problem is extended with transshipment facilities in [129]. In [135] a new formulation is presented which is used to obtain a Lagrangean relaxation based algorithm. The algorithm is modified to handle the transshipment facilities in [127]. In [136] a dynamic programming based algorithm is suggested and in [137] the problem is enhanced to allow multiple paths and the dynamic programming algorithm is modified to handle this. Paper [132] presents an arborescence formulation for the problem, with multiple primary nodes. Paper [126] solves a multi-level network problem with more destination nodes but only one source node of the highest level. The destination nodes require service of different levels, and potential source nodes of lower levels require service of higher levels to be able to supply the service. Yet another way to use multi-level (or -layer) network design is to let each layer represent a network protocol layer. This is the approach of [134]. The layers can have different properties; some layers may, for example, be able to reconfigure the routing of traffic to avoid cables which have failed. Thereby, robustness of the network can be addressed.

Hierarchical clustering

A number of authors have considered techniques for networking hierarchies and, although several of these discuss clusters, they do not actually use clustering algorithms to group nodes into clusters as we propose. Most proposals are related to multicasting or to Web caching [29].

An approach to hierarchical clustering for multicast routing based on Voronoi diagrams is discussed by Bacelli et al [90]. (A Voronoi diagram divides a plane into regions each based on a specified node. Within each region, any other node is closer to that regions specified node than to the specified node in any other region.) The technique divides the network into domains each fed by a core and cores are organized into a hierarchy of center-based trees.

Although this approach looks promising, their evaluation uses a uniform distribution of nodes, which may not be realistic. The authors found that optimum tree cost can be found by having much bigger fan-outs nearer the source of the multicast tree, with successively lower fan-outs at lower levels in the hierarchy, whereas practical systems may require similar fan-outs at each level.

Another hierarchical scheme is that of Chatterjee and Bassiouni who describe a two level hierarchy [25]. Optimization at the top level (linking the clusters) uses a minimum spanning tree and at the lower levels, a broadcast tree with optimal delay within each cluster. Yallcast is another protocol-based approach to hierarchical multicast trees in the Internet [119].

A project to evaluate hierarchical structures for reliable multicast distribution is being undertaken at Carnegie Mellon University [124]. One such scheme is the TRAM protocol [118] with a hierarchical tree structure of "repair heads" for repairing failures in transmission to a multicast group. Tree organization uses advertisement protocols with expanding ring searches and dynamic distributed procedures.

Web caching often uses hierarchical structures; these methods combine a method of detecting well-used pages with decisions about the hierarchy. For example, in the LSAM proxy cache [95], their Intelligent Request Routing uses a neighbour-search operation to configure the proxy hierarchy. Another example is the Squid Proxy Cache [94].

We believe that it will be useful to combine distributed algorithms with the more centralised approaches described here. Distribution and searching are best used for finding service-capable portions of the network and for dynamically updating existing hierarchies.

The description of the use of clustering algorithms for backbone design in telephony networks in Cahn ([91]) prompted our examination of their application to network evolution. Inspired by Waters' work [29, 93] we started to using optimization based clustering algorithms to solve the similar problems in [71, 72]. In paper [71] we compared several optimization based clustering methods and their combinations with the k-means method. In paper [72], we formulated this problem as an optimization problem with a non-smooth, non-convex objective function. We introduce the penalties to find the real nodes as centers and total center.

5.2 Setting of the problem

We now describe our problem. Assume that a set A of m nodes on the plane is given:

$$A = \{a^1, \ldots, a^m\}, \text{ where } a^i = (a_1^i, a_2^i)$$

The aim of the hard clustering is to decompose A into a given number k of disjoint subsets (clusters) A^i, $i = 1, \ldots, k$. The choice of k is according to the scale of network and hop-constraints (hierarchical levels). We can consider different value of k then choose the one which gives the best result. In each cluster we choose one node as a center (x_i) and all other nodes will be connected to the closest center. Then we will choose one node (x_*) from all nodes as a total center; all centers will connect to this total center. Assume that clusters $A^1, \ldots, A^k$, their centers $x_1, \ldots, x_k$ and the total center x_* are given. Then the total cost C of this tree can be calculated as

$$\begin{aligned} & C(A^1, \ldots, A^k, x_1, \ldots, x_k, x_*) \\ & = \sum_{i=1}^{k} \sum_{a \in A^i, a \neq x_*} \|a - x_i\| \\ & \quad + \sum_{i=1}^{k} \|x_i - x_*\|. \end{aligned}$$

We include the condition $a \neq x_*$, since when we do clustering the total center node belongs to one of the clusters. Our goal is to solve the following problem (P): find clusters $\bar{A}^i$, their centers $\bar{x}_i$ and the total center $\bar{x}_*$ such that

$$\begin{aligned} P: \qquad & C(\bar{A}^1, \ldots \bar{A}^k, \bar{x}_1, \ldots, \bar{x}_k, \bar{x}_*) \\ & \leq C(A^1, \ldots, A^k, x_1, \ldots, x_k, x_*) \end{aligned}$$

for all collections of clusters A^i, their centers x_i and the total center x_*.

Thus the problem of finding clusters A^i can be formulated as the following optimization problem (P_1):

$$P_1: \qquad \text{minimize } C(A^1, \ldots, A^k, x_1, \ldots, x_k, x_*)$$

subject to

$$\{A^i\}_i \in \bar{C}$$

Here $\bar{C}$ is a collection of all possible partitions of the set A.

This formulation is not suitable for direct application of optimization techniques. Therefore we replace problem (P) with the problem (P_1), which can be solved by methods of non-smooth optimization. First, we assume that the center of a cluster A_i $(i = 1, \ldots, k)$ is not necessarily a real node, it can be an arbitrary point y_i on the plane. In such a case we call it an artificial center. If the set $(y_1, \ldots, y_k)$ of artificial centers of clusters is known then the clusters themselves can be easily described; namely the cluster A_i consists of points $a \in A$ such that $\|y_i - a\| < \min_{i' \neq i} \|y_{i'} - a\|$. Here $\|x\|$ is the Euclidean norm of a point x. Thus

$$a \in A_i \iff \|y_i - a\| = \min_{i'=1,\ldots,k} \|y_{i'} - a\|. \tag{1}$$

It follows from (1) that

$$\sum_{i=1,\ldots,k} \sum_{a \in A^i} \|y_i - a\| = \sum_{a \in A} \min_{i=1,\ldots,k} \|y_i - a\|. \tag{2}$$

We also assume that the total center y_* can also be an artificial point. Moreover y_* is the centroid of the set $(y_1, \ldots, y_k)$:

$$y_* = (1/k)(y_1 + \ldots, y_k) \tag{3}$$

Assume that centers of clusters $y_1, \ldots, y_k$ are known. It follows from (1), (2) and the definition of y_* that the total cost $\tilde{C}$ of this tree depends only on centers:

$$\tilde{C}(y_1, \ldots, y_k) = \sum_{a \in A} \min_{i=1,\ldots,k} \|y_i - a\|$$

$$+ \sum_{i=1}^{k} \|y_i - y_*\|, \tag{4}$$

where y_* is defined by (3). Thus if we restrict ourselves to the search for clusters with artificial centers, we need to minimize the function $\tilde{C}$ defined by (4) without any constraints.

Recall that the search for clusters can be characterized as the simultaneous minimization of the variation within clusters and the maximization of the variation between clusters. A formalization of this idea can be given as follows:

We say that a set $(y_1, \ldots, y_k)$ of artificial points is the set of centers of k-clusters of the set A if

$$\sum_{a \in A} \min_{i=1,\ldots,k} \|y_i - a\| = \min_{y'_1,\ldots,y'_k} \sum_{a \in A} \min_{i=1,\ldots,k} \|y'_i - a\|, \tag{5}$$

where the minimum is taken over all collections of points $y'_1, \ldots, y'_k$, where y'_i belongs to the plane. A detailed discussion of this definition can be found in [27].

Let

$$f_1(y_1, \ldots, y_k) = \sum_{a \in A} \min_{i=1,\ldots,k} \|y_i - a\|, \tag{6}$$

$$f_2(y_1, \ldots, y_k) = \sum_{i=1}^{k} \|y_i - y_*\|. \tag{7}$$

Then $\tilde{C} = f_1 + f_2$. Thus the minimization of cost does not coincide with the search for clusters. Indeed, the search for clusters can be characterized as the simultaneous minimization of the variation within clusters and the maximization of the variation between clusters. This can be done by the minimization of f_1. See [27] for details. However we are not interested in the maximization of the variation between clusters. This is the reason for involving the term f_2.

The function $\tilde{C}$ is non-smooth and non-convex. Its optimization is a difficult problem. Since the result of this optimization is a collection of artificial centers $y_1, \ldots, y_k, y_*$, we need to replace these centers by real centers $x_1, \ldots, x_k, x_*$. In the rest of this section we shall discuss some ways of searching for the real centers with the smallest cost.

5.3 Optimization approach: search for artificial centers

Assume that we have found a collection of artificial centers $y_1, \ldots, y_k, y_*$ by the minimization of the cost function $\tilde{C}$. The most natural way to find real centers is

the following: we keep the clusters that are found by the minimization; then we substitute y_i with x_i which is the node from the cluster A_i closest to y_i and we substitute y_* with the closest node x_* from all of the nodes. However, this procedure can lead to a substantial increase in the cost. One possible approach to limiting this increase is to include an extra parameter within the definition of the function $\tilde{C}$ and then to choose the value of this parameter in order to reduce the difference between the artificial cost $\tilde{C}$ and the real cost C. Recall that $\tilde{C} = f_1 + f_2$. The simplest way to include a parameter γ is to consider a modified artificial cost of the form $\tilde{C} = f_1 + \gamma f_2$. Thus we consider a function

$$\tilde{C}_\gamma(y_1, \ldots, y_k) = \sum_{a \in A} \min_{i=1,\ldots,k} \|y_i - a\|$$

$$+\gamma \sum_{i=1}^{k} \|y_i - y_*\|, \tag{8}$$

Thus the problem of finding artificial centers is formulated as:

$$\text{minimize } \tilde{C}_\gamma(y_1, \ldots, y_k)$$

$$\text{subject to}$$

$$(y_1, \ldots, y_k, y_*) \in R^{2(k+1)}.$$

Since the number of variables in function $\tilde{C}_\gamma$ is large, so general-purposed global optimization method fails to solve such problem. We can only use methods of local optimization for the minimization of function $\tilde{C}_\gamma$. This function is a saw-tooth function with a large number of shallow local minima and saddle points. We use the derivative-free discrete gradient method (see subsection 2.3) for local minimization of $\tilde{C}_\gamma$. Numerical experiments confirm that the discrete gradient method escapes from saddle points and sometimes even from shallow local minima.

The discrete gradient method shows less dependance on the choice of an initial point than the k-means method. We propose to use a combination of optimization

techniques with the k-means method. In particular, we use the results obtained by the k-means method as an initial point for optimization.

5.4 Constraint optimization: centers are real nodes

We can also use constraint optimization in order to reduce the difference between the cost with artificial centers and the real situation. Consider the following optimization problem;

$$\text{minimize} \qquad \tilde{C}(y) \quad \text{s.t} \quad h(y) = 0, \tag{9}$$

where $y = (y_1, \ldots, y_k) \in R^{2K}$ and

$$h(y) = \sum_{i=1}^{k} \min_{a \in A} \|y_i - a\|. \tag{10}$$

It is easy to check that $h(y) = 0$ is equivalent to the following: for each i there exists $a \in A$ such that $y_i = a$. This means that each y_i is a real node. We can convert the constrained problem (9) to an unconstrained problem, using the penalty function method. Namely, a solution of the following unconstrained problem

$$\text{minimize} \qquad \tilde{C}(y_1, \ldots, y_k) + \lambda h(y_1, \ldots, y_k) \tag{11}$$

is close to a solution of (9) for all sufficiently large λ. Thus we can automatically find centers of clusters $y_1, \ldots, y_k$ that are real nodes. However, the total center is still artificial and we need to find a corresponding real node to replace this artificial center.

Numerical experiments show that this approach works well if the penalty coefficient λ is not too large.

5.5 Numerical experiments

In our first numeral experiment we use the similar data set of the geographical locations of 51 North American cities presented in the paper [29]. We got this data from the picture included in the paper [29]. The picture of this data set is shown in

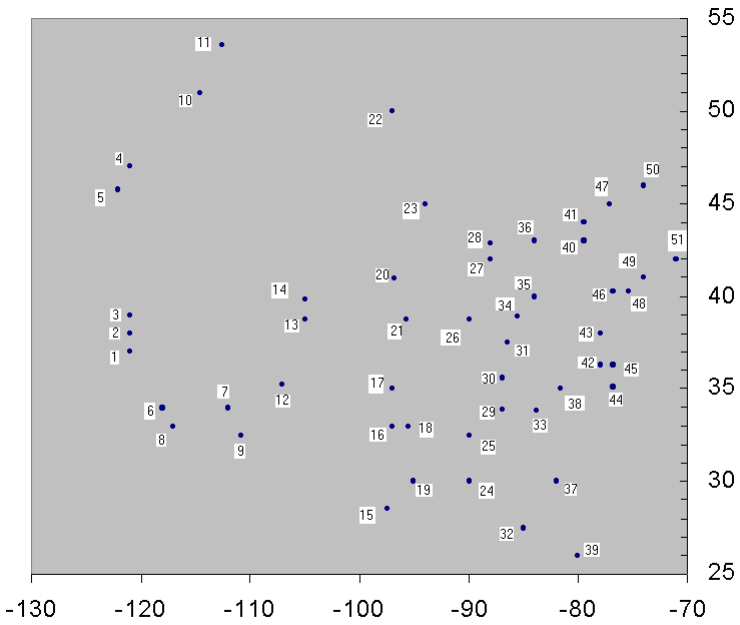

Figure 5.1: A similar data set of 51 North American cities

figure 5.1. In paper [29], the author chose $k = 6$ to maintain a maximum fan-out of six at each level. We first choose $k = 6$ within a two-level hierarchical tree topology in order to make our results comparable to those in [29]. The two-level hierarchical tree topology is shown in figure 5.2.

We designate the optimization algorithm described in Section 5.3 as op1, and the optimization algorithm described in Section 5.4 as op2.

Table 5.1 shows the relationship between the cost and the value of γ in op1. For each value of γ the left column represents the real total cost(in some units) of the tree, while the right column represents the total cost of the tree with artificial centers. Each row represents a different group of initial center nodes. Comparing the two columns under each γ we can see a big difference between the real cost (RC)(this cost is calculated by using the real nodes as centres) and artificial cost (AC) (this cost is calculated by using the artificial nodes as centres), and the minimum AC are not guaranteed to give the minimum RC. When the value of γ increases from 0.1 to 0.6, there are no changes in either RC or AC. When the value of γ is larger than 0.6,

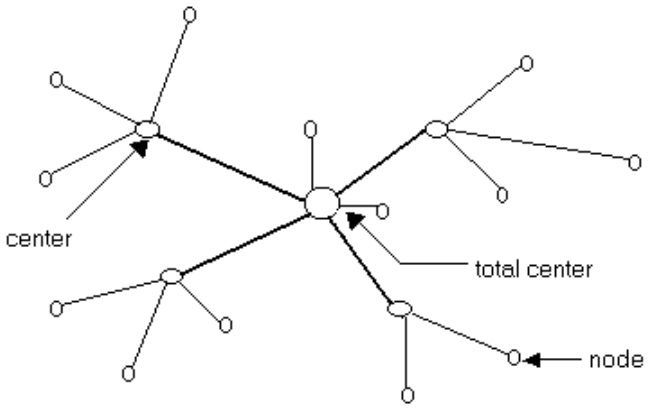

Figure 5.2: The two-level hierarchical tree topology

both AC and RC decrease.

Table 5.2 shows that the best value of γ for this example is 2.5. Comparing the results of $\gamma = 1.5$ and $\gamma = 2.5$ we find that the former gives better results with some initial nodes but the latter show less sensitivity with the initial nodes. The best result from this table is $RC = 317$, the picture is shown in Figure 5.3. When the value of γ increases (> 2.5) the cost also increases. In fact the results show that this method tends to reduce the number of clusters. If the value of γ is too large, the algorithm will try to put all nodes in one cluster in order to decrease the cost between clusters. So some clusters will have only one member, for example $\gamma = 5$, $RC = 365$, see the Figure 5.4. The more interesting result is that the value of AC becomes larger than that of RC.

Table 5.3 shows the relationship between the cost and the value of the penalty parameter. All results were obtained when $\gamma = 1$. The results show that our optimization with penalty ($op2$) is also sensitive to the choice of initial nodes. Generally speaking the values of penalty parameter λ less than 10 and more than 2 are suitable. When the value of penalty parameter is too big the algorithm will choose the initial nodes as centers and for some initial nodes there will be many clusters with only one node, and one cluster with nearly all the other nodes. That is the reason for high total cost obtained by the $op2$ method (for example, $RC = 913$ in the table).

