Title  Cost Effective UMTS Transport Topology Optimization
Name  Emanuel Thomas

This is a digitised version of a dissertation submitted to the University of Bedfordshire.

It is available to view only.

This item is subject to copyright.
COST EFFECTIVE UMTS TRANSPORT TOPOLOGY OPTIMIZATION

E.THOMAS

ABSTRACT

This thesis studies the logical topology design for UMTS Radio Access Networks. The topology planning is investigated with specific focus on the transport network links of the Iub interface, which connects the Node B with the RNC. These links are considered as one of the most important economic factors for the UMTS network dimensioning. In order to satisfy subscriber demand and to cope with high bit-rate multimedia services, a large number of base stations are required in future cellular radio access networks. The radio access network thus becomes more complex and the transport resources for the Iub interface become considerably costly. The ultimate goal of this thesis is to investigate important aspects related to the topology planning of these interconnection links and to give insight into suitable choices for the network design. The formulated task proves a combinatorial optimization problem, which seems to be NP-hard. Therefore, in order to provide a comprehensive investigation on the topology design of the access network various heuristic techniques are applied and studied. In the framework of this thesis, the 3G UMTS network based on WCDMA is considered as the basic UMTS network under investigation.
DEDICATION

To my sisters Gloria and Louise
ACKNOWLEDGEMENT

I would like to thank my Director of Studies, Dr Paul Sant, for his guidance and patient support of my research work.

I also wish to thank my friends Ebenezer and Agu for their loyal friendship and moral support through difficult times.
DECLARATION

I declare that this thesis is my own unaided work. It is being submitted for the degree of MPhil at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other University.

Name of candidate: Emanuel Thomas

Signature:

Date:

31st August 2010
# TABLE OF CONTENTS

Abstract ........................................................................................................................................ii
Dedication ..................................................................................................................................iii
Acknowledgement ........................................................................................................................iv
Declaration..................................................................................................................................v
Table of Contents ........................................................................................................................vi
List of Acronyms ..........................................................................................................................ix
List of Figures ................................................................................................................................xii
List of Tables ..................................................................................................................................xiii

**Chapter 1: INTRODUCTION** ..............................................................................................1

1.1 Introduction ......................................................................................................................... 1

1.2 Research Methodology ...................................................................................................... 3

1.3 Outline of the Thesis .......................................................................................................... 5

1.4 Scope and Objectives of the Research .............................................................................. 7

1.4.1 Complexity of the Optimisation Problem ................................................................. 9

**CHAPTER 2: OVERVIEW OF UMTS ARCHITECTURE** .................................................11

2.1 Introduction ......................................................................................................................... 11

2.2 Overview of the UMTS Architecture .............................................................................. 11

2.3 UMTS Terrestrial Radio Access Network (UTRAN) ...................................................... 16

2.3.1 Radio Network Controller (RNC) ............................................................................. 20

2.3.2 Node B ....................................................................................................................... 22

2.4 Transmission (Backhaul) in the UTRAN .......................................................................... 23

2.5 Evolution of the UMTS system towards LTE ................................................................. 28

**CHAPTER 3: UTRAN PLANNING AND OPTIMIZATION METHODS** .......................33

3.1 The planning problem of UMTS Networks ..................................................................... 33

3.2 The Modular Approach ..................................................................................................... 34

3.2.1 Cell Planning Subproblem ......................................................................................... 35

3.2.2 Core Network Planning ........................................................................................... 37
CHAPTER 4: UTRAN TOPOLOGY PLANNING PROBLEM ........................................ 67

4.1 Introduction ................................................................................................. 67

4.2 Objectives of UTRAN Topology Planning and Optimization ...................... 67

4.3 The Capacitated p-median Problem ............................................................ 69

4.4 Formulation of the UTRAN Planning Problem ............................................ 71

4.4.1 Cost Function Calculation in the RNC location problem: the Terminal Assignment ........................................................................................................ 73

4.4.2 Modelling RNC Location Problem as Capacitated p Median .......... 75

4.5 Solution Approaches .................................................................................... 77

4.5.1 Cellular Genetic Algorithms (cGA) .................................................. 78

4.5.2 Steady State Genetic algorithm (ssGA) ........................................... 82

4.5.3 Local Search using Simulated Annealing ....................................... 83

4.6 Implementation of the Algorithms ............................................................. 84

4.6.1 cGA-based Terminal Assignment .................................................. 85

4.6.2 cGA-based RNC location problem................................................. 88

CHAPTER 5: EXPERIMENTAL SETUP AND RESULTS................................. 89

5.1 Introduction ................................................................................................. 89

5.2 Network Configuration ................................................................................ 89

5.3 Algorithm configuration............................................................................... 91

5.4 Application Examples ................................................................................... 94
5.4.1 Experiments for the Terminal Assignment ........................................ 94
5.4.2 RNC Location using combined CGA and SA................................. 99
5.5 Choice of optimization method .......................................................... 103

CHAPTER 6: CONCLUSION AND FURTHER WORK..................................... 106

6.1 Conclusion .......................................................................................... 106
6.2 Contributions ...................................................................................... 108
6.3 Further work .................................................................................... 109