Table 5.1: Cost comparisons: $\gamma \leq 1$

$\gamma = 0.1$		$\gamma = 0.6$		$\gamma = 1.0$	
RC	AC	RC	AC	RC	AC
401	302	401	302	325	273
422	305	422	305	370	288
390	302	390	302	363	295
355	278	355	278	325	266
355	278	355	278	325	267
345	279	345	279	317	268
342	279	342	279	325	264
337	275	337	275	323	264

Table 5.2: Cost comparisons: $\gamma \geq 1$

$\gamma = 1.5$	$\gamma = 2.5$	$\gamma = 3$	$\gamma = 5$	$\gamma = 25$
RC	RC	RC	RC	RC
325	318	345	541	633
370	325	403	541	604
363	320	345	541	583
325	325	396	439	619
325	330	345	365	633
317	318	373	378	611
325	326	373	417	574
323	327	368	385	597

In Table 5.4 we compare the results from the $op1$ method, the k-means method, and the combination of both methods, with the same initial center nodes. Here $op1 - km$ means we choose the result of centers from the k-means method as the initial center nodes for the $op1$. While $km - op1$ means we choose the result of centers from $op1$ method as the initial center nodes for k-means method. We choose 2 for the value of γ. From our numeral experiments we can draw the following conclusions:

Table 5.3: Cost comparisons: different values of λ

$\lambda = 0.3$	$\lambda = 0.9$	$\lambda = 7.2$	$\lambda = 10$	$\lambda = 50$
RC	RC	RC	RC	RC
471	434	421	491	586
452	430	385	913	913
371	374	396	364	402
349	375	396	442	488
381	382	400	406	451
352	420	425	412	353
377	376	352	384	373
469	395	340	373	381

Table 5.4: Cost comparisons: the combination of op1 ($\gamma = 2.0$) and k-means method

k-means	op1		$km - op1$	$op1 - km$	
RC	RC	AC	RC	RC	AC
374	318	273	316	366	294
344	326	274	323	324	266
344	321	273	316	325	268
344	326	275	323	324	266
318	330	279	324	325	268
323	318	273	316	317	268
320	326	279	316	324	266
338	328	274	330	324	265

the k-means method is influenced by the choice of initial centers; while the $op1$ is much less sensitive (compared with the k-means method) to the initial center nodes; the cost of the objective function with artificial nodes is still much less than the real cost. This reminds us to find a way to narrow this gap. The best result from k-means method in Table 5.4 is shown in Figure 5.5. The results from the combination of the two methods show that the improvement of the k-means method after $op1$

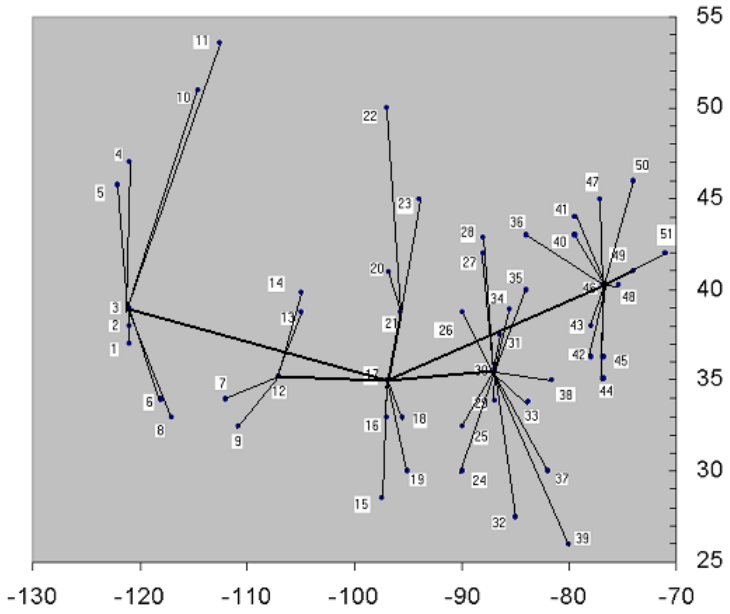

Figure 5.3: The best result from op1

$(km - op1)$ is obvious compared with the results of the k-means method only or $op1$ only. This might be a way to overcome the problem of the k-means method's strong dependence on the choice of initial nodes. But the improvement of $op1$ after the k-means method $(op1 - km)$ is not so obvious. This also indicates that the $op1$ method is not as dependent on the choice of initial nodes, if we can find a suitable value of γ.

Table 5.5 shows the results of the k-means method and the op2 with the same initial center nodes. The combinations of both methods are also shown in table 4. The results for $op2$ are under the condition of $\gamma = 2$, and $\lambda = 7.2$. For $op2 - km$ we try different values of λ in order to get the best result. We conclude that the combination of the k-means method with the op2 method $(op2 - km)$ often works efficiently and produces much better results than both results from the k-means method and the $op2$ method only. Using the $op2 - km$ method we achieved the best result over all of our experiments: $RC = 308$. The best result from $op2 - km$ method is shown in Figure 5.6. Thus the $op2 - km$ allows us to improve on the best result obtained by the k-means method (RC=318) by more than 2.5%. Although $km - op2$

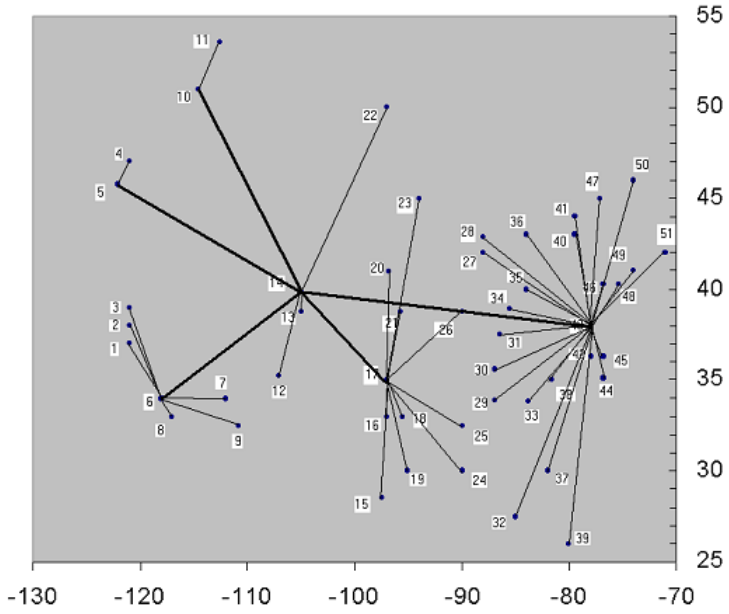

Figure 5.4: The result from op1 when $\gamma = 5$

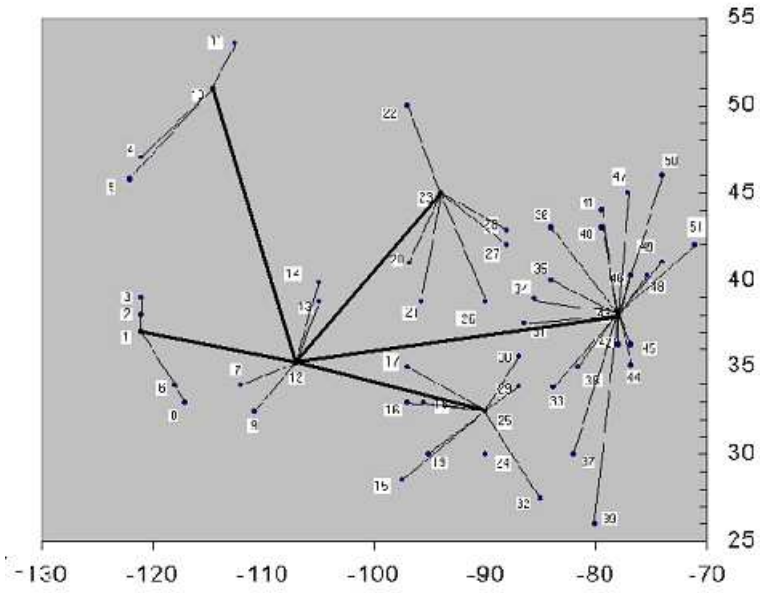

Figure 5.5: The best result from k-means

Table 5.5: Cost comparisons: the combination of op2 and the k-means method

k-means	$op2$	$op2 - km$	$km - op2$
RC	RC	RC	RC
374	392	308	371
344	415	322	371
344	392	325	322
344	389	322	317
318	390	317	312
323	352	320	325
320	384	315	336
338	383	324	333

also improved the result compared with the result obtained by op2 only, some of them are not as good as the results obtained by the k-means method only.

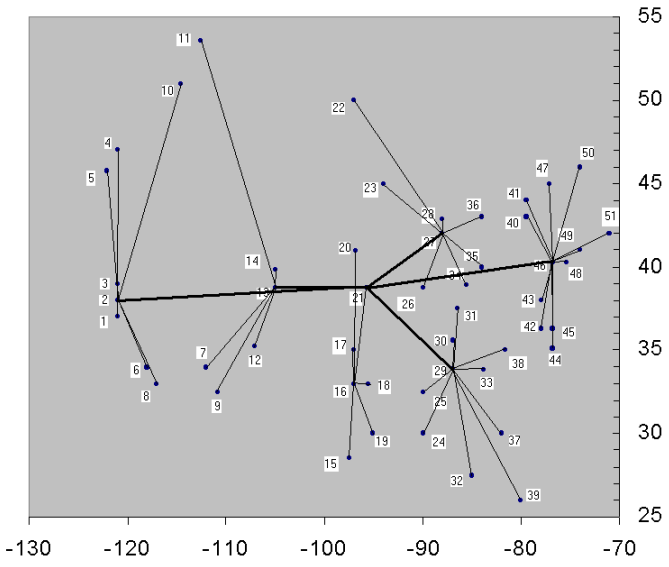

Figure 5.6: The best result from $op2 - km$

To compare the results obtained from k-means method (Figure 5.5) and $km - op2$ (Figure 5.6) and op1 method (Figure 5.3) we can see that the optimization based

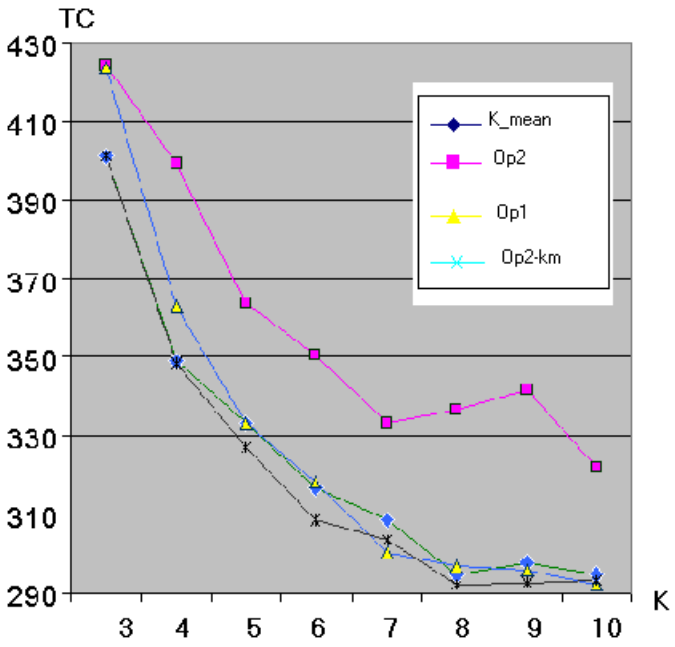

Figure 5.7: $TC(k)$ for 51-nodes

clustering (OBC) algorithms not only result in less cost, but can also give an even fan-out result, which is good to the traffic control for the network.

Finally, we used different methods to do the experiment with different k(number of clusters) for this data. This result is shown at Figure 5.7. This result shows that k-means method is the best one when used as a single method. The combination $op2 - km$ can improve the results from k-means method.

In our second numerical experiment, we use the set of the geographical locations of 88 American cities [34]. We did similar experiment as with the last data. The result is shown at Figure 5.8. As a single method, the k-means method is the best when the number of clusters k is less than 11. Then $op1$ shows better result. The $op2-km$ can improve the result from k-means method by up to more than 2 percent.

In our third numerical experiment we used a randomly generated database with 2,000 nodes on the plane. The result is presented in Figure 5.9. When k is less than 10, the k-means method is better than $op2$. Then the $op2$ method can offer more than 2.5 percent better result compared with the k-means.

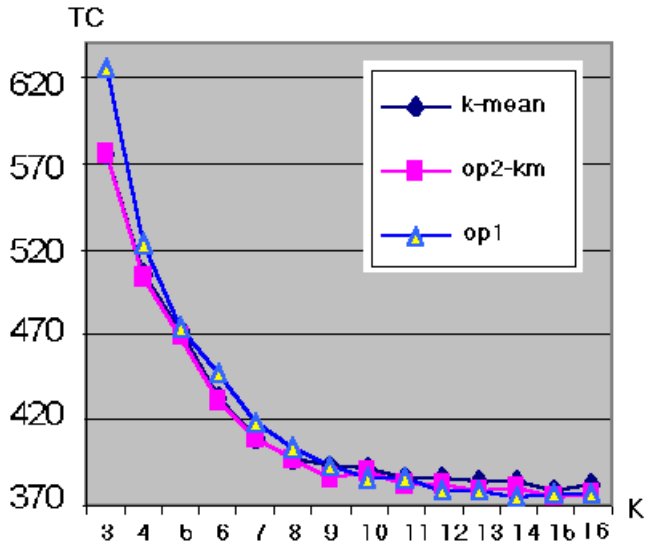

Figure 5.8: $TC(k)$ for 88-nodes

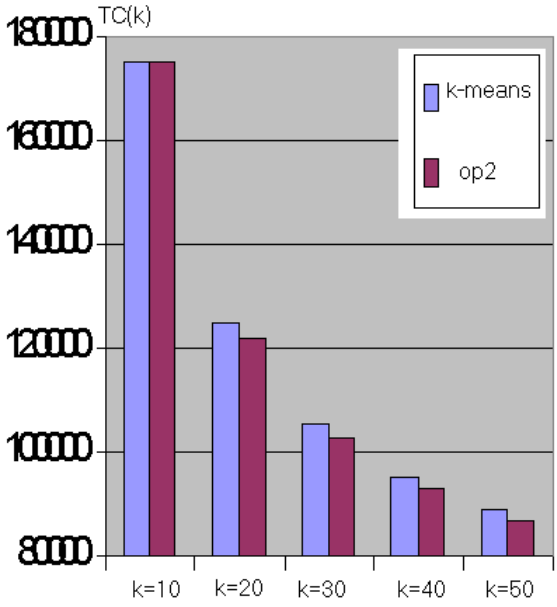

Figure 5.9: $TC(k)$ for 2000-nodes

5.6 Conclusion

In this chapter we formulated the problem of finding of networks with minimal cost as a problem of the cluster analysis. The latter problem is formulated as a non-smooth, non-convex optimization problem. We tried three clustering methods and their combinations. From our numeral experiments we conclude:

1. The k-means method, which is cheap and popular, could be used as a clustering method in network evolution.
2. The k-means method is influenced by the choice of initial centers. So an improvement of the k-means method or the combination of the k-means method with some other methods is needed.
3. Both $op1$ and $op2$ show less sensitivity to the choice of the initial center nodes, especially $op1$.
4. The combination of these methods has showed promising directions for further study.
5. The method $op2-km$ is the best method among all combination methods used in our experiments.
6. For large data, the remaining problem is how to choose suitable values of γ and λ.

The Internet is widely acclaimed for making interaction between distant locations more feasible and much faster. Over a decade ago, Gillespie and Williams (1988) [138] suggested that the Internet would allow for a certain measure of time-space convergence. When the time taken to communicate over 10,000 miles is indistinguishable from the time taken to communicate over 1 mile, then time-space convergence has taken place at a fairly profound scale. (Gillespie and Williams 1988, p. 1317)

However, many authors argue that relative location, now more than ever, plays a pivotal role in access to telecommunications infrastructure ([139], [140], [141]). For example, using a large database of commercial Internet backbones for the United States, Wheeler and OKelly [139] found that the most accessible American cities on

the Internet are located at major network access points. In another study Gorman and Malecki [141] suggest that the topology of the U.S. commercial Internet backbone favours coastal cities such as New York and San Francisco, versus interior cities such as Cleveland and Denver. In this case our algorithms can be used to solve this problem and evolve the Internet backbone in America.

Although the number of nodes and the model we use are small and simple and certain practical aspects are ignored, our objective is to gain insights into optimization clustering methods and their combinations that can be used in network evolution. Our techniques for organizing hierarchies using clustering can not only solve the scaling problem in network evolution but also can be used in multicast routing (This will be discussed in next chapter) and other services such as Reliable Multicast Transport(RMT), Quality of Service(QoS).