REFERENCES ............................................................................................. 112
**LIST OF ACRONYMS**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1G</td>
<td>First Generation</td>
</tr>
<tr>
<td>2G</td>
<td>Second Generation</td>
</tr>
<tr>
<td>2-D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>3G</td>
<td>Third Generation</td>
</tr>
<tr>
<td>4G</td>
<td>Fourth Generation</td>
</tr>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>AAL2</td>
<td>ATM Adaptation Layer type 2</td>
</tr>
<tr>
<td>AAL5</td>
<td>ATM Adaptation Layer type 5</td>
</tr>
<tr>
<td>ALCAP</td>
<td>Access Link Control Application Part</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>BSS</td>
<td>Base Station Subsystem</td>
</tr>
<tr>
<td>BSC</td>
<td>Base Station Controller</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>cGA</td>
<td>Cellular Genetic Algorithm</td>
</tr>
<tr>
<td>CN</td>
<td>Core Network</td>
</tr>
<tr>
<td>CRNC</td>
<td>Controlling RNC</td>
</tr>
<tr>
<td>CS</td>
<td>Circuit Switched</td>
</tr>
<tr>
<td>DRNC</td>
<td>Drift RNC</td>
</tr>
<tr>
<td>EA</td>
<td>Evolutionary Algorithms</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data rates for GSM Evolution</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>EPS</td>
<td>Evolved Packet System</td>
</tr>
<tr>
<td>GGSN</td>
<td>Gateway GPRS Support Node</td>
</tr>
<tr>
<td>GMSC</td>
<td>Gateway MSC</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>HSDPA</td>
<td>High Speed Downlink Packet Access</td>
</tr>
<tr>
<td>HSPA</td>
<td>High Speed Packet Access</td>
</tr>
<tr>
<td>HSS</td>
<td>Home Subscriber Server</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Sub-system</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunications Union</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>ME</td>
<td>Mobile Equipment</td>
</tr>
<tr>
<td>MGW</td>
<td>Media Gateway</td>
</tr>
<tr>
<td>MS</td>
<td>Mobile Station</td>
</tr>
<tr>
<td>NBAP</td>
<td>Node B Application Part</td>
</tr>
<tr>
<td>PLMN</td>
<td>Public Land Mobile Network</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>PS</td>
<td>Packet Switched</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAB</td>
<td>Radio Access Bearer</td>
</tr>
<tr>
<td>RAN</td>
<td>Radio Access Network</td>
</tr>
<tr>
<td>RANAP</td>
<td>Radio Access Network Application Part</td>
</tr>
<tr>
<td>RNC</td>
<td>Radio Network Controller</td>
</tr>
<tr>
<td>RNS</td>
<td>Radio Network Sub-system</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>SAE</td>
<td>Service Architecture Evolution</td>
</tr>
<tr>
<td>SGSN</td>
<td>Serving GPRS Support Node</td>
</tr>
<tr>
<td>SHO</td>
<td>Soft Handover</td>
</tr>
<tr>
<td>SRNC</td>
<td>Serving RNC</td>
</tr>
<tr>
<td>SRNS</td>
<td>Serving RNS</td>
</tr>
<tr>
<td>ssGA</td>
<td>Steady State Genetic Algorithm</td>
</tr>
<tr>
<td>SS</td>
<td>Scatter Search</td>
</tr>
<tr>
<td>TA</td>
<td>Terminal Assignment</td>
</tr>
<tr>
<td>TS</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunication Services</td>
</tr>
<tr>
<td>USIM</td>
<td>UMTS Subscriber Identity Module</td>
</tr>
<tr>
<td>UTRAN</td>
<td>UMTS Terrestrial Radio Access Network</td>
</tr>
<tr>
<td>WCDMA</td>
<td>Wideband Code Division Multiple Access</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 2.1: Logical view of UMTS architecture ..................................................... 12
Figure 2.2: Network elements in a PLMN ................................................................. 13
Figure 2.3: Logical architecture of the UTRAN ..................................................... 16
Figure 2.4: Physical UTRAN architecture (a) pre Release 5 (b) Release 5 ........ 18
Figure 2.5: Handover in UMTS network ................................................................. 19
Figure 2.6: Traditional approach to transport network planning ....................... 26
Figure 2.7: An example of a UMTS backhaul network ........................................ 27
Figure 2.8: E-UTRAN architecture ...................................................................... 31
Figure 3.1: UMTS Planning Subproblems ............................................................... 34
Figure 3.2: UTRAN Topology ............................................................................... 40
Figure 3.3: Example of the clustering method ....................................................... 44
Figure 3.4: Original network .................................................................................. 46
Figure 3.5: (a) the initial spanning tree (b) final optimized network .............. 46
Figure 3.6: Flowchart of a simple GA ................................................................... 64
Figure 4.1: Structure of a cellular GA ................................................................. 78
Figure 4.2: Chromosome representing an assignment of Node B-RNC ............ 85
Figure 4.3: Fitness function flow diagram ............................................................. 87
Figure 5.1: Network configuration ....................................................................... 90
Figure 5.2: TA test problem instance, Node B=100, RNC=5 ......................... 96
Figure 5.3: CPU time/s in the TA simulations ...................................................... 97
Figure 5.4: Infeasible solution to TA using cGA ............................................. 98
Figure 5.5: Feasible solution to TA using cGA ................................................. 98
Figure 5.6: TA solution using simulated annealing ........................................... 99
Figure 5.7: Optimized solution for RNC location in instance #2 ............... 102
LIST OF TABLES

Table 2.1     Evolution of the UMTS specifications ................................................... 28
Table 3.1     Example of network specification .......................................................... 45
Table 3.2:    Prim-Dijkstra algorithm........................................................................ 50
Table 3.3:    Analogy between physical system and the optimization problem .... 60
Table 3.4:    Template of Simulated Annealing Algorithm............................................ 61
Table 3.5:    Template of a population-based heuristic ............................................. 62
Table 4.1:    Pseudocode for cellular GA................................................................. 79
Table 4.2:    Crossover operator in binary string chromosome ......................... 80
Table 4.3:    Example of neighborhood patterns ................................................. 82
Table 4.4:    Pseudocode for ssGA ..................................................................... 82
Table 4.5:    Pseudocode for SA.......................................................................... 84
Table 5.1     Demands and capacities for one instance of TA ........................ 95
Table 5.2:    Main characteristics of RNC location instances ......................... 100
Chapter 1: INTRODUCTION

1.1 Introduction

Mobile Cellular networks are experiencing significant changes which have a direct effect on the planning of the networks. The total voice traffic has been growing slowly in the last couple of years while there has been an explosion in data traffic in the network. There continues to be enormous growth in subscriber numbers, and their ever increasing demand for high speed, high bit-rate data services (mobile internet access, video streaming, and web-based applications). As a result, cellular networks need to adapt for operators to stay competitive.

There has been an evolution in wireless communications almost every ten years. The first generation (1G) in the 1980s and the second generation (2G) mobile systems in the 1990s were designed to support Circuit Switched services to users. Data rates for 2G subscribers initially averaged up to 14 Kb/s, this was increased with the phased enhancement of the 2G GSM (Global System for Mobile Communications) standard on an annual basis. 2G enhancement services such as Enhanced Data Rates for GSM Evolution (EDGE), and General Packet Radio Service (GPPRS), offered users theoretical data rates up to 384 Kb/s in uplink and in downlink and simultaneous voice and data [1]. Universal Mobile Telecommunications System (UMTS) is the third generation (3G)
successor to 2G systems, specified by the European Telecommunications Standards Institute (ETSI) and Third generation Partnership Project (3GPP) within the framework defined by the International Telecommunication Union (ITU). WCDMA (Wideband Code Division Multiple Access) and its evolution HSPA (High Speed Packet Access) have emerged as the main 3G air interface globally. Since the launch of the first UMTS networks in 2002 there has been an intensive growth in subscriber population and in the number of operational networks around the world. Currently the number of WCDMA/HSPA subscribers has exceeded 500 million and there are over 300 commercial HSPA networks globally supporting peak data rates up to 42 Mbps [2].