Chapter 6

Optimization based clustering algorithms (OBC) in Multicast group Hierarchies

6.1 Introduction

In this chapter we use the optimization based clustering algorithms (OBC) in Multicast group Hierarchies. The setting of the problem is similar to that of the previous chapter, but we use different approaches to this problem. For example, in the optimization approach search for artificial centers, we use a different objective function in this chapter. Also, our second approach here is a direct calculation of centers which is not used in the previous chapter. Furthermore we realize a three-level hierarchy and conduct related numerical experiments.

In the following sections, we discuss the problems of multicast, and the advantages of hierarchies for multicasting; we briefly introduce the concepts of hierarchical routing, and hierarchies using clusters, their constraints and optimization criteria. We also give a short review of the related work. In section 6.3, we explain why we have chosen to apply optimization based clustering (OBC) algorithms to the problem. In section 6.4, we describe a non-smooth optimization approach for finding two-level hierarchies in multicast routing. In section 6.5, we examine a non-smooth optimiza-

tion approach for finding three-level hierarchies in multicast routing. In section 6.6, we discuss algorithms for solving problems using non-smooth optimization approach. This is followed the section 6.7 where we discuss the numerical experiments that were conducted using three different databases. We compared the results obtained by our algorithms with those obtained by the algorithms from [30]. In the final section, we summarize our work, suggest the best ways of applying the techniques.

6.2 Multicast

In today's Internet, the dominant model of communication is "unicast"—the data source must create a separate copy of the data for each recipient. When there are many recipients, and when large amounts of data (e.g. streaming video) are being sent, unicast becomes prohibitively wasteful of bandwidth. The key idea behind multicast is to create each recipient's copy of each message at a point as close to that recipient as possible, thus minimizing the bandwidth consumed. Generally speaking, there are three types of multicast services, namely one-way multicast, two-way multicast, and N-way multicast. The one-way multicast service requires point to multipoint routes that start from sender to all the members in the group. In the two-way multicast, a sender sends messages to all the members of the multicast group and may also receive replying messages from the group members. In N-way multicast, any message sent by a member is multicasted to every other member of the same multicast group (More details see [152]).

In order to cater to a very large number of internetwork-wide multicast applications, it is important that the multicast routing protocol used be first and foremost scalable with respect to a network of very large size, and low-cost in terms of computational overhead and storage requirements - properties lacking in current IP multicasting techniques [79]. Furthermore, with the introduction of applications demanding quality of service (QoS), the multicast problem becomes more challenging.

The problem of providing QoS in multicast routing is difficult due to a number of factors. First, distributed continuous media applications such as teleconference, video on demand, Internet telephony, and Web-based applications have very diverse requirements for delay, delay jitter, bandwidth, and packet loss probability. Multiple

constraints often make the multicast routing problem intractable. Second, there are many practical issues that have to be taken into account when a routing algorithm is incorporated as part of a multicast routing protocol (e.g., state collection and update, handling of dynamic topology and membership changes, tree maintenance, and scalability). Figuring in QoS further complicates the protocol design process. Moreover, one has to consider how to collect/maintain QoS-related state at minimal cost, how to construct a QoS-satisfying route/tree in the presence of aggregated imprecise state information, and how to maintain QoS across routing domains [78].

In summary, the main objective of multicast communication is to supply various group communication services with required QoS while reduce the cost of data transfer (i.e. minimizing the multicast tree cost). In this thesis we try to solve this problem by using optimization based clustering algorithms to determine Multicast group Hierarchical trees. Some requirements of QoS will become our objectives and some become constraints.

To simplify the problem we first only consider a two-level constrained hierarchical multicast tree, see Figure 6.1. It is similar to the hop constrained spanning tree problem or HCSP [45]. This problem has been shown to be NP-hard [44]. We further only consider the geographical locations of the nodes: all nodes are equally weighted and the distance covered by the tree is the total cost of our tree topology network. However, the approach also allows us to consider different types of cost functions and constraints. Then we consider a three-level hierarchical multicast tree. Based on this idea we can develop more levels in the hierarchical multicast tree as needed.

We are interested in hierarchical multicast trees capable of scaling to very large groups. Our chosen strategy has the potential for adaptation to a number of different optimization criteria. These include the number of levels in the hierarchy, the size of the clusters (and thus the number of children served, which is the degree of the node.), and the level of delay to the recipients.

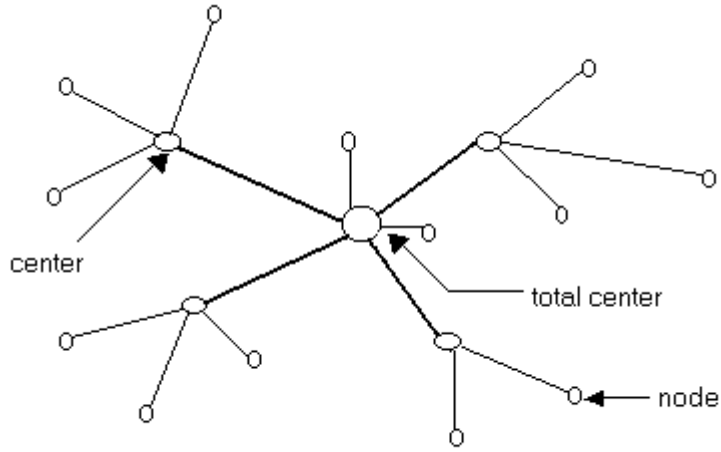

Figure 6.1: Two-level constrained hierarchical multicast tree

6.2.1 Hierarchical routing for multicast

Hierarchical structuring as a means to improve the performance of routing in large networks was first introduced by McQuillan [65] and first analyzed by Kamoun and Kleinrock in the early 80's [100, 101]. In their paper [77], Robert C. Chalmers, and Kevin C. Almeroth suggested that it is helpful to understand hierarchical structure in multicast by understanding the evolution of multicast deployment in the Internet:

- From 1992 until 1997 (the first large-scale experiments), deployment consisted of a flat, overlay network referred to as the multicast backbone (MBone) [97]. This flat routing topology was inefficient and became difficult to manage as multicast deployment increased.

- After 1997, work began to develop a hierarchical multicast infrastructure. An Autonomous System (AS) is a network under a single administrative authority. ASs connect to each other through border routers, so the Internet can be considered as consisting of interconnected ASs. Analogous to interdomain unicast routing, ASs are allowed to deploy separate intra-domain multicast routing protocols. Each AS exchanges with its peers information concerning the reachability and activity of multicast sources. Based on these exchanges, a global, interdomain distribution tree is constructed for a multicast group,

connecting the individual intra-domain trees.

- By mid-1999, the two Internet2 backbone networks, vBNS and Abilene, had deployed interdomain multicast with peering points across the US. Around this time, the existing MBone and its collection of tunnels were relegated to a special AS, AS10888. Since then, the size of the old MBone has diminished significantly [98]. Although native multicast support in commercial backbones has been slow to evolve, recent developments in Sprints backbone and several other major ISP networks have followed the deployment model of Internet2.

Hierarchical multicast trees have been shown to be easily implemented in ATM networks, using the inherent PNNI hierarchy [102]. Hierarchical trees have been introduced in the Internet community as they represent the most scalable multicast routing solution for use in large networks. Their utilization in ATM networks would allow to better optimize network resources.

In PNNI [99] and at each level, nodes which have the same address prefix form a Peer Group (PG). Each PG elects one of its node as a leader which represents the group at the higher level. This leader aggregates the information about the PG and passes it up. It forwards down the information received from higher levels on the rest of the network. At the bottom of the hierarchy, we find the physical switches. At the other levels, the nodes and the links are logical. A logical node represents the set of PGs below it and a logical link aggregates the information on the physical links between the PGs represented by logical nodes.

For multicast applications, hierarchies are useful for network layer routing and at the transport and application layers. For example, scalable reliable file distribution can be achieved with a hierarchy of systems that provide error recovery on behalf of the users at their level (Reliable Multicast Transport). Real time video distribution is an example of an application that is likely to require optimized hierarchies to scale to very large groups.

An example of reliable multicast transport is the TRAM protocol [96] with a hierarchical tree structure of “repair heads” for repairing failures in transmission to a multicast group. The IETF group on Reliable Multicast Transport (RMT) is putting together a number of building blocks which will enable the selection of different tech-

niques suited to a variety of applications (IETF). At the 47th IETF meeting in Adelaide, tree building for RMT was raised as an important issue. Requirements include techniques for optimally organizing hosts into a tree and optionally, re-organizing the trees. Solutions should be capable of scaling to 10,000 receivers with fan-out sizes from 50 to 1000. It is likely that solutions will include top-down techniques (such as discussed in the optimization approach of this paper) and bottom-up techniques (e.g. by receivers carrying out a local search for the nearest Repair Heads).

Web caching often uses hierarchical structures; these methods combine a method of detecting well-used pages with decisions about the hierarchy. For example, in the LSAM proxy cache [95], their Intelligent Request Routing uses a neighbour-search operation to configure the proxy hierarchy. Another example is the Squid Proxy Cache [94].

6.2.2 Application of optimization techniques in multicast

Application of optimization techniques was considered in [47]. This problem can be formulated as follows: Given a multicast group M and a set of possible optimization objective functions O, multicast routing is a process of constructing, based on network topology and network state, a multicast tree T that optimizes the objective functions (Figure 6.2). In the case of constraint-based multicast routing, a set of constraints C in the form of end-to-end delay bound, interreceiver delay jitter bound, minimum bandwidth, packet loss probability, and/or a combination thereof is given. The resulting multicast tree must provide not only reachability from source(s) to a set of destinations, but also certain QoS merits on the routes found in order to satisfy the constraints.

There are many performance measures that could be used as the objective of the optimization in multicast. Using different objective functions and constraints produces different optimization problems. For example, in [100], an optimal hierarchical structure was outlined in order to minimize the routing table length. These results lead to general dimensioning rules on the number of hierarchical levels and the size of the different peer-groups for a given network.

The aim of our work is to understand and quantify some aspects of the influence

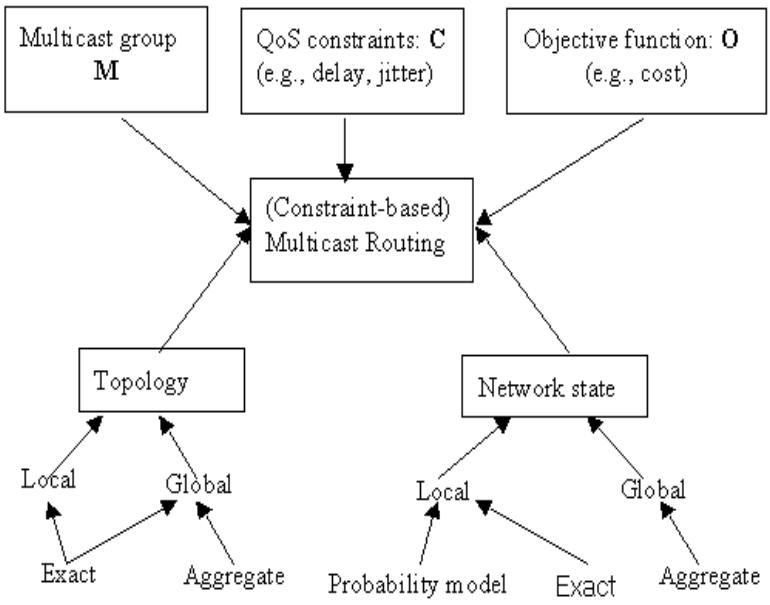

Figure 6.2: The various components in multicast routing

of optimization based clustering (OBC) process on network performance, and to determine how a given network (of a given topology) should be clustered under some optimization criteria in order to provide multicast with QoS.

6.2.3 Tree construction for multicast

A widely-used approach in solving the multicast routing problem requires tree construction. Hierarchical trees have been introduced in the Internet community as they represent the most scalable multicast routing solution for use in large networks.

The properties of a good multicast tree are:

- Low Cost: The cost of the multicast tree is the sum of costs of all individual tree links. The cost of the multicast tree should be minimized.
- Low Delay: The end-to-end delay from a source node to a group member is the sum of the delay along the tree links. The multicast protocol should try to minimize the delay for each source-destination pair.

- Scalability: It should be possible to create a multicast tree for a large number of nodes with reasonable amounts of time and resources. Each node should also be able to support a large number of trees.
- Survivability: The multicast tree should be able to survive multiple link and node failures.

Other properties for a multicast tree include loop freedom and the ability to support dynamic group membership.

Multicast trees can be classified into two categories: source-tree and shared-tree. A key difference between the them is that the source tree is optimized for source specific multicast communication, while a shared tree is optimized for communication among the whole group. We will briefly introduce some trees mainly used in multicast as follows.

Minimum Steiner Tree (MS_tT) **and Minimum Spanning Tree** (MS_pT)

In order to optimize network resources, a shared tree can be designed to minimize the overall cost consumed by its branches. The cost can represent the number of links used, the capacity reserved etc. This optimization is known as the Steiner problem [85]: given an undirected, edge-weighted graph $G(V, E)$, $T(V', E')$ is the subgraph of G that finds a spanning tree with the smallest weight among all spanning trees of T.

The optimal Steiner Tree problem has been shown to be NP-Complete. Some heuristics can be used in practice for constructing a Steiner tree. One heuristic algorithm is based on joining clusters of small trees, which contain required destination nodes, to build up a Steiner tree [43]. In paper [62], Hai Zhou discussed the efficient Steiner Tree construction based on spanning graphs. Many authors generate a Steiner tree by improving on a Minimal Spanning Tree (MS_pT) topology [60], since it was proved that a minimal spanning tree is a 3/2 approximation of a SMT [61]. Let $n = |V|$, and $m = |V'|$. If $m = 2$, MS_pT reduces to the shortest path problem. If $m = n$, MS_tT is a MS_pT.

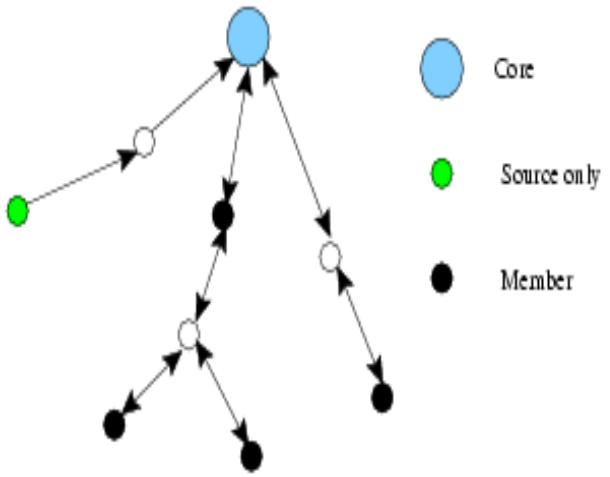

Figure 6.3: Center Based Tree (CBT)

Center Based Tree (CBT)

The most widely used multicast tree is the Center Based Tree (CBT [84], PIM [83]). With CBT, a particular node of the network is selected as the center (designated "Core" in CBT or "Rendezvous Point" in PIM). The members join the shared tree by connecting to the center (along the shortest path) — see figure 6.3. This solution reduces the information volume needed to maintain this tree, but its performance can degrade significantly if the center is not suitably placed. The Center placement problem is, however NP-Complete. For more discussion on CBT, see [73, 74]

6.2.4 ALM (Application Layer Multicast)

Initial multicast proposals focused on network layer based solutions in which multicast packet replication and forwarding mechanisms are all implemented in network routers. Indeed, network layer solutions make the most efficient use of network resources such as bandwidth. Unfortunately, global deployment of IP multicasting has been hindered for a number of practical reasons [57]. Consequently, various alternative proposals, such as simple multicast [55], source specific multicast [56] and application layer multicast (ALM)([48, 49, 50, 51, 52, 53, 54])have been in the

limelight in recent years. This is because they utilise the existing Internet unicast protocols, and facilitate instant deployment without modifying the existing network infrastructure.

ALM members form an overlay network, which is a set of unicast connections among themselves, for multicasting. Overlay multicast networks provide multicast services through a set of distributed Multicast Service Nodes (MSN), which communicate with hosts and with each other using standard unicast mechanisms. Overlay networks effectively use the Internet as a lower level infrastructure, to provide higher level services to end users.

An overlay multicast network can be modelled as a complete graph since there exists a unicast path between each pair of MSNs. For each multicast session, we create a shared overlay multicast tree spanning all MSNs serving participants of a session, with each tree edge corresponding to a unicast path in the underlying physical network [58]. The amount of available interface bandwidth at an MSN imposes a constraint on the degree of that node in the multicast tree. We let $d_{max}(v)$ denote this degree constraint at node v. In their paper [58], Sherlia Shi and Jonathan S. Turner introduced two natural formulations of the overlay multicast routing problem. The first minimizes diameter while respecting the degree constraints. They define it as a "Minimum diameter, degree-limited spanning tree problem (MDDL)":

Given an undirected complete graph G = (V;E), a degree bound $d_{max}(v) \in N$ for each vertex $v \in V$ and a cost $c(e) \in Z^+$ for each edge $e \in E$; find a spanning tree T of G of a minimum diameter, subject to the constraint that $d_{T(v)} \leq d_{max(v)}$ for all $v \in T$. The MDDL problem is NP-hard.