The early WCDMA deployments paved the way for the introduction of mobile broadband. The introduction of commercial HSPA upgrade to existing WCDMA networks in 2005 meant that peak bit rates increased from 0.384 Mbps initially to 1.8–3.6 Mbps and later to 7.2–14.4 Mbps [2]. These speeds were comparable to low end fixed Asymmetric Digital Subscriber Line (ADSL) and had the effect of pushing the cost per bit down, eventually leading to a flat rate pricing model for HSPA mobile broadband. The combination of the flat rate pricing together with higher data rates allowed users to consume large volumes of data, leading to an explosion in the total data volume in HSPA networks. As a result, cellular networks have in recent years shifted from being voice-centric to being data-dominated.
This type of fast data growth has led to many challenges in terms of cost efficiency, not least because increases in data traffic do not lead to extra revenue due to the flat rate pricing.

Future cellular networks beyond existing 3G systems are expected to be high capacity, multi-service networks that will play an important role in future telecommunications [3]. These systems are expected to provide even higher transmission rates and larger capacity than existing 3G systems, both in terms of number of subscribers and volume of traffic carried. As a result, the transport network of future cellular systems will be a very complex, costly, and high capacity system[4]. The emergence of these new technologies and traffic types raises many network planning and optimization problems that need to be solved.

1.2 Research Methodology

One of the most important problems of UMTS network planning it is required to solve the topological planning of RAN physical links configuration. Topological planning to a large extent determines the investment cost of the network and can have a considered effect on the long-term network performance. As a consequence, a near-optimal topological design is of critical importance. Due to the high complexity and costs of future RAN architectures, it is appropriate to use algorithmic optimisation methods. In the context of our research we focus on the topological design of the UMTS Terrestrial Radio Access Network (UTRAN) and investigate solutions to problems associated with mobile
backhaul costs. We approach the UTRAN planning problem by applying some basic principles in network design. Network planning generally comprises the following steps in sequence [5]:

- **Design issues determination:** this is to clarify aspects of the design tasks in the initial stages. For example, whether the network is built on top of existing capabilities; which aspect should be the most important: network cost, its performance or manageability; what is the focused problem: node placement and sizing or link topology optimization etc. In this case we assume that a minimal network cost is the objective and the focused problem is a combination of node placement and link topology.

- **Input data collection:** traffic demands, QoS requirements, and device characteristics. Almost all of the input data in our experiments is hypothetical and based on a number of simplifications.

- **Design task formalisation:** this is a technical step where the aim is to mathematically combine all the information retrieved in the previous steps in a consistent and systematic way (i.e. the formulation of the planning problem). This is carried in chapter four.

- **Design tool choice:** according to the features of the formulated task, a method is developed for the design process. I have chosen heuristic methods in the form of genetic algorithms and simulated
annealing, firstly because of their wide applicability, and secondly because of the size and complexity of the problem.

- Presentation of design results and assessment of the proposed design method.

The topology planning part of the thesis strictly applies to the aforementioned design concept. The idea of building an UMTS access network with optimal topology, on top of the existing UTRAN infrastructure for given traffic demands and capacity requirements, represents the principle of the first two steps. The formulated task proves an integer programming problem, which seems to be NP-hard[6]. Therefore, in the fourth step heuristic algorithms are utilized.

1.3 Outline of the Thesis

Chapter one introduces the basic concepts of the research and the motivation for the study. The scope and the limitations of the study are also discussed while the aims and objectives are highlighted.

Chapter two gives an overview of the UMTS architecture and discusses the major entities which make up its hierarchical structure. The UTRAN which is the focus of our research is discussed in some detail from both logical and physical viewpoints. The main hardware elements of the UTRAN are introduced and their functions outlined.

Chapter three starts with the background information on the planning problem of UMTS networks and discusses key aspects of its complexity. It
identifies the UTRAN topology planning as an important aspect of the overall network planning. Different approaches to solving the UTRAN topology design are explored both within the context of the overall UMTS planning problem and the UTRAN problem in isolation. The literature review of related works is also discussed. The literature review discusses the modern metaheuristic methods that are appropriate for such complex optimization problems. Finally we cover the theoretical analysis relevant to the subject. These include the description of optimization methods for access network planning, and a closer look at Genetic Algorithms (GA) and Simulated Annealing (SA).

Chapter four describes the methodology and the framework of the system. The formulation of the UTRAN topology planning problem is discussed. A detailed approach to solving the problem is outlined. The use of genetic algorithms and simulated annealing approach to solving the design problem is described, along with implementation details.

Chapter five illustrates details of the experiments performed. These include the graphical representation of the network layout and the optimized topology solutions obtained. The results are discussed and some conclusions are drawn.

Chapter six is the concluding chapter of the thesis; it covers the analysis and evaluation of the experimental results obtained in chapter five. It
highlights the contributions made. The aims and objectives of the work are reflected upon and the major contributions from the work are evaluated.

1.4 Scope and Objectives of the Research

In UMTS network planning it is required to solve the topological planning of both the core and access networks. Due to the complexity of the design task it is usually divided into subproblems. This research focuses on one aspect of the complex network design problem, the planning and optimisation of the access part of the UMTS network. The UMTS access network design problem consists of finding the number and location of the Radio Network Controllers (RNC) and the interconnection topology between the RNC and the Node B (base stations) which they serve.

Network planners face many difficult decisions when designing a network that must meet the communication requirements of users with a high probability. These decisions are interdependent and influenced by a variety of parameters including, hardware cost, communication requirements, budget limitations and integration with existing network. One very important feature of the evolution of mobile communications concerns changes in the future wireless systems. It is expected that an enhanced CN of 3G systems can provide the functional capabilities needed to support future services; as a result, the CN will remain largely unchanged, with the main changes taking place in the Radio Access Network (RAN), in particular on a physical transmission layer. It is for this reason that our
main focus is on the topology design of the UMTS radio access network (UTRAN).

It is expected that future 3G systems will be designed to offer bit rates of 100 Mb/s (in mobile environment) to 1 Gb/s (fixed indoors) [3]. In order to achieve such high rates a larger frequency than the 5 MHz currently used by 3G systems needs to be considered along with transmission systems more suited for high-rate transmission. However, the use of higher frequency bands to achieve higher transmission generally results in strong signal degradation on non line-of-sight paths and reduces the area of a cell that a base station can cover [7]. Doubling the frequency almost halves the non line-of-sight range and so could require four times the number of cells to achieve the same coverage [8].

To maintain the coverage area in future cellular access networks more base stations will have to be used resulting in smaller cell sizes and a larger number of base stations in the RAN [4, 6]. This has the undesirable effect of increasing the network cost. The expected increase in number of base stations also implies more frequent handover in the system. In addition, the significantly higher bit rates will impose a heavier load on interconnecting links between Node Bs and RNC, suggesting changes to the RAN architecture[3, 6]. For this reason, physical RAN topologies will be explored along with a thorough analysis of traditional RAN topologies currently used in 3G UMTS systems[9, 10]. It is important also to analyse
RAN physical links topologies with respect to both reliability and cost, compare the different topologies and make recommendations on applying the different ones in future UTRAN designs.

The transport network of the RAN is key to successful operations of 3G networks as it provides reliable connectivity between and within the core and access domains. Currently, most RAN topologies are based on point-to-point leased lines between base station and RNC, and are not ideally suited for future IP-based multimedia traffic. Industry consensus estimates that twenty five percent of the total mobile network cost is Transport, of which seventy five percent is the transport flows from the base stations to the mobile service provider’s core network. Since mobile operators will have to deal with higher volumes of traffic with higher transmission rates in future 3G systems, there is an urgent need to adapt their transport network to lower transport costs. It is important therefore to investigate the optimal configuration of physical links between RNC and Node B in future RAN architectures to evaluate the impact of proposed changes on RAN deployment costs.

1.4.1 Complexity of the Optimisation Problem
If we consider the configuration of a small geographical area with say 100 Node Bs and 5 RNCs, the resulting number of possible network configurations is beyond imagination. With this network arrangement the optimization has to consider $5^{100}$ possible configurations. This phenomenon is called combinatorial explosion[11], where the number of
solutions to a combinatorial optimisation problem grows extremely quickly with the problem size. Given that a typical UTRAN can contain several hundreds of Node Bs, it is clear that the problem of planning the access links between Node B and RNC is computationally hard.

Finding a feasible solution to such a problem can usually be done quickly, but finding an optimal one is typically hard. There is a formalised concept for measuring ‘hardness’ of optimisation problems in complexity theory, see the classical text [12]. A central notion in complexity theory is \textit{NP-hardness}. The planning problem discussed in this thesis is NP-hard. It is commonly believed that ‘hard’ optimisation problems cannot be solved to proven optimality ‘quickly’. Due to the complexity of these problems classical optimization methods are not appropriate and so modern heuristic approaches are used. The next chapter takes a closer look at the UMTS system under investigation and gives an overview of both the current system architecture and also the proposed changes for the future.
CHAPTER 2: OVERVIEW OF UMTS ARCHITECTURE

2.1 Introduction

In this chapter, a wide overview of the UMTS system architecture, including the logical network elements and interfaces, is presented. The UMTS system mainly utilises the same well-known architecture that has been used by all main second generation systems, but there are some significant differences introduced as well. The UMTS system consists of a number of logical network elements, which can be grouped based on similar functionality or based on which sub-network they belong to. In this context, the first one of these approaches is preferred: each network element has its own logical functionality and the interfaces between the elements are well defined.

2.2 Overview of the UMTS Architecture

As can be seen from figure 2.1, a UMTS network consists of three interacting domains, the UTRAN, the CN, and the User Equipment (UE). Figure 2.1 shows a high-level system architecture with external reference points and the internal interfaces to the UTRAN. From a specification and standardization point of view, both UE and UTRAN consist of new protocols whose designs are based on the needs of the WCDMA radio technology. By contrast, the definition of the CN is adopted from GSM.
Another way of grouping UMTS network elements is to divide them into sub-networks. The UMTS is sometimes described as modular because it is possible to have several network elements of the same type. This possibility allows the division of the UMTS into sub-networks that are operational either on their own or together with other sub-networks, distinguished from each other with unique identities. Such a sub-network is called a UMTS Public Land Mobile Network (PLMN).

Typically, one PLMN is operated by a single operator, and is connected to other PLMNs as well as to other types of networks such as the Public Switched Telephone Network (PSTN), Integrated services Digital Network (ISDN), and the internet. Figure 2.2 (figure from [2]) shows the elements in a PLMN and their connections to external networks. A short introduction to all the elements is given here, while a detailed description of the UTRAN is given later in the chapter.
A UE consists of the Mobile Equipment (ME), and the UMTS Subscriber Identity Module (USIM). The ME is the radio terminal used for communication over the Uu interface commonly known as the air interface. The USIM holds the subscriber identity performs and stores authentication information and subscription related information needed in a terminal. The UTRAN also consists of two main elements, namely the Node B and the RNC. The UTRAN provides the air interface access method for User Equipment, handling all radio-related functionalities such as access control, radio resource management, mobility management, broadcast and multicast services. The main objective of the UTRAN is to make the link between mobile users and the core network.

The CN consists of physical entities that provide support for the network features and telecommunication services. The main function of the CN is to provide switching, routing and transit for user traffic to external networks such as the Public Switched Telephone (PSTN) and the internet. In addition, it also contains the databases and network management functions. The CN is divided in circuit switched (CS) and packet switched...
(PS) domains. Some of the CS elements include the Mobile Services Switching Center (MSC), Visitor Location Register (VLR), and Gateway MSC (GMSC). The MSC performs all necessary functions to handle circuit switched services to and from the mobile stations. Its major jobs are authentication, location management, handovers, registration and the routing of calls to roaming mobile stations. The Gateway MSC (GMSC) is an MSC located between an external CS network such as the PSTN and the other MSCs in the network. It functions as a switch routing incoming calls to the appropriate MSCs by first interrogating the appropriate HLR.

The VLR is a database that contains information about all roaming mobile subscribers that are currently visiting a particular MSC or SGSN service area. In an all IP based wireless network the Home Subscriber Server (HSS) acts as a central repository of all subscriber-specific authorizations and service profiles to support calls and/or data sessions. It is analogous to the Home Location Register (HLR) in mobile networks with the important differences that there is no VLR and the HSS supports all types of devices, not just wireless.

Packet switched elements include Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). The Serving GPRS (General Packet Radio Service) Support Node (SGSN) is the central element in the packet switched network (PS). It handles subscription and location information to and from UEs which are located within the SGSN service
The GGSN corresponds to the GMSC in the circuit switched (CS) network, however, unlike the GMSC that only routes incoming traffic, the GGSN also routes outgoing traffic. The GGSN interfaces to an external IP network and handles the UMTS session management and communication setup with the external network. The external networks can be divided into two groups:

- **CS networks** – these provide circuit-switched connections such as the existing telephony service. ISDN and PSTN are examples of CS networks.

- **PS networks** – these provide connections for packet data services. The internet is one such example of a PS network.

The approach of the UMTS standardization committees is such that the standards do not specify internal functionalities of network elements in detail, but rather the interfaces between elements is completely specified and standardized. Such an interface is said to be open and it becomes possible to connect equipment from different manufacturers to the interface [1]. All UMTS interfaces (not only the interfaces visible from outside the UTRAN but also internal interfaces) are completely standardized. This is therefore the case for the Cu, Uu, Iu, lub, and lur, the main open interfaces specified for UMTS.