In this thesis we consider a similar model. However, instead of a degree-limited spanning tree, we propose a kind of level-constrained spanning tree. In fact, application layer multicasting has additional advantages: optimizations can be geared to specific requirements and the likely distribution of users may be known in advance. Thus we can use the known cities' locations and populations to do some optimizations in advance.

6.2.5 Model for Multicast

A network is modelled as an undirected graph G(V,E). Each edge (i, j) is assigned a positive cost $c_{ij} = c_{ji}$ which represents the cost to transport unit traffic from node i to node j (or from j to i). Given a multicast tree T, the total cost to distribute a unit amount of data over that tree is

$$C(T) = \sum c_{ij},\ link(i, j) \in T. \tag{6.2.1}$$

Given a multicast group g and a tree T, we say that tree T covers group g if all members of g are in tree nodes of T (i.e., in the vertex set of T). If a group g is covered by a tree T, then any data packet delivered over T will reach all members of g.

An approach to hierarchical clustering for multicast routing based on Voronoi diagrams is discussed by Baccelli et al [90]. (A Voronoi diagram divides a plane into regions each based on a specified node. Within each region, any other node is closer to that region's specified node than to the specified node in any other region.)

Another hierarchical routing scheme is that of Chatterjee and Bassiouni who describe a two level hierarchy [89]. Optimization at the top level (linking the clusters) uses a minimum spanning tree and, at the lower levels, a broadcast tree is found with optimal delay within each cluster.

In their paper [93] Waters and Sei Guan Lim form their hierarchical trees by performing k-means clustering at each hierarchical level and choosing a representative member to act as a server for each of the k clusters found. Each cluster may then be decomposed using k-means again to form a new layer of sub-clusters, whose servers become children of the server in the cluster just partitioned. The top-level servers become children of the source. The process is repeated until a suitable terminating condition is reached, as discussed below. The non-server members of any cluster that have not been partitioned become children of the clusters server and are leaf nodes in the tree. Servers receive multicast messages from their parent and pass them on to their children; they may also perform other functions (e.g. retransmission for reliable protocols). Waters and Sei Guan Lim called their technique KMC (K-Means Clustering).

Inspired by Waters' work [29, 93] we initially used optimization based clustering algorithms to solve similar problems [71, 72]. We compared several optimization based clustering methods and their combinations with the k-means method described in [71]. We formulated this problem as an optimization problem with a non-smooth, non-convex objective function as described in [72]. We introduced penalties to find the real nodes as centers and identify a real node as total center. We only considered a two-level hierarchical multicast tree in the previous two papers [71, 72]. In this thesis we will continue our work based on the work in [72] to build a three-level hierarchical multicast tree and $n - level$ hierarchical multicast tree.

To our best knowledge we have not found other work studying optimization based clustering algorithms for multicast trees. In this thesis we only consider the routing service (i.e., as a replacement for, or complement to, IP multicast) and ignore any upper-level services such as reliability or congestion control. Hierarchies based on clustering have also been useful to define scalable routing solutions for multihop wireless networks [66, 67, 68, 69].

6.3 Forming the hierarchical tree using clustering

The tree topology plays a very important role in whether the tree-first or mesh-first approach or the hierarchical tree topology is used. In this chapter we only consider the multicast routing problem as forming a hierarchical tree problem. Our algorithms will be available to form a basis for other approaches in multicast.

The basic idea to form hierarchical trees is by performing clustering algorithms at each hierarchical level and choosing a representative member to act as a center for each of the k clusters found. Each cluster may then be decomposed using clustering algorithms again to form a new layer of sub-clusters, whose centers become children of the center in the cluster just partitioned. The top-level centers become children of the source. The process is repeated until a suitable terminating condition discussed below is reached. The non-center members of any cluster that has not been partitioned become children of the clusters center and are leaf nodes in the tree. Centers receive multicast messages from their parent and pass them on to their children; they may also perform other functions (e.g. retransmission for reliable protocols).

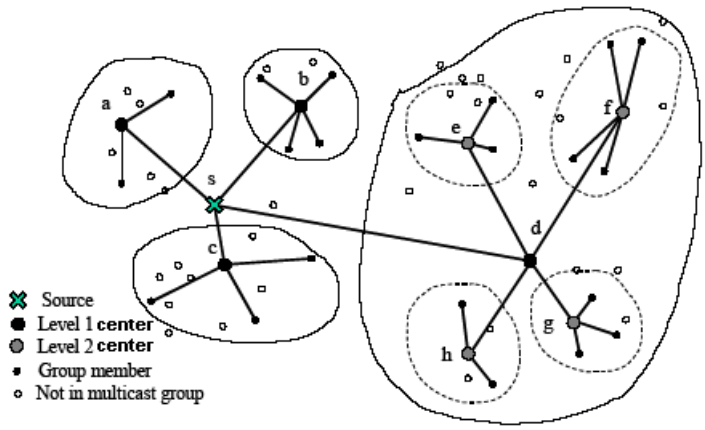

Figure 6.4: Forming multicast tree using clustering

In their paper [93], Waters and Sei Guan Lim form a multicast tree using k-means algorithm. Figure 6.4 shows an example of the tree formation for a multicast group with source s, using $k = 4$. The first application of clustering methods results in four clusters. The nearest multicast member to the center of each cluster is chosen to act as a center for that cluster. At level 1, these centers are a, b, c and d. The cluster with center d is then further decomposed into four clusters whose centers (at level 2 in the tree hierarchy) are e, f, g and h.

In our study our first approach is similar to Waters' [93]. In stead of the k-means algorithm, we use the OBC algorithm to find the artificial centers first, then choose the nearest real nodes as the real centers (6.6.1). Our second approach is a direct calculation of the real nodes as centers (6.6.2).

6.4 Two-level hierarchies

In this section we describe a non-smooth optimization approach for finding two-level hierarchies in multicast routing.

Let

$$A = \{a^1, \ldots, a^m\}, \text{ where } a^i = (a_1^i, a_2^i)$$

be a given set of nodes on the plane. We assume that the coordinates a_1^i and a_2^i

represent the geographical location of the node a^i and all nodes are equally weighted.

Our aim is to find a total center and centers of the first level in order to get a network with the least cost. We assume that

- the number k of centers in the first level is known;
- the total center is the node from the set A;
- the first level centers are nodes from the set A.

Let $x_t = x_{total}$ be a total center and $x = (x^1, \ldots, x^k) \in R^{2k}$ be a total vector of the first level cluster centers. Here $x^i \in R^2$ stands for the center of the i-th cluster. Then the cost of this network is as follows:

$$f_1(x_t, x^1, \ldots, x^k) = \sum_{i=1}^{k} \|x_t - x^i\| + \sum_{j=1}^{m} \min_{1 \le i \le k} \|x^i - a^j\|. \tag{6.4.1}$$

Here $\|\cdot\|$ is an Euclidean norm on R^2. In (6.4.1) the first term characterizes the cost between the total center and the first level centers. The second term in the expression of this function is the non-smooth optimization formulation of the clustering function (see [27, 109, 110]). It represents the cost between the first level centers and nodes from the lowest level (leafs of the hierarchical tree). Thus the first level centers are defined as centers of clusters of the set A.

The total center of the set can be found either as the centroid of the set A or as the centroid of a set of cluster centers $X = \{x^1, \ldots, x^k\}$. Consequently either

$$x_t = \frac{1}{m} \sum_{j=1}^{m} a^j \tag{6.4.2}$$

or

$$x_t = \frac{1}{k} \sum_{i=1}^{k} x^i. \tag{6.4.3}$$

Thus the function f_1 from (6.4.1) can be rewritten as

$$f_1(x^1, \ldots, x^k) = \sum_{i=1}^{k} \|x_t - x^i\| + \sum_{j=1}^{m} \min_{1 \le i \le k} \|x^i - a^j\| \tag{6.4.4}$$

where x_t is defined from (6.4.2) or (6.4.3), that is the function f_1 depends on $2k$ variables. In this scenario we will use (6.4.3) for the calculation of the total center

x_t. In this case x_t is the cluster center of the set X of the first level centers and the problem of finding the two-level hierarchies in the set A is equivalent to the two-level clustering problem.

In multicast, choosing the centroid of the set as the total center means that the multicast tree is a shared tree by the group. Otherwise we can choose the sender node as the total center if the multicast consists of only one sender with all others being receivers.

Thus the problem of the finding two-level hierarchies in the set A can be reduced to the following optimization problem:

$$\text{minimize} \quad f_1(x) \tag{6.4.5}$$

subject to

$$x_t = \frac{1}{k}\sum_{i=1}^{k} x^i \in A, \tag{6.4.6}$$

$$x^i \in A, \ i = 1, \ldots, k. \tag{6.4.7}$$

The problem (6.4.5)-(6.4.7) is reduced to the following unconstrained non-smooth optimization problem:

$$\text{minimize} \quad F_1(x) \quad \text{subject to} \quad x^i \in R^2, \quad i = 1, \ldots, k \tag{6.4.8}$$

where

$$F_1(x^1, \ldots, x^k) = \sum_{i=1}^{k} \|x_t - x^i\| + \sum_{j=1}^{m} \min_{1 \le i \le k} \|x^i - a^j\| + \tau_1 \min_{1 \le j \le m} \left\| \frac{1}{k}\sum_{i=1}^{k} x^i - a^j \right\|$$

$$+\tau_2 \sum_{i=1}^{k} \min_{1 \le j \le m} \|x^i - a^j\|.$$

In the expression for the function F_1 the first two terms represent the objective function f_1, the third term represents the penalty for (6.4.6) and the fourth term represents the penalty for (6.4.7). Both $\tau_1 > 0$ and $\tau_2 > 0$ are penalty coefficients.

6.5 Three-level hierarchies

In this section we describe a non-smooth optimization approach for finding three-level hierarchies in multicast routing.

We again assume that

$$A = \{a^1, \ldots, a^m\}, \text{ where } a^i = (a_1^i, a_2^i)$$

is a given set of nodes on the plane.

Our aim is to find a total center and centers of the first level and the second level so we describe a network with the least cost. We assume that

- the number k of centers in the first level is known;
- the maximum number q of the second level centers is known;
- the total center is the node from the set A;
- the first and second level centers are nodes from the set A.

Let $x_t = x_{total}$ be a total center, $x = (x^1, \ldots, x^k) \in R^{2k}$ be a total vector of the first level cluster centers and $y = (y^1, \ldots, y^q) \in R^{2q}$ be a total vector of the second level centers. We define the total center x_t by (6.4.3). Then the cost of this network is as follows:

$$f_2(x, y) \equiv f_2(x^1, \ldots, x^k, y^1, \ldots, y^q) = \sum_{i=1}^{k} \|x_t - x^i\| + \sum_{j=1}^{q} \min_{1 \le i \le k} \|x^i - y^j\|$$

$$+ \sum_{j=1}^{m} \min_{1 \le i \le q} \|y^i - a^j\| \tag{6.5.1}$$

In (6.5.1) the first term characterizes the cost between the total center and the first level centers. The second term is the cost between the first and second level centers. This term corresponds to the problem of finding k cluster centers in the set Y of the second level centers $Y = \{y^1, \ldots, y^q\}$. Again we use the non-smooth optimization formulation of the clustering problem. Finally, the third term represents the cost between the second level centers and the third lowest level nodes. The points $y^1, \ldots, y^q$ are the centers of q clusters in the set A. We can conclude that the search of three-level tree in the set A is reduced to the three-level clustering problem. Here the total center x_t is the only center in the set $X = \{x^1, \ldots, x^k\}$.

Thus the problem of finding three-level hierarchies in the set A can be reduced to the following mathematical programming problem:

$$\text{minimize} \quad f_2(x, y) \tag{6.5.2}$$

subject to

$$x_t = \frac{1}{k}\sum_{i=1}^{k} x^i \in A. \tag{6.5.3}$$

$$x^i \in A, \ i = 1, \ldots, k, \tag{6.5.4}$$

$$y^j \in A, \ j = 1, \ldots, q. \tag{6.5.5}$$

Here the conditions (6.5.3), (6.5.4) and (6.5.5) imply that the total center, the first and second level centers are nodes from the set A.

The problem (6.5.2)-(6.5.5) can be reduced to the following unconstrained optimization problem using penalty functions:

$$\text{minimize} \quad F_2(x, y) \tag{6.5.6}$$

subject to

$$x^i \in R^2, \quad i = 1, \ldots, k, \quad y^j \in R^2, \quad j = 1, \ldots, q \tag{6.5.7}$$

where

$$F_2(x^1, \ldots, x^k, y^1, \ldots, y^q) = \sum_{i=1}^{k} \|x_t - x^i\| + \sum_{j=1}^{q} \min_{1 \le i \le k} \|x^i - y^j\| + \sum_{j=1}^{m} \min_{1 \le i \le q} \|y^i - a^j\|$$

$$+\tau_1 \min_{1 \le j \le m} \left\| \frac{1}{k}\sum_{i=1}^{k} x^i - a^j \right\| + \tau_2 \sum_{i=1}^{k} \min_{1 \le j \le m} \|x^i - a^j\| + \tau_3 \sum_{i=1}^{q} \min_{1 \le j \le m} \|y^i - a^j\|.$$

In the expression for the function F_2 the first three terms represent the objective function f_2, the fourth term represents the penalty for (6.5.3), the fifth term represents the penalty for (6.5.4) and the sixth term represents the penalty for (6.5.5). $\tau_1, \tau_2 > 0$ and $\tau_3 > 0$ are penalty coefficients.

6.6 Solution algorithms

In this section we will discuss algorithms for solving problems (6.4.5)-(6.4.7) and (6.5.2)-(6.5.5). We will consider two approaches to solve these problems.

6.6.1 First approach: the use of artificial centers

In the first approach we use artificial centers, that is instead of problem (6.4.5)-(6.4.7) we consider the following unconstrained optimization problem:

$$\text{minimize} \quad f_1(x) \quad \text{subject to} \quad x^i \in R^2,\ i = 1, \ldots, k. \tag{6.6.1}$$

The solutions $x^{1*}, \ldots, x^{k*}$ to this problem need not be nodes from the set A. We call them *the artificial cluster centers*. First we calculate these centers solving problem (6.6.1) and find the total artificial center $\bar{x}_t$ as a centroid of the set $X^* = \{x^{1*}, \ldots, x^{k*}\}$. Then we calculate closest nodes from the set A to the artificial centers $\bar{x}_t, x^{1*}, \ldots, x^{k*}$. We accept this set of nodes as an approximate solution to the problem (6.4.5)-(6.4.7).

A similar approach can be used to find an approximate solution to the problem (6.5.2)-(6.5.5). In this case, instead of problem (6.5.2)-(6.5.5) we consider the following unconstrained optimization problem:

$$\text{minimize} \quad f_2(x, y) \tag{6.6.2}$$

subject to

$$x^i \in R^2,\ i = 1, \ldots, k,\ y^j \in R^2,\ j = 1, \ldots, q.$$

First we find the solution (x^*, y^*) to the problem (6.6.2) and calculate the artificial total center $\bar{x}_t$ as a centroid of the set $X^* = \{x^{1*}, \ldots, x^{k*}\}$. Then we calculate the closest node from the set A to $\bar{x}_t$, replace it with this node and exclude it from further consideration. In second stage, we calculate the closest nodes from the remaining set A to the points $x^{1*}, \ldots, x^{k*}$, and replace them with corresponding nodes from the set A. Finally, we calculate the closest nodes from the remaining set A to the points $y^{1*}, \ldots, y^{q*}$. We identify these centers with the closest centers from the first level centers. All other nodes are identified with the closest second level centers.

6.6.2 Second approach: direct calculation of centers

The two-level and three-level hierarchies in the set A can be found by solving problems (6.4.8) and (6.5.6), (6.5.7) directly. In this case, the choice of penalty parameters τ_1, τ_2 in the expression of the function F_1 and τ_1, τ_2, τ_3 in the expression of the

function F_2 is crucial. If these parameters are too large then penalty function in these functions becomes very large and the clustering part becomes indecisive and an algorithm can be trapped in a local minimizer which does not provide a good clustering description of the set A. Therefore the penalty parameters in functions F_1 and F_2 have to be small enough. Results of numerical experiments show best values for these parameters are $\tau_1, \tau_2, \tau_3 \in [1, 2]$. In this case it is quite possible that the final solutions to the problems (6.4.8) and (6.5.6), (6.5.7) are not nodes from the set A. If they are not they can be replaced by the nodes from the set A using a similar algorithm as in the case of the artificial centers.

6.6.3 Solving optimization problems

In this subsection we will discuss an algorithm for solving optimization problems (6.4.8), (6.5.6)-(6.5.7), (6.6.1) and (6.6.2).