The Cu is the electrical interface between the USIM smartcard and the ME, and follows the standard format for smartcards. The Uu is the WCDMA radio interface through which the UE accesses the fixed part of
the system. The Iu interface connects the UTRAN to the CN by linking the
RNC with either a MSC or a SGSN. The Iu-CS interface links the RNC to
the circuit switched CN via a MSC, while the Iu-PS interface links the RNC
to the packet switched CN via a SGSN.

2.3 UMTS Terrestrial Radio Access Network (UTRAN)
The current access network of UMTS is known as the UTRAN. The
UTRAN consists of one or more Radio Network Subsystems (RNS) as
shown in figure 2.3 (figure from [2]). An RNS includes one RNC and one or
more Node Bs. Each RNS controls the allocation and release of specific
radio resources to establish a connection between a UE and the UTRAN.
Each RNS has a physical tree topology, with the RNC at the root of each
tree. In this scenario Node Bs are connected either directly to the RNC or
via other Node Bs allowing traffic to be aggregated on the links
interconnecting them.

Figure 2.3: Logical architecture of the UTRAN

The main network elements and interfaces referred to in the 3GPP
specifications [14] include the terrestrial components of the 3G UMTS
system, referred to collectively as the UTRAN. The main elements of the UTRAN are:

- Node B or base station, providing gateway services between the handset/RF interface and the RNC, via the lub interface.
- RNC, responsible for the control of the radio resources in its domain UTRAN (the Node Bs connected to it).

The lub interface is the fully open interface that sets up connection between a Node B and a RNC. The RNC manages the Node B via the lub interface, performing functions such as Operation and Maintenance of the Node B, traffic management of dedicated, common, and shared channels, and timing and synchronization management, etc. Although this interface was initially designed to support the inter-RNC soft handover, more features were added during development of the standard and the lur interface now supports four distinct functions:

- Support of dedicated channel traffic
- Support of common channel traffic
- Support of global resource management
- Support of basic inter-RNC mobility[2]

Figure 2.4 (figure from [15]) shows the corresponding physical architecture for the logical UTRAN shown earlier in figure 2.3. The ellipses on the left represent the cells controlled by the Node B. The figure highlights the
changes in the UTRAN architecture between the different releases. The most important feature to observe is the tree-like structure.

One Node B is responsible for several cells, one RNC is responsible for several Node Bs, and one SGSN (or MSC) is responsible for several RNCs. This tree-like structure simplifies the control of the network; however, the hierarchical structure is inflexible making redesign of the physical network difficult. As a result, in release 5 the hierarchical relation between RNC and SGSN was abandoned as shown in figure 2.4 (b).

When a user’s communication switches from one Node B to another, the new Node B becomes responsible for relaying this communication through the allocation of a new radio channel for the user. Supporting the transfer
of the communication from one Node B to another is called a ‘handover’. This mechanism, which primarily involves the RNCs, occurs mainly when the level of signal received by the user reaches a certain threshold. There are two types of handovers, complex and soft. These are further illustrated in figure 2.5. (figure from [16])

In Figure 2.5, for example, a user’s movement from Node Bi to Node Bj (represented by the green arrow) is referred to as a soft handover because these two nodes are connected to the same RNC (RNC1). The RNC that supervises the two nodes remains the same, and the cost is low in terms of system resources. By contrast, a case where the user moves from Node Bi to Node Bk (represented by the red arrow) is considered a complex handover. The cost associated to this concept is significantly higher, as both RNCs 1 and 2 remain active during the handover procedure, and the database that contains subscribers' information requires an update. In
addressing the UTRAN topology design problem one of the main objectives is to limit the number of hard handovers which occur in the system. In practice this can be tackled by assigning Node Bs to RNCs based on their geographical relationship. In order to deal with the future increase in traffic and their largely varying traffic types within the UTRAN, it is necessary to extend/modify the existing topology to improve its flexibility.

This can be done by introducing additional links in the topology or changing the topology altogether. Also, the introduction of IP in the access network puts new demands on the system to provide the necessary level of QoS and dependability to the end users. The question of how to extend/modify the current topology to meet future requirements is a non-trivial dimensioning and optimization problem. In this work one solution to this problem is tackled by applying genetic algorithms (GA). Genetic algorithms are used because they allow a less restrictive problem formulation as compared to mathematical programming.

2.3.1 Radio Network Controller (RNC)

The RNC is the main element in the UTRAN that controls the use and the reliability of the radio resources. It manages the radio network resources, mobility and the user data transport using Radio Access Bearers between UEs and the CN. The intelligence of an RNC, e.g., radio resource management function, resides in the various feature controller cards. These feature controller cards process the Node B Application Part
(NBAP) and Radio Access Network Application Part (RANAP) signaling messages and make decisions on which voice calls/data sessions to admit. To allow for scalability, the feature controller card may itself consist of multiple boards with multiple processors. For redundancy purposes, two or more signaling cards and gateway cards are provided in each RNC chassis. In order to understand the behaviour and functionality of the RNC in different situations, three different RNC roles have been defined.

Concerning one connection between UTRAN and one UE, the following roles of RNCs exist:

- The Serving RNC (SRNC) that controls the connections to a UE.
- The Drift RNC (DRNC) that lends its resources for the serving RNC for a particular UE.

Each RNC also has the controlling RNC towards its Node Bs.

- The Controlling RNC (CRNC) is responsible for the load and congestion control of its own cells, and also executes admission control and code allocation for new radio links that are established in those cells[1].

When an UE to UTRAN connection uses resources from more than one RNS, the RNCs involved can have two different logical roles. The Serving RNC terminates the Iub link for the transport of user data and the corresponding Radio Access Network Application Part (RANAP) signalling to and from the core network. The SRNC also terminates the Radio Resource Control (RRC) signalling on Layer 3 between the UE and
UTRAN. The Drift RNC is any RNC other than the SRNC that controls cells used by the UE. It routes the data transparently to the SRNC between the lub and the lur interface, except when the UE is using a common or shared transport channel.

Normally, an RNC is designed to handle a certain capacity, e.g., $x$ Erlangs of voice traffic (with 12.2 Kb/s per channel). Each RNC can support a certain maximum number of cells and Node Bs (typically we assume 3 cells/Node B) [17]. It has a certain number of Synchronous Transport Module level-1 (STM-1) interfaces, and a certain maximum raw throughput over the lu, lur, lub interfaces. Typically, the maximum raw lub throughput is 150 percent of the maximum raw lu throughput while the maximum raw lur throughput is designed to be about one third of the maximum lu throughput [17].