The objective functions in these problems are non-smooth and non-convex, that is these problems are non-smooth global optimization problems. There are $2k$ variables in problems (6.4.8), (6.6.1) and $2(k+q)$ in problems (6.5.6)-(6.5.7), (6.6.2). However, for networks with several hundred nodes the numbers k and q can be quite large and the global optimization methods cannot be directly applied to solve problems (6.4.8), (6.5.6)-(6.5.7), (6.6.1) and (6.6.2). Therefore we will discuss algorithms for finding local minima of the functions f_1, f_2, F_1 and F_2.

All these functions are locally Lipschitz continuous however they are not Clarke regular (for the definition of the Clarke regular functions see [111]) and therefore the evaluation of subgradients of these functions is a difficult task. Consequently methods of non-smooth optimization based on subgradient information at each iteration are not effective for solving problems (6.4.8), (6.5.6)-(6.5.7), (6.6.1) and (6.6.2). Direct search methods of optimization seem to be the best option for solving these problems. Two of the most widely used are the Powell method ([114]) and Nelder-Mead's simplex method ([113]). Both methods perform well when the objective function is smooth and there are less than 20 variables. However, in problems (6.4.8), (6.5.6)-(6.5.7), (6.6.1) and (6.6.2), as a rule, the number of variables is quite large and the objective functions are complicated non-smooth functions. These two factors make

the mentioned methods not applicable for solving the problems under consideration.

We use the discrete gradient method to solve problems (6.4.8), (6.5.6)-(6.5.7), (6.6.1) and (6.6.2). The description of this method can be found in [106, 107, 108]. The discrete gradient method is a derivative-free method of non-smooth optimization. Numerical experiments confirm that the discrete gradient method escapes from saddle points and sometimes even from shallow local minima. However, this is a local search method. The discrete gradient is an approximation of a subgradient of a locally Lipschitz continuous function. The discrete gradient method can be considered as a version of the bundle method ([112]) where the discrete gradient is used instead of subgradients.

6.7 Numerical experiments

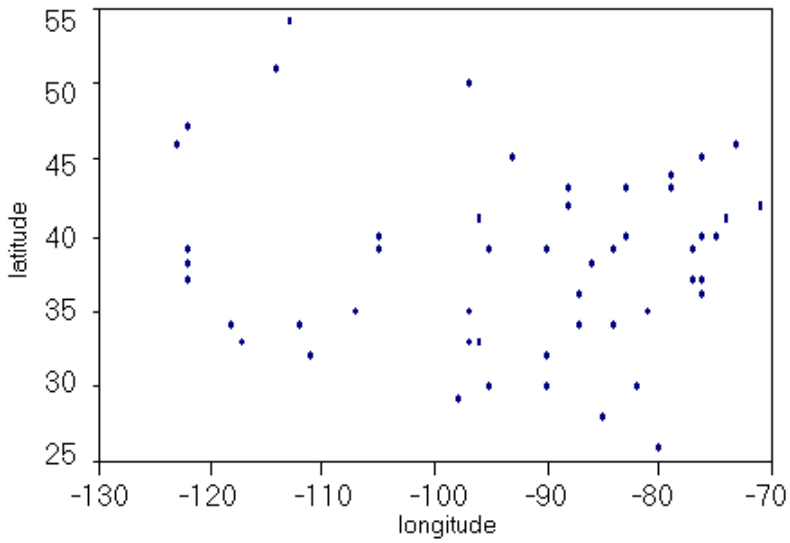

Figure 6.5: 51 cities in America

In order to verify the effectiveness of the proposed approach we carry out numerical experiments using three different databases. The first database contains the geographical locations of 51 North American cities (See Figure 6.5). The description of this database can be found in [93]. In order to provide the comparison of different

algorithms we use results from [93] where an algorithm for finding hierarchies based on k-means algorithm was developed.

The second database used in this thesis contains the geographical locations of 88 cities. The third database was randomly generated and contains 2000 points on the plane.

Results for the first database are presented in Table 6.1. These results show that the algorithm based on the direct calculation of centers, as a rule, generates trees with less cost than the algorithm which uses artificial centers. However, this is not always the case. We can also see a big difference between the costs of two-level and three-level multicast hierarchies. Results from Table 6.1 show that the proposed algorithm performs better than algorithms based on k-means algorithm.

Figures 6.6 and 6.7 present the two-level and the three-level hierarchies in this database obtained by the proposed algorithm.

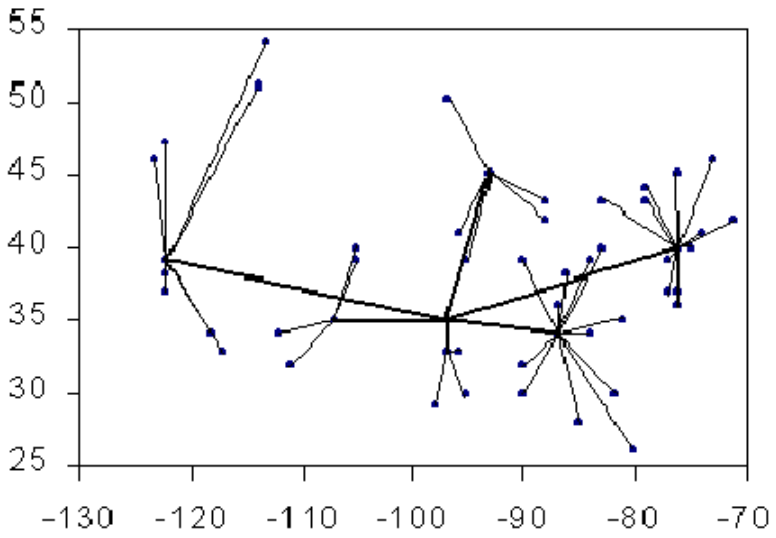

Figure 6.6: Two-level hierarchical multicast tree

To compare with results from paper [93], we put their results in figure 6.8, 6.9 and 6.10. Waters' algorithms are heavily depend on the initial points which can be found in the Figure 6.8 and Figure 6.9: different initial points result in a different topology and different cost. Our algorithms in this section do not depend on any initial points. Although we get a similar two-level multicast tree to Waters', but our

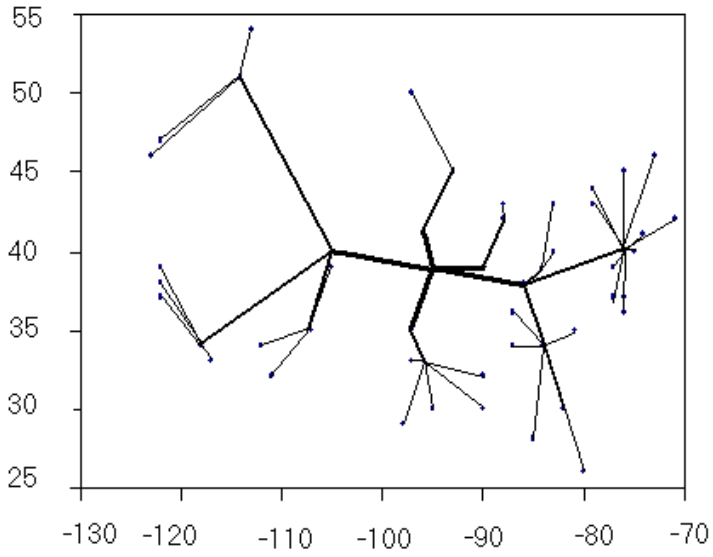

Figure 6.7: Three-level hierarchical multicast tree

result is better with less cost.

Table 6.1: Results for the network with 51 cities

No. of 1-st level centers	Cost: 1-st approach		Cost: 2-nd approach		Results from [93]	
	2-level	3-level	2-level	3-level	2-level	3-level
2	484.55	272.63	486.98	272.63	-	-
3	406.25	249.08	406.25	255.57	-	-
4	357.44	246.64	376.49	246.61	-	-
5	337.41	243.86	337.41	238.43	-	-
6	314.30	243.86	314.20	230.40	316.14	296.34

Results for the network with the geographical locations of 88 cities are presented in Table 6.2. We can see that the algorithm based on the direct calculation of centers generates trees with less cost, except in the case where there are 2 first level centers. Results from this table show that the three-level hierarchies are much more efficient than the two-level hierarchies.

Results for the network with the artificial nodes are given in Table 6.3. For this

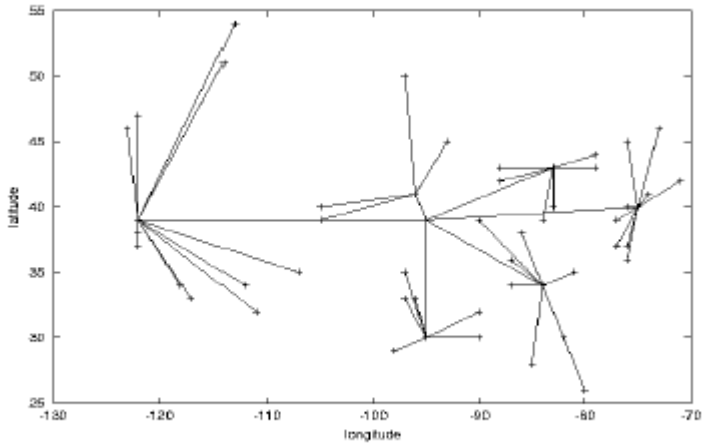

Figure 6.8: Two-level multicast tree one from Waters' paper

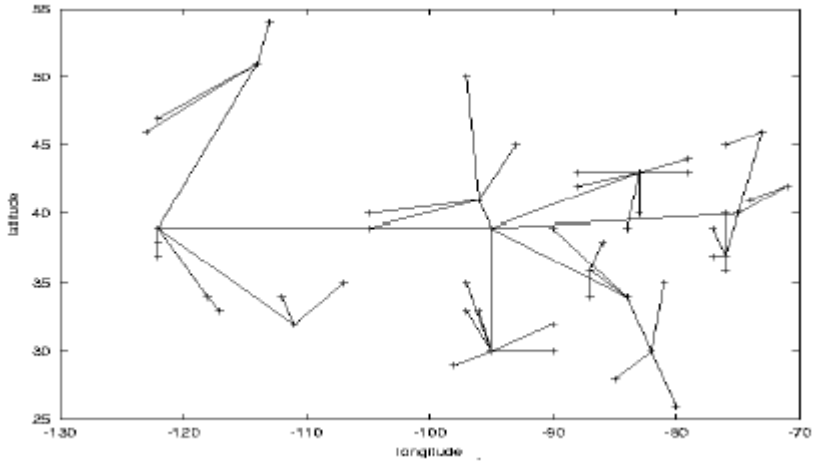

Figure 6.9: Two-level multicast tree two from Waters' paper

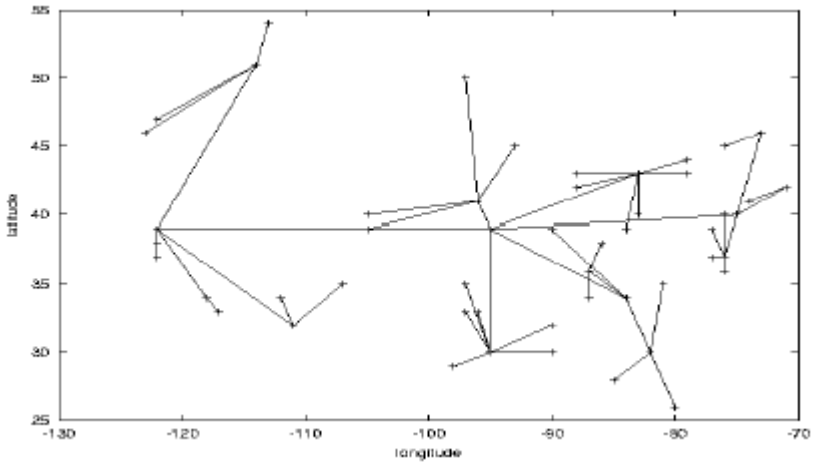

Figure 6.10: Three-level multicast tree from Waters' paper

Table 6.2: Results for the network with 88 cities

No. of 1-st level centers	Cost: 1-st approach		Cost: 2-nd approach	
	2-level	3-level	2-level	3-level
2	789.18	394.96	789.18	396.42
3	621.32	347.80	589.00	333.45
4	518.12	309.39	517.07	306.76
5	486.20	297.02	486.20	292.29
6	448.13	296.97	453.38	292.29

network the performance of the algorithm based on the use of artificial centers and the algorithm based on the direct calculation of centers are similar, however the calculation of the objective function in the first case is significantly cheaper. Results from this table again confirm that the three-level hierarchies are much more efficient than the two-level hierarchies.

We summarize the results of the comparison of our algorithms and Waters' in Table 6.4. In this table CFLS means center of first level servers.

Table 6.3: Results for the network with artificial nodes

No. of 1-st level centers	Cost: 1-st approach		Cost: 2-nd approach	
	2-level	3-level	2-level	3-level
2	383882.09	191917.12	384078.15	191917.12
3	323494.66	160224.60	323494.66	160286.38
4	287138.11	135901.55	287138.11	135728.92
5	252712.80	121624.25	252420.10	121005.05
6	226493.03	112069.79	226493.03	110726.61
7	207447.49	103901.05	207447.47	102855.74
8	192628.36	97538.62	192628.36	96983.50
9	183810.66	92608.78	183810.66	93160.24
10	176659.05	90411.71	175535.10	88685.65

Table 6.4: Cost comparisons: our algorithms and Waters' method

Initial centers	Source	number of levels	cost(Waters)	Cost(OBC1)	Cost(OBC2)
(no clustering)	Overall center	1	749.64	739.36	739.36
Waters one	CFLS	3	296.34	221	221
As above	As above	2	345.53	316.14	316
As above	As above	1	768.82	739.36	739.36
Waters two	CFLS	3	316.02	0	0
As above	As above	2	373.06	316.02	0
As above	As above	1	953.96	739.36	739.36

6.8 Conclusion

In this chapter a new non-smooth optimization based clustering (OBC) algorithm has been developed to find multi-level hierarchies in multicast routing. We described two different versions of this algorithm. The first version is based on the use of artificial centers whereas in the second version the centers are calculated directly

using penalty functions. In the first version of the algorithm the calculation of the objective function is much cheaper. Results of numerical experiments show that in many cases the performance of these two versions of the algorithm is similar, however in some cases the second version works better. Therefore the first version of the algorithm can be used when the evaluation of the objective function in the second version is too expensive. Results of numerical experiments show the effectiveness of the proposed algorithm. These results also show that the multi-level hierarchies are much more effective than the two-level hierarchies.

Our center-based trees may not provide the most optimal paths between members of a group; The most obvious point of vulnerability of a center-based tree is its center, whose failure can result in a tree becoming partitioned. Having multiple centers associated with each tree solves this problem (though at the cost of increased complexity). The idea of vice-centers was used by some authors in order to provide a more reliable multicast routing. Our algorithms allow us to do the same thing with some changes. We would consider this as worthy of a further study.

And lastly we would like to point out that our OBC algorithms studied both in the previous chapter and this chapter can be used in both applications.

Conclusion and future work

This thesis contains two research topics: 1) optimization based stochastic and queueing models in network corrective maintenance 2) optimization based clustering (OBC) algorithms for network evolution and multicast routing. The common point between these two topics is the application of optimization. Our goal is to solve some problems in Telecommunications and the Internet using optimization based methods. The main contribution of this study is to develop algorithms for solving some problems from telecommunications and the Internet based on derivative-free methods optimization, including discrete gradient method and cutting angle method developed by Dr. Adil Bagirov and Professor Alex Rubinov. Inspired by H. S. Gan and Professor A. Wirth's work and Dr. Gill Waters' work we investigate different approaches to solve the network corrective maintenance and network evolution and multicast routing problems.

7.1 Networks corrective maintenance

Making a link between telecommunications business objectives and the requirements typically stated for network corrective maintenance is a challenge. Our models in Chapter 3 and Chapter 4 are intended to help service providers and network operators automate their business process in a cost- and time- effective way by using an optimization approach.

In chapter 3 we solved the so-called simplified model (SM) [1] with a direct method. We introduced a new version of the SM model, which is based on binomial and Poisson distributions. It is hard to take the time factor into account in the stochastic approach. We proposed another approach in chapter 4.

Stochastic programming approach is emphasizing the static state of this problem, so it is good for a decision maker to consider it as a long term monitoring approach. While the queueing approach takes the sort of dynamic states into account and it also reduces this problem to a certain optimization problem.

We can extend our models by considering more properties of other equipments such as routers, bridges. Furthermore, we also can combine more complicated events together, such as effects of backlog, non-homogeneity of repair persons' skills and variation in travel distances, and so on. Thus more complicate and practical models

could be studied further. As an important part of TMN model (the Telecommunications Management Network model produced by ITU-T), our models can be easily applied by telecommunications companies into their TMN models.