2.3.2 Node B
The Node B is the element within the UTRAN that provides the physical radio link between the UE and the network. The Node B performs the air interface physical layer (L1) processing such as channel coding and interleaving, rate adaptation, and spreading. The Node B also performs some basic radio resource management operation such as the inner loop power control, mapping logical resources onto hardware resources, and transmitting system information message according to scheduling parameters given by the RNC.
Along with the transmission and reception of data across the radio interface the Node B also applies the codes that are necessary to describe channels in a CDMA system. A Node B typically consists of various line cards that interface to the RNC, a main controller that processes the radio signaling messages, transport signaling messages, and performs resource management, and various radio cards that house channel elements/radios that allow the Node B to send/receive radio signals to/from the mobile terminals[17]. The Node B in the UMTS system is analogous to the base transceiver station (BTS) in the older GSM system.

2.4 Transmission (Backhaul) in the UTRAN

In a cellular network the traffic from cell sites to the RNCs and further to the MSCs and Data Gateways and finally to the outside world is carried by the transport network or transmission network. The main purpose of the transmission network is to provide reliable aggregation and delivery infrastructure for the transport of any type of traffic across the cellular network. In the UMTS system the UTRAN segment of the transport network is referred to as the backhaul network [18], the very network that will eventually carry the increasing broadband traffic.

More specifically, cellular backhaul refers to the transport of Node B flows to one or more aggregation sites which are then connected to the RNC. Tens or hundreds of Node Bs are connected to the RNC via the backhaul network. Cellular backhaul traffic is carried over lines of up to 100’s of km long, across multiple hops of, copper, microwave, satellite, or fiber
infrastructure. Backhaul traffic is compressed and so overall traffic in the backhaul network has been relatively low. As mobile broadband technologies start to penetrate the market, the usage of new technologies such as mobile TV and video calling will rise, increasing the amount of traffic originating and terminating in the UTRAN. This may have the unwanted effect of creating bottlenecks in the rest of the transport network.

To implement and operate backhaul networks, operators may either lease capacity from fixed line operators or own it. For owned capacity for backhaul infrastructure, wireless point-to-point microwave links are the most commonly used technology. Microwave is relatively inexpensive for low to medium capacity links, and point-to-point microwave systems offer simple and cost efficient backhauling for voice and high-speed data services. This is mainly because point-to-point microwave supports higher data rates than copper T1/E1 lines, and overcomes the high cost and limited availability associated with fiber. It is difficult however to scale up point-to-point microwave systems in support of the increased capacity demands of newer cellular technologies.

With the increasing popularity of devices such as iPhones, cellular providers are struggling to handle traffic congestion and the escalating bandwidth costs caused by the explosion of texting, Web browsing, and applications on iPhones and other smartphones. As a result, cellular
operators are now turning to fiber infrastructures and Ethernet is fast becoming the transport technology of choice for cellular backhaul.

The logical links in the UTRAN (Iub, Iur) do not suffice to establish communication. There must exist physical transmission links with sufficient capacity, and physical transmission paths to connect the end nodes of a logical link. The end nodes of the Iub link for instance are the Node B and the RNC. While the logical links are defined in the 3GPP standards, there is nothing in the standards that defines how the physical transmission links and paths should be implemented. Most existing backhaul infrastructures have been implemented based on point-to-point microwave radios and E1/T1 leased lines. T1 leased lines are heavily used in North America while microwave radios and E1 are used in the rest of the world.

A variety of transport systems are available for the backhaul infrastructure [18]:

- Microwave Radios – the microwave links are both point-to-point and point-to-multipoint and typically operate in frequencies below 40 GHz.
- Gigabit Ethernet Radio
- Local Multipoint Distribution System (LDMS) – LMDS is a fixed point-to-multipoint wireless service typically using spectrum in the 26 – 31.3 GHz and 40.5 – 42.5 range depending on country of use.
- Wimax systems
Wireless Gateways

As always, for any technology chosen there is a trade-off between various needs, requirements and parameters. Backhaul network design typically involves a trade-off between network reliability and speed of deployment and price. Traditional approaches to the backhaul network planning problem usually involved three steps as shown in figure 2.6:

- Select topology
- Choose a routing for communication demands using selected topology
- Choose capacity to accommodate chosen routings

![Figure 2.6: Traditional approach to transport network planning](image)

If the network planner chooses the least cost network topology, the result would always be a Minimum Spanning Tree (MST)[19]. This decision as the flowchart indicates determines the other planning steps. If for example
the topology chosen is an MST, then there is only a single path from each
node to the root. This implies that the routing is fixed and capacities on
any given path are severely limited. An example of a UMTS radio
infrastructure with radio relay links is shown in figure 2.7 (figure from [11]).
Traffic from the Node Bs to the RNC is aggregated on the interconnecting
links with branching stations allowing to comprise individual low capacity
E1 links into higher and multiple capacity links. Links closer to the RNC
are high capacity STM-1 (311 Mb/s) as they obviously have to carry the
highest volume of traffic.

Figure 2.7: An example of a UMTS backhaul network

The base station trunk is the entire physical transmission link between two
Node Bs or between a Node B and an RNC. In the case of a link failure
trunk rerouting switches all traffic from the main trunk to the backup trunk.
Rerouting need not be applied to all sites but only to the critical hub sites
where traffic aggregates from other sites. In the case of the UTRAN
planning problem, the cost and efficiency of the underlying physical
transmission network connecting UTRAN elements is dependent on an optimized logical topology at a higher level.

2.5 Evolution of the UMTS system towards LTE

The UMTS system continues to evolve and has been developed in stages and annual releases. The UMTS specifications are being written by 3GPP which is a partnership of standards development organizations. Each release of the 3GPP specifications represents a defined set of features; some new features are added while some old ones may be abandoned. Table 2.1 (adapted from [20]) summarizes the evolution of the 3GPP UMTS specifications towards Long Term Evolution (LTE).

<table>
<thead>
<tr>
<th>Release</th>
<th>Functional freeze</th>
<th>Main UMTS features of release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel-99</td>
<td>March 2000</td>
<td>Basic 3.84 Mcps W-CDMA (FDD &amp; TDD)</td>
</tr>
<tr>
<td>Rel-4</td>
<td>March 2001</td>
<td>1.28 Mcps TDD (TD-SCDMA)</td>
</tr>
<tr>
<td>Rel-5</td>
<td>June 2002</td>
<td>HSDPA</td>
</tr>
<tr>
<td>Rel-6</td>
<td>March 2005</td>
<td>HSUPA (E-DCH)</td>
</tr>
<tr>
<td>Rel-7</td>
<td>Dec 2007</td>
<td>HSPA+ (64QAM downlink, MIMO, 16QAM uplink)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LTE and SAE feasibility study</td>
</tr>
<tr>
<td>Rel-8</td>
<td>Dec 2008</td>
<td>LTE - OFDMA/SC-FDMA air interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAE - new IP core network</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Further HSPA improvements</td>
</tr>
</tbody>
</table>

The date given for the functional freeze refers to the dates when no new further items can be added to the release. There is usually a delay
between the functional freeze and the actual commercial launch of the release. After the first UMTS release (Rel-99), 3GPP stopped naming releases after the year and opted for a new scheme starting with release 4. The brief description of the releases provided here only focuses on their main features and is by no means complete. A comprehensive overview can be found at the 3GPP website [21].

The first UMTS networks based on release 99 were deployed in 2003 and offered typical bandwidths of 384 Kb/s. Release 4 contained mostly minor updates but introduced the 1.28 Mcps (millions of chips per second) narrow band version of WCDMA also known as Time Domain Synchronous Code Division Multiple Access (TD-SCDMA). The significant enhancements as compared to release 99 were the division of the GMSC into a transport node, the Media Gateway, and a control node (the GMSC Server). Following this was release 5, a major release published in 2002, in which High Speed Downlink Packet Access (HSDPA) introduced packet-based data services to UMTS. It enhanced the UMTS air interface to introduce a downlink shared channel with packet rates up to a theoretical value of 14.4 Mb/s. An IP Multimedia System (IMS) was also introduced, which meant that UMTS now had a standardized means of offering the multimedia services for which it was built [15].

Release 6 was also a major release in which the completion of packet data for UMTS was achieved with the addition of High Speed Uplink Packet
Access (HSUPA). HSUPA extends the downlink HSDPA from release 5 by adding a dedicated uplink channel with theoretical bandwidth up to 5.76 Mb/s. Release 7 continued the enhancement of the HSPA technology with the introduction of HSPA+. Improvements were made to HSPA such as downlink Multiple Input-Multiple Output (MIMO), and higher bandwidths of 28 Mb/s on the downlink and 11 Mb/s on the uplink. Release 7 also contained the first work on Long Term Evolution/Service Architecture Evolution (LTE/SAE) with the completion of feasibility studies.

Release 8 differs substantially from previous releases, the result of which is called the Evolved Packet System (EPS). This is the work of the LTE project whose main goals were a higher data rate and lower packet delay in the UMTS system. Compared to earlier releases, the radio interface is completely new as is the architecture of the RAN. These updates are known as the Evolved UTRA (E-UTRA), and the Evolved UTRAN (E-UTRAN) respectively. The packet-switched core network of the EPS has also been redesigned and is called the Evolved Packet Core (EPC). The EPC is split into the Mobility Management Entity (MME) and the Server Gateway (SGW). The main goal of the EPC work is the convergence of heterogeneous access networks.

The main goal of the E-UTRAN architecture is a reduction of the cost and complexity of the equipment. This is achieved by removing one level of hierarchy in the E-UTRAN architecture as shown in figure 2.8 (figure
from [15]). Compared to the UTRAN architecture of figure 2.4, the main control node, the RNC, has been completely removed. The RNC functions of radio resource control, QoS control, and mobility control, have been integrated into the evolved Node B (eNodeB). The eNodeBs have logical interfaces X2 with neighboring eNodeBs and S1 with the EPC.

![Figure 2.8 E-UTRAN architecture](image)

This setup allows for a so called ‘flat architecture’ where intelligence is added to the Node B, the number of interconnections in the UTRAN is reduced and so is the complexity. This continued evolution of the UTRAN architecture shows that the access network is an area of much innovation and new and more flexible topologies are constantly needed.

In this chapter an overview of the UMTS architecture and its constituent parts was given. The UTRAN was described in some detail, explaining the
importance of the key elements such as Node B and RNC. Changes to the UTRAN architecture were also explored in relation to the evolution of the UMTS system and releases. The next chapter describes in some detail the UTRAN planning problem and outlines the optimization methods and solution approaches existing in the literature.
CHAPTER 3: UTRAN PLANNING AND OPTIMIZATION METHODS

3.1 The planning problem of UMTS Networks

Network planning and optimal topological design has been an area of extensive research for many years. The main goal is to identify geographical locations of nodes and the network connectivity to accommodate the expected traffic demand with a reasonable cost-performance trade-off. The design of cost-efficient telecommunications networks often involves solving several challenging combinatorial optimization problems simultaneously. In UMTS network planning it is required to solve the topological planning of both the RAN and the CN. Topological planning to a large extent determines the investment cost of the network and can have a considerable effect on the long-term network performance. As a consequence, a near-optimal topological design is of critical importance.

Assuming that future services will be based on IP networks[1, 3]; efficient transmission of IP packets over wireless connections is a pre-requisite for delivering successful broadband services in a mobile communications environment. Due to the high complexity and costs of future IP-based RAN architectures, it is appropriate to use algorithmic network topology optimization methods. Within the thesis UMTS as a third generation
platform for mobility and services is considered. In the context of this research the focus is on the topological design of the UMTS Radio Access Network (UTRAN) with the aim of solving the problems associated with cellular backhaul costs. In the following sections we look at the approaches to solving the UMTS and in particular the UTRAN planning problem.

3.2 The Modular Approach

The UMTS network planning problem is a very complex problem to solve, due to its complexity it has been divided into three different subproblems:

- The cell Planning Subproblem[22-26]
- The access network Planning Subproblem[6, 19, 27-29]
- The core network planning subproblem[30-32]

![Figure 3.1: UMTS Planning Subproblems](image)

We can see from figure 3.1(figure adapted from [33]) that the output from each subproblem acts as input to the successive subproblem and so on until we reach a final solution. This whole process is further complicated by the fact that each subproblem also has its own inputs. Due to the large numbers of parameters to optimize and the inter-dependency of these parameters, the UMTS network planning is a complex optimization task.
that belongs to the class of NP-hard problems[11]. Each of these subproblems has also been shown to be NP-hard [19, 31, 34]. The concept of hardness is discussed in more detail in section 3.5. The main goal of using the modular approach is to reduce the complexity of the overall network planning problem, as a result we end up with three different subproblems each one easier to solve than the whole problem. The modular approach allows us to tackle each subproblem in more detail facilitating better planning. Considering each subproblem independently can however have its disadvantages, namely resulting in local optimization. Most often, combining the solutions of each subproblem does not give an optimal solution to the global problem. Thus far, no integration strategies have been developed to combine independent solutions in order to obtain a global solution.

A different way of solving the planning problem consists of using a global approach which is discussed in section 3.3. A brief description of each subproblem is given in the following sections along with the related work that has been done in these areas. Because the focus of the research is the access network planning problem, only brief descriptions of the cell and core network planning problems are given. The access network planning problem is dealt with in some detail in section 3.4.