Our models also can be easily adapted into the Internet networks maintenance. In paper [155] authors quoted other research results and wrote: a median network availability equivalent to a breakdown of 471 min/year while the average breakdown in telecommunications networks is less than 5 min/year.

This area is not completed if only a study the corrective maintenance without preventive maintenance is done. So as an appendix we introduce preventive maintenance. We believe that it has the same importance as corrective maintenance. In some preventive maintenance study [204] both queueing theory and hierarchical modelling were used. We could extend our study to this area because of we used the similar approaches.

7.2 Networks evolution and multicast routing

To solve the scaling problems in networks evolution and multicast routing optimization based clustering (OBC) algorithms has been implemented in chapter 5 and chapter 6. This problem is formulated as an optimization problem with a non-smooth, non-convex objective function. Different algorithms are examined for solving this problem. Results of numerical experiments using some artificial and real-world databases proved these approaches are available. In particular, the comparison of the results obtained from our algorithms and from Waters' show the improvement.

In this thesis we assumed that the all nodes are equally weighted. The case when different nodes are differently weighted will be the subject of our further research. The problem of finding of hierarchies with more than three levels will also be a subject of our future work.

For further research we need to do more numerical experiments with a large number of nodes (real database if available) and try to find a way to choose the values of γ and λ. The possible combination of both optimization approaches will also be considered. We also need to consider a hierarchical tree topology with more levels,

similar fan-outs at each level, and unequally weighted nodes which can represent such things as the traffic of a network and population of a city.

We use cost-based optimization in our OBC networks evolution and multicast routing tree algorithms in this thesis. Our algorithms could be modified to a delay-based optimization. In this case the idea of introducing a parameter γ in previous chapter can be used to emphasize the important influence of the trunks between the centers and total center. The even fanout constraint which is very important for bandwidth intensive applications is not considered and will also be the subject of future work.

To increase the reliability of the center-based trees, the idea of vice-centers was used by some authors in order to provide a more reliable multicast routing. Our algorithms allow us to take this idea into account. We would consider this as worthy of a further study.

Finally, this thesis introduce the idea of Gold Service, Silver Service and Bronze Service in modelling and the "toy bricks" method to solve in chapter 3. These can be combined into the other part of thesis, such as chapter 4.

7.3 Optimization in telecommunications networks preventive maintenance

7.3.1 Introduction

In this section we will briefly introduce and give a short review of preventive maintenance. There are several key factors in preventive maintenance: on-line preventive maintenance, inspection, and maintenance databases. The object of introducing preventive maintenance is two-fold: (1) to increase the reliability of the networks and (2) to avoid failure in the operation of networks, which may be costly and dangerous. As mentioned before, the nine hour breakdown of AT&T's long-distance telephone network in January 1990 resulted in a $60 million to $75 million loss in AT&T's revenues [8].

7.3.2 Preventive maintenance (PM)

Preventive maintenance is a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. "Much as the name implies, preventive maintenance, often abbreviated PM, refers to performing proactive maintenance in order to prevent system problems. This is contrasted to diagnostic or corrective maintenance, which is performed to correct an already-existing problem. Anyone who has ever owned or cared for a car knows all about what preventive maintenance is. After all, you don't change your oil and air filter in response to a problem situation (normally), you do it so that your engine will last and you won't have car troubles down the road.[199]" Preventive maintenance activities include equipment checks, partial or complete overhauls at specified periods, oil changes, lubrication and so on. In addition, workers can record equipment deterioration so they know to replace or repair worn parts before they cause system failure. Recent technological advances in tools for inspection and diagnosis have enabled even more accurate and effective equipment maintenance. The ideal preventive maintenance program would prevent all equipment failure before it occurs [11].

7.3.3 Value of Preventive Maintenance

There are multiple misconceptions about preventive maintenance (PM). One such misconception is that PM is unduly costly. This logic dictates that it would cost more for regularly scheduled downtime and maintenance than it would normally cost to operate equipment until repair is absolutely necessary. This may be true for some components; however, one should compare not only the costs but the long-term benefits and savings associated with preventive maintenance. Without preventive maintenance, for example, costs for lost production time from unscheduled equipment breakdown will be incurred. Also, preventive maintenance will result in savings due to an increase of effective system service life. Long-term benefits of preventive maintenance include:

- Improved system reliability.
- Decreased cost of replacement.

- Decreased system downtime.
- Better spares inventory management.

Long-term effects and cost comparisons usually favor preventive maintenance over performing maintenance actions only when the system fails [200].

7.3.4 Optimization in PM

There are many system performance measures that could be used as the objective of the optimization in PM. For example, authors in paper [205] investigated five maintenance policies: predictive maintenance policy, reactive policy, opportunistic policy, time-based preventive policy, and MTBF-based preventive policy.

There are many different optimization techniques in preventive maintenance. For example, in paper [201] Joseph B. Keller discussed the optimum inspection policies to minimize the expected loss due to down time plus the cost of checking. Paper [202] dealt with the cost analysis of a one-unit repairable system subject to on-line preventive maintenance and/or repair.In paper [203] authors addressed optimal PM for manufacturing systems.

In Xiaodong Yao's thesis [204] they attempted to consider problems of optimal preventive maintenance explicitly under the context of unreliable queueing and production-inventory systems. They proposed a two-level hierarchical modeling framework for PM planning and scheduling problems. In the higher level, the objective is to characterize structure of optimal PM policies. They started with a simple case in which queueing is not taken into account in the model. They showed that a randomized PM policy, like the widely used time-window policy in industry, is in general not optimal. They then consider the problem of optimal PM policies for an M/G/1 queueing system with an unreliable server. The decision problem is formulated as a semi-Markov decision process .

7.4 Optimization in service men resource management

7.4.1 Introduction

When a customer wants a leased line he/she asks the operator for a certain bandwidth, but also for the "five nines" (i.e., an availability of 99.999 percent). The network operator can guarantee this availability only if the telecommunications network is equipped and maintained with mechanisms that are able to cope with failures.

In chapter 3 and 4 we studied the Telecommunications Network Maintenance—Corrective maintenance and we mentioned the preventive maintenance in last section. This study focused on finding how many repairmen were necessary. But how should the service men resource be managed in daily life to provide the best service to customers: to minimize the breakdown time, i.e., to guarantee availability? This results in a new problem called service men resource management. Service men resource management is one of the most important, yet confusing topics in telecommunications today. It is understandable that most firms continue to deal with it at the level of individual managers in an ad hoc, and largely intuitive manner without adequate methodology of optimization that available from mathematics and economics. Because this problem is a complicated one, which may include integer programming, dynamic programming and global optimization. We need employ different approaches and strategies to model this problem, solve it, and develop software to help telecommunications companies in their daily operations.

7.4.2 Setting of the problem

The devices that are involved in communication are most commonly referred to as hosts and nodes. These are the terminating devices, that is, the originating and final destinations for the communication. Those devices that the communication may encounter on its journey between hosts may also be referred to as nodes and these include repeaters, bridges, routers, gateways, and switches. All types of devices are connected by means of a cable or with the use of electromagnetic waves, and these

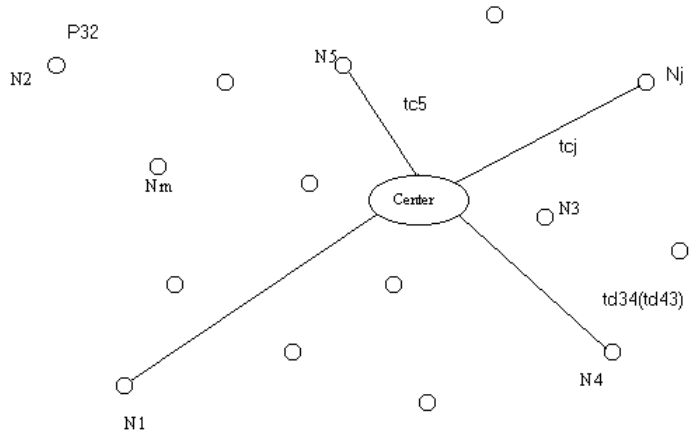

Figure 7.11: Telecommunications facilities distribution network

are referred to as channels and links. A telecommunications itself is referred to as data and this exists in the form of bits, bytes, frames, packets, and messages.

In our model, we call all the devices facilities. They are identified according to their importance in the model. We consider a telecommunications company with a map of facilities distribution as Figure 7.11. There are s service men, m facilities, and one information center, which will collect information of failure and assign jobs to service men as a control center.

In Figure 7.11, tcj is the the time needed to go to the facility number j from the center. P32 means that the person 3 among the s service men is assigned to the job to fix facility number 2.

There are several objectives. The first objective may be: to minimize the time taken to finish the first s jobs. In this case, our problem can be formulated as minimize the time taken to finish all jobs and subject to Time limitations for different jobs

Rule one: There are many different types of problems related to different facilities. We will assume that different repair personnel are able to fix different problems. And different time limitations apply to different facilities.

Rule two: The telecommunications service should be a 24-hours per day service.

The second objective:

Most companies would like to utilize ideas from the field of economics since it provides the conceptual understanding of profit, the cost, and the risk. Now we will use the cost objective function, and the constraints will be met by paying the penalty if breakdown time exceeds the time limitation. The formula can be:

minimize the cost of profit loss and penalty

The impact of a single failure to some clients can be catastrophic or it can affect a large number of services. The penalty will limit the break down time to some clients.

Rule three: According to the facility's importance or the clients which it serves we will define several service levels. For example, the center switch is the highest level, then the routers, and hubs and so on. The hub which links to the bank will also be treated as high level.

The third objective can be the mix of the above objectives.

7.4.3 Database

We can set up the database based on the companies' database.

Input Triggers

1. The number of facility and the location.
2. The symptoms.
3. The time limitation.

Output Triggers

1. The possible problems.
2. The weights for repair persons (who is good at it).
3. The possible solution.
4. Inventory management.
5. The update.

7.4.4 The principles of rules or policies

The rules or policies should

1. encourage repair personnel to update the data base.
2. take into account the impact of sudden event.
3. yield effective and simple management.
4. take into account the multiple priorities for different clients.
5. encourage repair personnel to improve their skills.
6. allow the managers to examine the way in which all repair personnel are being fully used.
7. provide the necessary information for adjusting the models to improve effectiveness.

7.4.5 Aims and expected outcomes

The goals of this research are

1. To help managers to make rules or policies by supporting them with optimization based methods or tools.
2. To support the service level agreements.
3. The development of Operation Research models for finding efficient daily repair men resource management policies for a telecommunications company.
4. The study of corresponding problems of dynamic and discrete optimization and development of algorithms for solving these problems.
5. The development software for finding the most efficient way to manage repair personnel in order to provide the best service to customers.
6. To meet the changes in new technique and customer demands.
7. To lower the skill level required for the repair personnel and the managers.

7.5 Optimization in bandwidth planning

7.5.1 Introduction

The knapsack problem is a well-known and well-studied problem in combinatorial optimization [16][17] (Also see subsection 2.1.2). Many real life optimization problems can be modelled as knapsack problems, for example capital budgeting, network planning, cutting stock, and cargo loading. In general, knapsack problems are NP-hard . The unbounded knapsack problem (UKP), where an unlimited number of items of each type is available, is also NP-hard . In this section, we try to use a special case of the unbounded knapsack problem that is characterized by a set of simple inequalities that relate item weights to item costs in our bandwidth planning. We hope that further study can find more applications that can be modelled by knapsack problems in telecommunications optimization.

7.5.2 The problem

The core of a communications services provider is service. The key objectives are 'more for less '-faster service introduction, improved quality of service at a lower cost. On the other side, the users want to get as much service as possible with lower cost. The bandwidth management is an important way for a company to make the best use of the telecommunications while expending less money. Here we try to use a special case of the unbounded knapsack problem that is characterized by a set of simple inequalities that relate item weights to item costs in our bandwidth planning. In fact we use the results from the paper [12]. For example: Telstra provides such ATM services in Figure 7.12.

We can see that broadband affords economy of scale. The price of one 4 Mbit/s is cheaper than the price of two 2 Mbit/s; The price of one 300 Mbit/s is cheaper than the price of two 155 Mbit/s. How can we optimize our choice if we need 2 Gbit/s?

7.5.3 The model

We may use the following formulation: (knapsack problem)

Table 2.1 - Annual Interface Charge				
METRO (Capital city of each state/Territory)				
Kbps	Metro local calling area		Metro Non-local calling area up to (and including) 50km	
	GST excl.	GST incl.	GST excl.	GST incl.
2Mbit/s UNI	$27,168.00	$29,884.80	$45,480.00	$50,028.00
Nominal 4M UNI	$37,832.00	$41,615.20	$63,320.00	$69,652.00
Nominal 6M UNI	$48,496.00	$53,345.60	$81,160.00	$89,276.00
Nominal 8M UNI	$59,160.00	$65,076.00	$99,000.00	$108,900.00
Nominal 12M UNI	$66,840.00	$73,524.00	$111,840.00	$123,024.00
Nominal 16M UNI	$71,400.00	$78,540.00	$119,520.00	$131,472.00
34Mbit/s UNI	$90,000.00	$99,000.00	$150,720.00	$165,792.00
45Mbit/s UNI	$90,000.00	$99,000.00	$150,720.00	$165,792.00
155M bit/s UNI	$120,000.00	$132,000.00	$201,000.00	$221,100.00
200M bit/s UNI	$135,700.00	$149,270.00	$227,280.00	$250,008.00
250M bit/s UNI	$151,400.00	$166,540.00	$253,560.00	$278,916.00
300M bit/s UNI	$167,100.00	$183,810.00	$279,840.00	$307,824.00
350M bit/s UNI	$182,800.00	$201,080.00	$306,120.00	$336,732.00
400M bit/s UNI	$198,500.00	$218,350.00	$332,400.00	$365,640.00
450Mbit/s UNI	$214,200.00	$235,620.00	$358,680.00	$394,548.00
500M bit/s UNI	$229,900.00	$252,890.00	$384,960.00	$423,456.00
550Mbit/s UNI	$245,600.00	$270,160.00	$411,240.00	$452,364.00
622M bit/s UNI	$261,300.00	$287,430.00	$437,530.00	$481,283.00

Figure 7.12: The table of annual interface charge

$$\text{minimize } Z \equiv \sum_{k=1}^{n} c_k x_k \tag{7.5.1}$$

subject to

$$\sum_{k=1}^{n} a_k x_k \geq B \tag{7.5.2}$$

$$x_k \text{ is nonnegative integers}, \qquad k = 1, 2, 3, ... \tag{7.5.3}$$

Where c_k is the price (cost coefficients) and a_k= rates (weigh coefficients). How many these UNIs do we need to meet our need B(threshold, 2 Gbit/s in our example)? We will define a new special case of the UKP that is characterized by a set of fairly simple inequalities that involve the cost and weight coefficients.

7.5.4 A special case of this problem

Suppose there are weights a_k and a_l and costs c_k and c_l with $a_k \leq a_l$ and $c_k \leq c_l$. Then in any feasible solution one may replace all occurrences of the kth item by occurrences of the lth item, without increasing the objective value and without making the solution infeasible. Therefore we will from now on assume without loss of generality that

$$a_k < a_{k+1} \; for \; k = 1, ..., n-1 \tag{7.5.4}$$

and

$$c_k < c_{k+1} \; for \; k = 1, ..., n-1 \tag{7.5.5}$$

Our polynomially solvable special case of the UKP is defined by the following set of inequalities (which implicitly assume that (7.5.4) and (7.5.5) hold):

$$c_{k+1} < \lfloor a_{k+1}/a_k \rfloor c_k \; for \; k = 1, ..., n-1. \tag{7.5.6}$$

Although the conditions in (7.5.6) make the UKP polynomially solvable, they are still fairly close to NP-hard versions of the UKP. For example, if we replace the Moor-function in (7.5.6) by ordinary brackets, then the resulting special case of the UKP is still NP-hard; this follows from the fact that even the UKP with $a_k = c_k$ for $k = 1, ..., n$ is NP-hard [13].

In paper [12] we proved that: unbounded knapsack instances whose cost and weight coefficients fulfill the conditions (7.5.4), (7.5.5) and (7.5.6) can be solved in linear time.

Bibliography

[1] Heng Soon Gan and Andrew Wirth, *Telecommunications Networks Ownership Cost Optimisation*, Proc. of The 16th National Conference The Australian Society for Operations Research in conjuction with Optimization Day, 2001.

[2] H. S. Gan and S. H. Lim, *Research Project 2000: Telstra Network Ownership Cost Optimisation*, Department of Mechanical and Manufacturing Engineering, The University of Melbourne, 2000.

[3] W. S. Soo and P. V. Truong, *Research Project 1999: Efficient Maintenance Policies for a Telecommunications Network (with Telstra Research Laboratories)*, Department of Mechanical and Manufacturing Engineering, The University of Melbourne, 1999.

[4] K. H. Wang and B. D. Sivazlian, *Cost Analysis of the M/M/R Machine Repair Problem with Spares Operating Under Variable Service Rates*, Microelectronics and Reliability 32, pp 1171-1183, 1992.

[5] K. H. Wang, *Cost Analysis of the M/M/R Machine-Repair Problem with Mixed Standby Spares*, Microelectronics and Reliability 33, pp 1293-1301, 1993.

[6] K. H. Wang and H. C. Lee, *Cost analysis of the cold-standy M/M/R machine repair problem with multiple models of failure*, Microelectronics and Reliability 38, pp 435-441, 1998.

[7] Donald R. Byrkit, *Statistics Today — A comprehensive Introduction*, The Benjamin/Cummings Publishing Company, Inc, Menlo Park, California, 1985.

[8] Aiko Pras, *Network management* , [CTIT Ph. D.-thesis series No. 95-02], ISSN 1381-3617 / ISBN 90-365-0728-6, 1995.

[9] Yunquan-Hu, Yaohuang-Guo, *Operations research textbook*, (Chinese), Tsinghua University press, Beijing, China, 1998.

[10] Brian D. Bunday, *Basic Queueing Theory*, Edward Arnold Ltd, 1986.

[11] P. Lyonnet, *Maintenance Planning Methods and mathematics*, Chapman & Hall, 1991.

[12] Moshe Zukerman, Long Jia, Timothy Neame, Gerhard J. Woegingerb, *A polynomially solvable special case of the unbounded knapsack problem*, Operations Research Letters, www.elsevier.com/locate/dsw, 2001.

[13] G.S. Lueker, *Two NP-Complete Problems in Nonnegative Integer Programming,* Report No. 178, Computer Science Laboratory, Princeton University, 1975.

[14] Fred Glover, Darwin Klingman, and Nancy V. Phillips, *Network Models in Optimization and their applications in practice*, John Wiley & Sons, INC., 1992.

[15] NEOS, http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/index.html

[16] S. Martello, P. Toth, *Knapsack problems: Algorithms and Computer Implementations*, Wiley, England, 1990.

[17] G.L. Nemhauser, L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, 1988.

[18] W. Feller, *An Introduction to probability Theory and its Application*, Wiley, New York, 1957.

[19] George H. Weiss, *A problem in equipment maintenance*, Management Science, 1962.

[20] R. Cleroux, S. Dubuc and C. Tilquin, *The age replacement problem with minimal repair and random repair cost*, Operational Research, 27, 1158-1167, 1979.

[21] M. Berg and R. Cleroux, *A marginal cost analysis for an age replacement policy with minimal repair*, INFOR, 20, 258-263, 1982.

[22] P. P. Gupta, R. K. Gupta and R. K. Sharma, *Cost analysis of a two-unit standby redundant electronic system with critical human errors*, Microelectronics and Reliability, 26, 841-846, 1986.

[23] G. L. Newhanser, A. H. G. Riuuooy Kan, M. J. Todd (eds), Amsterdam, *Optimization* , Elsevier Science, North Holland, 1989.

[24] R. Horst, P. M. Pardalos and N. V. Thoai, *Introduction to global optimization*, Kluwer Academic Publishers, Dordrecht, 1995.

[25] Samir Chatterjee and Mostafa A. Bassiouni, *Hierarchical Message Dissemination in Very Large WAN's*, IEEE Computer Socity, Technical committee on computer communications, 1992

[26] G.B. Dantzig. *Linear programming under uncertainty.*Management Science1, pp.197-206, 1955.

[27] A.M. Bagirov, A.M. Rubinov and J. Yearwood, A global optimization approach to classification, *Optimization and Engineering*, 3, 2002, 129-155.

[28] F. Baccelli, D.Kofman, J. L. Rougier, *Self organizing hierarchical multicast trees and their optimization*, Proceedings of IEEE Inforcom'99, Vol. 3, pp1081-1089, 1999.

[29] Gill Waters, *Hierarchies for network evolution*, Sixteenth UK Teletraffic Symposium on " Management of Quality of Service - the New Challenge", Harlow, UK, May 2000

[30] Gill Waters, *Applying clustering algorithms to multicast group hierarchies*, private communication.

[31] Hongyi Li, Hung Keng Pung, and Lek Heng Ngoh, *A survey of multicasting Protocols for Multimedia Communication*, MMM'95, also at http://lucan.ddns.comp.nus.edu.sg/Publications/hkp-pub/conf/mmm95_hkp_mc_survey.pdf

[32] Hagouel Jacob, *Issues in routing for large and dynamic networks*, Ph.D thesis, Columbia University, 1983

[33] Leonard Kleinrock and Farouk Kamoun, *Hierarchical routing for large networks_Performance evaluation and optimization*, Computer networks, pp155-174,1977

[34] Mark S. Daskin, *Network and discrete location models, algorithms, and applications*, John Wiley&Sons, 1995

[35] Matthew J. Moyer, Josyula R. Rao and Pankai Rohatgi, *A survey of security issues in multicast communications*, IEEE Network, Novermber/December 1999

[36] Peter Scheuermann, Geoffrey Wu, *Heuristic Algorithms for Broadcasting in Point-to-Point Computer Networks*, IEEE Transaction on computers, vol. c-33, No. 9, Sep. 1984

[37] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V.P.Kumar, *Hierarchical Multicast Routing in ATM Networks*, IEEE International Conference on Communications, 1996

[38] Samir Chatterjee and Mostafa A. Bassiouni, *Hierarchical Message Dissemination in Very Large WAN's*,17th Conference on Local Computer Networks, IEEE computer Society Press,1992

[39] Suman Banerjee,Bobby Bhattacharjee, *Scalable secure group communication over IP multicast* , IEEE Journal on selected areas in communication, Vol. 20, No.8, October 2002

[40] Vachaspathi Peter Kompella, *Multicast Routing Algorithms for Multimedia Traffic*, Dissertation, University of California, San Diego, 1993

[41] W.T.TSAI, C. V. Ramamoorthy, Wei k. Tsai and Osamu Nishiguchi *An Adaptive Hierarchical Routing Protocol*, IEEE Transactions on computers, Vol. 38, No. 8, August 1989

[42] Zhengying, Wang, Bingxin, Shi and Wei, Liu *A Distributed Dynamic Heuristic for Delay-Constrained Least-Cost Multicast Routing*,Journal of Interconnection Networks, Dec2000, Vol. 1 Issue 4,

[43] X. Jiang, *Routing Broadband Multicast Streams*, Computer Communications, Vol. 15, 1, 1992

[44] Hussein F. Salama, Douglas S. Reeves Yannis Viniotis *The Delay-Constrained Minimum Spanning Tree Problem*, In second IEEE symposium on computers and communications (ISCC'97), July 1997.

[45] Prabhu Manyem, *Constrained Spanning and Steiner Trees with Triangle Inequality* [Presented at the ASOR (Australian Society of Operations Research) meeting, September 2001. Published by Kluwer Academic Publishers

[46] R. Krishnan, R. Ramanathan, and M. Steenstrup, *"Optimization algorithms for large self-structuring networks,"* in INFOCOM: The Conference on Computer Communications, joint conference of the IEEE Computer and Communications Societies, 1999.

[47] J. Hou and B. Wang, *Multicast routing and its QoS extension: Problems, algorithms, and Protocols,* in IEEE Network, Jan./Feb.2000.

[48] Helder, D.A. and Jamin, S., *End-host Multicast Communication Using Switch-trees Protocols.* in Workshop on Global and Peer to Peer Computing on Large Scale Distributed System (GP2PC), (2002).

[49] Francis, P. *Yoid: Extending the Internet Multicast Architecture* , ISI, 2000, 38.

[50] Chu, Y.H., Rao, S.G. and Zhang, H., *A Case for End System Multicast.* in ACM SIGMETRICS, (Santa Clara, CA, 2000), ACM, 1-12.

[51] Banerjee, S., Bhattacharjee, B. and Kommareddy, C., *Scalable Application Layer Multicast.* in ACM SIGCOMM, (Pittsburgh, PA,2002), ACM.

[52] Jannotti, J., DGifford, D., Johnson, K., Kaashoek, M. and O'Toole, J., *Overcast: Reliable Multicasting with an Overlay Network.* in Symposium on Operating Systems Design and Implementation (OSDI), (San Diego, California, USA, 2000).

[53] Li, Z. and Mohapatra, P., *HostCast: A New Overlay Multicast Protocol.* in IEEE International Conference on Communications (ICC), (Anchorage, Alaska, USA, 2003), IEEE.

[54] Rowstron, A. and Kermarre, A.M, *Scribe: A Large-scale and Decentralized Application-level Multicast Infrastructure.* IEEE JSAC, 20 (8).

[55] Perlman, R., Lee, C., Ballardie, T., Crowcroft, J., Wang, Z., Maufer, T., Diot, C., Thoo, J. and Green, M, *Simple Multicast: A Design for Simple, Low-overhead Multicast.*, IETF, 1999.

[56] Holbrook, H. and Cain, B, *Source-specific Multicast for IP,* IEFT, 2000.

[57] C. Diot, B.N. Levine, B. Lyles, H. Kassem, and D. Balensiefe, *"Deployment issues for the IP multicast service and architecture,"* IEEE Network, vol.14, pp.88–98, Jan. 2000.

[58] Sherlia Shi and Jonathan S. Turner, *"Routing in Overlay Multicast Networks"*, IEEE INFOCOM, New York City, June 2002.

[59] Kin-Ching Chan and S. -H. Cary Chan, *Distributed Servers Approach for Large-Scale Secure Multicast*, IEEE Journal on selected areas in communications, vol. 20, No. 8, October 2002.

[60] J. M. Ho, G. Vijayan, and C. K. Wong. *New algorithms for the rectilinear steiner tree problem.* IEEE Transactions on Computer Aided Design, 9: 185193, 1990.

[61] F. K. Hwang, *On Steiner minimal trees with rectilinear distance.* SIAM Journal on Applied Mathematics, 30: 104114, 1976.

[62] Hai Zhou. *Efficient Steiner Tree Construction Based on Spanning Graphs.* In ACM International Symposium on Physical Design, Monterey, CA,2003.

[63] Matula, D. W., *Cluster Analysis via Graph Theoretic Techniques,* In Proceedings of Louisiana Conference on Combinatorics, Graph Theory and Computing, K. B. Reid, and D. P. Roselle, Editors, University of Manitoba, Winnipeg, 1970, pp.199212.

[64] Matula, D. W., *k-Components, Clusters and Slicings in Graphs,*SIAMJ. Appl. Math., 22(3), 1972, pp.459480.

[65] J. McQuillan, *Adaptive Routing Algorithms for Distributed Computer Networks,* BBN Report No. 2831, May 1974.

[66] M. Gerla and J.T.-C. Tsai, *Multicluster, mobile, multimedia radio network,* ACM-Baltzer Journal of Wireless Networks, vol. 1, no. 3, 1995.

[67] C.R. Lin and M. Gerla, *Adaptive clustering for mobile, wireless networks,* Journal on Selected Areas of Communication, vol. 15, no. 7, Sept. 1997.

[68] S. Basagni, I. Chlamtac, and A. Farago, *A generalized clustering algorithm for peer-to-peer networks,* Workshop on Algorithmic Aspects of Communication, July 1997.

[69] B. Das, R. Sivakumar, and V. Bhargavan, *Routing in ad-hoc networks using a spine,* International Conference on Computers and Communications Networks, Sept. 1997.

[70] L.Wei and D.Estrin, *The Trade-offs of Multicast Trees and Algorithms.* Proceedings of the Third International Conference on Computer Communications and Networking (IC3N'94), pages17–24,1994.

[71] L. Jia, I. Ouveysi, A. M. Rubinov *A comparison of optimization methods in Multicast group Hierarchies*, The 5th International Congress on Industrial and Applied Mathematics, ICIAM 2003, Sydney, Australia, July 2003.

[72] L. Jia, A. Bagirov, I. Ouveysi, A. M. Rubinov *Optimization based clustering algorithms in Multicast group Hierarchies*, accepted by Australian Telecommunications, Networks and Applications Conference (ATNAC), Melbourne, Australia, 2003

[73] C. Shields and J.J. Garcia-Luna-Aceves, *"The Ordered Core Based Tree Protocol"*, Proc. IEEE INFOCOM 97, Kobe, Japan, April 7-11, 1997.

[74] A. Chakrabarti, A. Striegel, G. Manimaran *A Case for Tree Evolution in QoS Multicasting* Proc. of Int'l Workshop on QoS (IWQoS),2002

[75] Garey, M.R., Johnson D.S, *Computers and Intractability: A Guide to the Theory of NPCompleteness.* W.H. Freeman, San Francisco (1979)

[76] Narsingh Deo and Ayman Abdalla *Computing a Diameter-Constrained Minimum Spanning Tree in Parallel*

[77] Robert C. Chalmers,and Kevin C. Almeroth *On the Topology of Multicast Trees*, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 1, FEBRUARY 2003

[78] Ayman El-Sayed,Vincent Roca,Laurent Mathy, *A Survey of Proposals for an Alternative Group Communication Service*IEEE Network. Jannuary/February 2003

[79] Tony Ballarie, Paul Francis, Jon Crowcroft *Core Based Trees (CBT) An Architecture for Scalable Inter-Domain Multicast Routing*, Conference proceedings communications architectures, protocols, and applications, September 13-17, 1993 San Francisco, California

[80] L. Aguilar, *Datagram Routing for Internet Multicasting,* ACM Computer Communications Review, Vol.14, No.2, 1984, pp.58-63.

[81] S. E. Deering, D. R. Cheriton, *Multicast Routing in Datagram Internetworks and Extended LANs,* ACM Transactions on Computer Systems, Vol. 8, No. 2, May 1990, pp. 85-110.

[82] G. N. Rouskas and I. Baldine. *Multicast routing with endtoend delay and delay variation constraints* IEEE Journal on Selected Areas in Communications, 15(3): 346356, April 1997.

[83] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. *Protocol Independent MulticastSparse Mode (PIMSM) Protocol Specification.* Internet draft, September 1997, Work in progress.

[84] T. Ballardie, B. Cain, and Z. Zhang, *Core Based Trees (CBT version 3) Multicast Routing Protocol Specification.* Internet draft, March 1998, Work in progress.

[85] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees.* SIAM Journal on Applied Mathematics, 16(1): 129, January 1968.

[86] B. W. Waxman, *Routing of multipoint connections* IEEE Journal on Selected Areas in Communications, 6(9): 1617 1622, December 1988.

[87] L. Kou, G. Markowsky, and L. Berman, *A fast algorithm for Steiner trees.* Acta Informatica, 15: 141145, 1981.

[88] M. Doar and I. Leslie, *How bad is naive multicast routing.* IEEE INFOCOM'93, pp.8289, 1993.

[89] Chatterjee, S. and M. A. Bassiouni (1992), *Hierarchical message dissemination in very large WANs.* IEEE Computer Soc. Press: 397-403.

[90] Bacelli, F., D. Kofman, et al. (1999), *Self Organizing Hierarchical Multicast Trees And Their Optimization.* INFOCOM,IEEE.

[91] Robert S. Cahn, *Wide Area Network Design: concepts and tools for optimization* , Morgan Kanufmann, 1998

[92] Sharma, S. (1996), *Chapter 7 Clustering Algorithms* Applied Multivariate Techniques, Wiley.

[93] Gill Waters and Sei Guan Lim, *Applying clustering algorithms to multicast group hierarchies* Technical Report No. 4-03 August 2003, University of Kent, Canterbury, Kent CT2 7NF, UK

[94] *Squid The Squid Web Proxy Cache.* http://squid.nlanr.net/

[95] Touch, J. and A. S. Hughes, *LSAM proxy cache: a multicast distributed virtual cache* Computer Networks and ISDN Systems 30: 2245-2252.1998

[96] Chiu, D. M., S. Hurst, et al, *TRAM: A tree-based reliable multicast routing protocol*, Sun Microsystems Laboratories. SML TR-9896,1998.

[97] K. Almeroth, *The evolution of multicast: From the MBone to interdomain multicast to Internet2 deployment*, IEEE Network, pp.1020, Jan./Feb. 2000.

[98] P. Rajvaidya and K. Almeroth, *Analysis of routing characteristics in the multicast infrastructure* ,University of California, Santa Barbara, Tech. Rep., Jan. 2002.

[99] *The ATM Forum Technical Committee. Private NetworkNetwork* Interface Specification Version 1.0. March 1996.

[100] L.Kleinrock, F.Kamoun. *Hierarchical Routing for Large Networks* Computer Networks, vol.1. 1977.

[101] L.Kleinrock, F.Kamoun. *Stochastic Performance Evaluation of Hierarchical Routing for Large Networks* Computer Networks, vol.3. 1979.

[102] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V. Kumar. *Hierarchical Multicast Routing in ATM Networks* In IEEE Intl. Conf. on Communications, volume 3, pages 16901694, June 1996.

[103] A. G. Waters, *A new heuristic for ATM multicast routing*, In 2nd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, July, 1994

[104] G. Ballintijn, M. van Steen, A.S. Tanenbaum, *Characterizing Internet Performance to Support Wide-area Application Development*, Operating Systems Review, 34(4): 41-47, 2000.

[105] Mike Tanner, *Practical Queueing Analysis* McGraw Hill, 1995.

[106] A.M. Bagirov, Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis, *Investigacao Operacional*, 19, 1999, 75-93.

[107] A.M. Bagirov, Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices, In: A. Eberhard et al. (eds.) *Progress in Optimization: Contribution from Australasia*, Kluwer Academic Publishers, 1999, 147-175.

[108] A.M. Bagirov, A method for minimization of quasidifferentiable functions, *Optimization Methods and Software*, 17(1), 2002, 31-60.

[109] A.M. Bagirov, A.M. Rubinov, N.V. Soukhoroukova and J. Yearwood, Supervised and unsupervised data classification via nonsmooth and global optimisation, TOP: Spanish Operations Research Journal, Vol. 11, No. 1, 2003, 1-93.

[110] H. H. Bock, *Automatische Klassifikation*, Vandenhoeck & Ruprecht, Gottingen, 1974.

[111] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

[112] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms,* Vol. 1 and 2, Springer Verlag, Berlin, Heidelberg, New York, 1993.

[113] J.A. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.,* Vol. 7, 1965, 308-313.

[114] M.J.D. Powell, UOBYQA: unconstrained optimization by quadratic approximation, *Mathematical Programming*, Series B, 92(3), 2002, 555-582.

[115] Bacelli, F., Kofman D. and Rougier J.L., *Self-organising hierarchical multicast trees and their optimization* , Proc. of IEEE INFOCOM 99: Conference on Computer Communications, March 1999, pp 1081-1089

[116] Callon R., P. Doolan P., Feldman N., Fredette A., Swallow G. and Viswanathan A., *A Framework for Multiprotocol Label Switching,* Internet Draft September 1999. n.b. For the latest document see http://www.ietf.org/ID.html

[117] Chatterjee, S and Bassiouni, MA, *Hierarchical message dissemination in very large WANs,* IEEE Computer Soc. Press, 13-16 Sept 19992, pp 397-403

[118] Chiu, D-M, Hurst, S, Kadansky M. and Wesley, J, *TRAM: A tree-based reliable multicast routing protocol,* Sun Microsystems Laboratories technical report SML TR-9896, July 1998

[119] Francis, Paul, *Yallcast: Extending the Internet Multicast Architecture* , September 1999, available from http://nttsl.mfeed.ne.jp/francis/yallcast/

[120] Braden, R., Clark D. and Shenker S., *Integrated Services in the Internet Architecture: an Overview* , Internet Request for Comments1633 (Informational), June 1994

[121] Braden, R., Zhang L., Berson S., Herzog S. and Jamin S., *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification,* Internet Request for Comments 2205 (Standards Track), September 1997

[122] S. Blake S., D. Black, D., Carlson M., Davies E., Wang Z. and Weiss W., An Architecture for Differentiated Services, Internet Request for Comments 2475 (Informational), December 1998

[123] Sharma, Subhash, Applied Multivariate Techniques, John Wiley and Sons, 1996

[124] Sripanidkulchai, K, Myers, M and Zhang, H., Reliable Multicast Recovery Structures, August 1999, see http://www.cs.cmu.edu/~

kunwadee/research/rm-trees/

[125] M. Pioro, A. Jutner, J. Harmatos, Szentesi. A, P. Gajowniczek, and Myslek A. *Topological design of telecommunications networks - nodes and links localization under demand constraints.* submitted to 17th International Teletraffic Congress, Salvador de Bahia, 2001. "http://www.tele.pw.edu.pl/ amyslek/papers/itc2001.pdf".

[126] F.R.B. Cruz, J. MacGregor Smith, and G.R. Mateus, *Algorithms for a multi-level network optimization problem.* European Journal of Operational Research, 118(1): 164-180, 1999.

[127] J. Current and H. Pirkul. *The hierarchical network design problem with transshipment facilities.* European Journal of Operational Research, 51(3): 338-47, 1991.

[128] Ioannis Gamvros, Bruce Golden, and S. Raghavan. *An evolutionary approach to the multi-level capacitated minimum spanning tree problem.* Sixth INFORMS Telecommunications Conference, 2002.

[129] J.R. Current, *The design of a hierarchical transportation network with transshipment facilities.* Transportation Science, 22(4): 270-7, 1988.

[130] J.R. Current, C.S. ReVelle, and J.L. Cohon, *The hierarchical network design problem.* European Journal of Operational Research, 27(1): 57-66, 1986.

[131] Andre Girard, Brunilde Sanso, and Linda Dadjo. *A tabu search algorithm for access network design.* Annals of Operations Research, 106(1-4): 229-262, 2001.

[132] Luis Gouveia and Joao Telhada, *An augmented arborescence formulation for the two-level network design problem.* Annals of Operations Research, 106(1-4): 47-61, 2001.

[133] J. Petrek and V. Siedt, *A large hierarchical network star-star topology design algorithm.* European Transactions on Telecommunications, 12(6): 511-22, 2001.

[134] M. Pioro and T. Szymaski. *Basic reconfiguration options in multi-layer robust telecommunications networks - design and performance issues.* Teletraffic Engineering in the Internet Era, pages 271-284, 2001.

[135] H. Pirkul, J. Current, and V. Nagarajan. *The hierarchical network design problem: a new formulation and solution procedures.* Transportation Science, 25(3): 175-82, 1991.

[136] N.G.F. Sancho. *A suboptimal solution to a hierarchial network design problem using dynamic programming.*European Journal of Operational Research, 83(1): 237-244, 1995.

[137] N.G.F. Sancho. *The hierarchical network design problem with multiple primary paths.* European Journal of Operational Research, 96(2): 323-328, 1997.

[138] Gillespie A, Williams H *Telecommunications and the reconstruction of regional comparative advantage.* Environment and Planning A 20: 13111321,1988

[139] Wheeler D, OKelly ME *Network topology and city accessibility of the commercial Internet.* Professional Geographer 51: 327339(1999)

[140] Moss ML, Townsend A *The Internet backbone and the American metropolis.* Information Society Journal 16: 3547 (2000)

[141] Gorman S, Malecki E *The networks of the Internet: An analysis of provider networks in the USA.* Telecommunications Policy 24: 113134(2000)

[142] Duc A. Tran, Kien A. Hua, and Tai T. Do, "Scalable media streaming in large peer-to-peer networks," in ACM Multimedia Conference, Juan Les Pins, France, December 2002.

[143] Birge, J.R. [1997]. *"Stochastic Programming Computation and Applications,"* INFORMS J. on Computing, 9, pp. 111-133.

[144] Birge,J.R. and F.V. Louveaux [1997] *Introduction to Stochastic Programming* , Springer, New York.

[145] Carino, D.R., T. Kent, D.H. Meyers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, and W.T. Ziemba [1994]. *"The Russell-Yasuda Kasai Model: An asset/liability model for a Japanese insurance company using multistage stochastic programming,"* Interfaces, 24, pp.29-49.

[146] Fisher, M., J. Hammond, W. Obermeyer, and A. Raman [1997]. *"Configuring a supply chain to reduce the cost of demand uncertainty,"* Production and Operations Management, 6, pp.211-225.

[147] Kall, P. and S.W. Wallace [1994]. *Stochastic Programming,* John Wiley & Sons, Chichester, England.

[148] Murphy, F.H., S. Sen and A.L. Soyster [1982]. *"Electric utility capacity expansion planning with uncertain load forecasts,"* AIIE Transaction, 14, pp. 52-59.

[149] Prekopa, A. . *Stochastic Programming,* Kluwer Academic Publishers, Dordrecht, 1995.

[150] Sen, S. R.D. Doverspike and S. Cosares [1994]. *"Network Planning with Random Demand,"* Telecommunications Systems, 3, pp. 11-30.

[151] Sen,S. and J.L. Higle [1999]. *"An Introductory Tutorial on Stochastic Linear Programming Models,"* Interfaces, 29, pp. 33-61.

[152] Ralph Wittmann, Martina Zitterbart *Multicast Communication: Protocols, Programming, and Applications*, Morgan Kaufmann publishers, 2001

[153] C.Barakat, J.L. Rougier *Optimization of Hierarchical Multicast Trees in ATM Networks*, in proceedings of IFIP ATM'98, Ilkley (UK), July 98.

[154] P. F. Tsuchiya *The landmark hierarchy: a new hierarchy for routing in very large networks*, Applications, Technologies, Architectures,and Protocols for Computer Communication, Symposium proceedings on Communications architectures and protocols. Pages: 35 - 42, Stanford, California, United States, 1988

[155] Pablo Molinero-Fernndez , Nick McKeown , Hui Zhang, *Is IP going to take over the world (of communications)?* , ACM SIGCOMM Computer Communication Review, v.33 n.1, p.113-118, January 2003

[156] K.S. Al-Sultan, A tabu search approach to the clustering problem, *Pattern Recognition*, 28(9) (1995), 1443-1451.

[157] K.S. Al-Sultan and M.M. Khan, Computational experience on four algorithms for the hard clustering problem, *Pattern Recognition Letters*, 17, (1996), 295-308.

[158] Andramonov, M.Y., Rubinov, A.M. and Glover, B.M., Cutting angle method for minimizing increasing convex-along-rays functions, Research Report 97/7, SITMS, University of Ballarat, 1997.

[159] Andramonov, M.Y., Rubinov, A.M. and Glover, B.M., Cutting angle method in global optimization, *Applied Mathematics Letters* **12** (1999) 95-100.

[160] A.M. Bagirov, A method for minimization of quasidifferentiable functions, *Optimization Methods and Software*, 17(1) (2002), 31-60.

[161] Bagirov A.M. and Rubinov, A.M., Global minimization of increasing positively homogeneous functions over the unix simplex, *Annals of Operations Research* **98** (2000) 171-187.

[162] Bagirov, A.M. and Rubinov, A.M., Modified versions of the cutting angle method, in N. Hadjisavvas and P. M. Pardalos, eds., *Advances in Convex Analysis and Global Optimization*, Kluwer Academic Publishers, Dordrecht, 2001, 245-268.

[163] A.M. Bagirov, A.M. Rubinov and J. Yearwood, Using global optimization to improve classification for medical diagnosis and prognosis, *Topics in Health Information Management* 22(2001) 65-74.

[164] Batten, L.M. and Beliakov, G., Fast algorithm for the cutting angle method of global optimization, *Journal of Global Optimization*, Vol. 24, Issue 2, 2002, 149-161.

[165] H.H. Bock, Clustering and neural networks, In: A. Rizzi, M. Vichi and H.H. Bock (eds), *Advances in Data Science and Classification*, Springer-Verlag, Berlin, (1998), 265-277.

[166] D.E. Brown and C.L. Entail, A practical application of simulated annealing to the clustering problem, *Pattern Recognition*, 25(1992), 401-412.

[167] V.F. Demyanov and A.M. Rubinov, *Constructive Nonsmooth Analysis*, Peter Lang, Frankfurt am Main, 1995.

[168] I.S. Dhillon, J. Fan and Y. Guan, Efficient clustering of very large document collections, In: *Data Mining for Scientific and Engineering Applications*, Kluwer Academic Publishers, Dordrecht, 2001.

[169] G. Diehr, Evaluation of a branch and bound algorithm for clustering, *SIAM J. Scientific and Statistical Computing*, 6(1985), 268-284.

[170] R. Dubes and A.K. Jain, Clustering techniques: the user's dilemma, *Pattern Recognition*, 8(1976), 247-260.

[171] O. du Merle, P. Hansen, B. Jaumard and N. Mladenovic, An interior point method for minimum sum-of-squares clustering, *SIAM J. on Scientific Computing,* 21, 2001, 1485-1505.

[172] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugenics,* VII part II (1936), 179-188. Reprinted in R.A. Fisher, *Contributions to Mathematical Statistics,* Wiley, 1950.

[173] P. Hanjoul and D. Peeters, A comparison of two dual-based procedures for solving the p-median problem, *European Journal of Operational Research,* 20(1985), 387-396.

[174] P. Hansen and B. Jaumard, Cluster analysis and mathematical programming, *Mathematical Programming,* 79(1-3) (1997), 191-215.

[175] P. Hansen and N. Mladenovic, J-means: a new heuristic for minimum sum-of-squares clustering, *Pattern Recognition*, 4, (2001), 405-413.

[176] P. Hansen and N. Mladenovic, Variable neighborhood decomposition search, *Journal of Heuristic,* 7, (2001), 335-350.

[177] P. Hansen, E. Ngai, B.K. Cheung and N. Mladenovic, Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering, submitted

[178] J.-P. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms,* Vol. 1 and 2, Springer-Verlag, Berlin, New York, 1993.

[179] D.M. Houkins, M.W. Muller and J.A. ten Krooden, Cluster analysis, In: *Topics in Applied Multivariate Analysis*, Cambridge University press, Cambridge, 1982.

[180] A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: a review, *ACM Computing Surveys* 31(3)(1999), 264-323.

[181] R.E. Jensen, A dynamic programming algorithm for cluster analysis, *Operations Research,* 17(1969), 1034-1057.

[182] K.C. Kiwiel, *Methods of descent for nondifferentiable optimization, Lecture Notes in Mathematics,* 1133, Springer-Verlag, Berlin, 1985.

[183] W.L.G. Koontz, P.M. Narendra and K. Fukunaga, A branch and bound clustering algorithm, *IEEE Transactions on Computers*, 24(1975), 908-915.

[184] A. Likas, M. Vlassis and J. Verbeek, The global k-means clustering algorithm, *Pattern Recognition*, 36, (2003) 451-461.

[185] O.L. Mangasarian, Mathematical programming in data mining, *Data Mining and Knowledge Discovery,* 1(1997) 183-201.

[186] N. Mladenovic and P. Hansen, Variable neighborhood search, *Computer and Operations Research,* 24(1997) 1097-1100.

[187] P.M. Murphy and D.W. Aha, UCI repository of machine learning databases, Technical report, Department of Information and Computer Science, University of California, Irvine, 1992, www.ics.uci.edu/mlearn/MLRepository.html.

[188] C.R. Reeves (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell, London, 1993.

[189] G. Reinelt, TSP-LIB-A Traveling Salesman Library, *ORSA J. Comput.* 3(1991), 319-350.

[190] Rubinov, A.M., *Abstract Convexity and Global Optimization,* Kluwer Academic Publishers, Dordrecht, 2000.

[191] Rubinov, A.M. and Andramonov, M., Lipschitz programming via increasing convex-along-rays functions, *Optimization Methods and Software,* Vol. 10, 1999, pp. 763-781.

[192] S.Z. Selim and M.A. Ismail, k-means-type algorithm: generalized convergence theorem and characterization of local optimality, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1984), 81-87.

[193] S.Z. Selim and K.S. Al-Sultan, A simulated annealing algorithm for the clustering, *Pattern Recognition*, 24(10) (1991), 1003-1008.

[194] H. Spath, *Cluster Analysis Algorithms*, Ellis Horwood Limited, Chichester, 1980.

[195] L.X. Sun, Y.L. Xie, X.H. Song, J.H. Wang and R.Q. Yu, Cluster analysis by simulated annealing, *Computers and Chemistry*, 18(1994), 103-108.

[196] Leonard kleinrock, Farouk Kamoun, *Optimal clustering structures for hierarchical topological design of large computer networks*, Networks, Vol. 10(1980) 221-228

[197] Srinivas Vegesna, *IP quality of Service*, Cisco Press, 2001.

[198] A.M. Bagirov and A.M. Rubinov, *Global minimization of increasing positively homogeneous functions over the unit simplex,* Annals of Operation Research 98 (2000), 171-188.

[199] $http://www.pcguide.com/care/pm-c.html$

[200] $http://www.weibull.com/SystemRelWeb/preventive\_maintenance.htm$

[201] Joseph B. Keller, *Optimum inspection policies*, Management Science Vol. 28. No. 4, April 1982.

[202] M. N. Gopalan and N. N. Murulidhar, *Cost analysis of a one-unit repairable system subject to on-line preventive maintenance and/or repair*, Microelectron. Reliab. , Vol. 31 pp. 223-238, 1991.

[203] Xiaodong Yao, Michael Fu, Steven I. Marcus, *Optimization of preventive maintenance scheduling for semiconductor manufacturing systems: models and implementation*, Proceedings of the 2001 IEEE International conference on Control applications, September, Mexico City, 2001.

[204] Xiaodong Yao,*OPTIMAL PREVENTIVE MAINTENANCE POLICIES FOR UNRELIABLE QUEUEING AND PRODUCTION SYSTEMS*, PhD 2003-4. $http://techreports.isr.umd.edu/reports/2003/PhD_2003-4.pdf$

[205] Farhas Azadivar, J. Victor Shu,*Use of simulation in optimization of maintenance policies*, Proceddings of the 1998 Winter Simulation Conference, http://www.informs-cs.org/wsc98papers/145.PDF