3.2.1 Cell Planning Subproblem
Cell planning can be regarded as the foundation of the cellular network design process and is of great importance for overall network quality. The
The cell planning problem consists in finding the number, the location and the types of the Node Bs subject to signal quality and coverage constraints. The general objective behind cell planning is to cover all mobile users within a given region while minimizing the cost of the Node Bs (including the installation cost). More precisely, the cell planning problem usually involves one or more of the following:

- Finding the optimal number of Node Bs.
- The best location to install the Node Bs.
- The type/model of Node B.
- The configuration of Node Bs.

The objective of cell planning problem can vary depending on the objective of the network planner. The network planner is usually interested in minimizing the cost of the network, maximising the coverage in a given area, and maximising the signal quality. Most often these objectives are contradictory. If the network planner wants for example to maximise the coverage area he would need to deploy more Node Bs, thus increasing the cost of the network. In their paper, Yang et al [22] discuss the idea of a weighted multi-objective function with weights (0 to 1) assigned to different objectives. By assigning weights to the objectives the network planner has more flexibility and can prioritise objectives as needed.

The main inputs to the cell planning problem include, the set of possible sites to install the Node Bs, the different types of Node Bs, the cost of the Node B types, and the estimated traffic (in erlang) from each given
requested coverage area. Interestingly, Brigitte and Sebbah [35] add an extra input to the cell planning problem. They include the “handshake” between the radio and the core networks, taking into account the cost of the base station location and the wired link capacity of both the radio and the core networks. For a detailed description of cell planning and solution algorithms the reader is directed to [4, 23, 36]. The output from the cell planning problem acts as input to the access network planning problem.

3.2.2 Core Network Planning

The UMTS core network planning involves the design of the core network elements and interfaces (MSC/SGSN) as well as the interconnection with the RNCs. The core network planning considers the interconnection between the core network and access network, as well as the interconnection between the core network and the external network (i.e. Public Switched Telephone Network (PSTN), Internet Service Provider (ISP), etc). Given the location and the traffic going through each RNC, core network planning subproblem consists in determining:

- The number, the location, and the type of MSCs and SGSN;
- The assignment of the RNCs to the MSCs and SGSNs;
- The number, the location, and the type of links used to connect the RNCs to the MSCs and SGSNs;

The objective is to minimize the overall cost of the core network (including, the cost of the MSCs, the cost of the SGSNs, and the cost of the links and interfaces) subject to a number of assumptions and constraints. Very little research has been done on the core network planning problem, mainly
because it is similar to the wired network planning problems for which many solutions exist. Ricciato et al[30] and Harmatos[31] provide further reading on the core network problem.

3.3 Global Approach

Instead of solving a single subproblem, several researchers focused on more than one simultaneously. Since the global approach focuses on the interactions of the subproblems, it has the advantage of potentially providing solutions that are closer to the global optimum. The problem however becomes considerably more complex because as mentioned in section 3.2, each subproblem is NP-Hard. St-Hilaire et al [37] [38] propose an integrated approach in which all three subproblems are considered simultaneously. They developed a mathematical model in [32] to plan the UMTS network in the uplink direction, and gave a detailed example in order to compare the global approach with the decomposition approach when the problems are solved successively.

In [33] they try to address the complexity of the global approach by using heuristics based on local search and tabu search. More recently Liu and St-Hilaire [39] develop an approximate algorithm based on simulated annealing algorithm to solve the global planning problem. They conclude that the main advantage of using such an approach is the speed up in terms of the CPU (central processing unit) execution time, allowing larger instances of the problem to be tackled. In view of the fact that proposed
changes to existing 3G systems will take place mainly in the RAN, the work of this thesis focuses on the optimization of this subproblem.

3.4 Access Network Planning Subproblem

The access network planning subproblem as shown in figure 3.2 (figure adapted from [19]) is concerned with RNC planning and dimensioning, the interconnection topology, and the sizing of the interconnection links connecting the Node Bs to the RNCs. In figure 3.2 the general tree topology of the UTRAN is depicted showing the location of the Node Bs and RNCs and the interconnection links between them. The inter-RNC links in blue represent the connection between RNCs that facilitate handover on the Iur interface. The Node Bs communicate directly with the mobile devices of users such as cellular phones, notebooks, and smartphones, collect the traffic of a small region and forward it to their controlling RNC using the so-called ‘access links’. The objective is to minimize the cost of the access network (including the cost of RNCs and the cost of the links and interfaces). There are a number of inputs required to plan the access network including:

- The location of the Node Bs.
- The traffic demand going through each Node B.
- The set of potential locations to install the RNCs.
- The handover frequency between adjacent cells.
- The different types of links available to connect the Node Bs to the RNC.
In the case of UMTS access networks the topology is generally restricted to that of a tree. Node Bs can be connected either directly to an RNC or to other Node Bs in a cascaded way. In order to take advantage of the economies of scale, traffic is aggregated on the links interconnecting Node B and RNC. The ‘heaviest’ links are designed to be those that are connected directly to the RNC as they must be able to handle all of the traffic in their subtree. The RNC is the central source connecting all other nodes. Nodes Bs however do not have any routing capability, they simply forward all received traffic to their controlling RNC. As a consequence of this arrangement the access network is divided into independent trees of Node Bs each rooted at an RNC. These trees are the Radio Network Subsystems (RNS) described in section 2.3. Due to technical limitations of the equipment[17, 40], very strict topology constraints apply in the case of tree topologies, these include:
• There is a significant delay in communication caused by a combination of the limited resources of the Node B and the relatively low bandwidth of the access links. To reduce this delay, the maximum number of the access links on a routing path is kept low by limiting the depth of a tree.

• A Node B can be connected to a limited number of other Node Bs. One simple reason is that commercial devices have a limited number of ports. In addition, if several Node Bs are close to each other, this may cause interference problems in their radio interface if they are interconnected using microwave links.

• The number of Node Bs connected to a RNC is also limited. This again is because of the limited number of ports or interfaces in the RNC.

Given the inputs and the constraints outlined, the task of designing the access network can generally be decomposed into the following tasks:

• Finding the cost-optimal number and location of RNCs in the network

• The cost-optimal planning of the physical links connecting Node Bs to their corresponding RNCs taking into account specific topological constraints.

The total cost of the UTRAN is made up of these two variable factors: the cost of the RNCs, and the cost of the access links. These two tasks are interdependent and optimal solutions for each may not necessarily
combine easily to give a good overall solution. The task of planning the access network has been shown to be NP-hard[19]. It is therefore appropriate to use heuristic methods that enable the determination of a near-optimum network topology more easily. This subproblem is similar to the cell to switch assignment problem in 2G networks[41] and can be formulated as follows.

**Objective:** Minimize the sum of the cost of the RNCs and the cost of the access links.

**Subject to**

- Each Node B is connected to exactly one RNC;
- At most one RNC can be installed at an RNC site;
- The number of links connected to a Node B cannot exceed the maximum number of interfaces that can be installed in that Node B;
- The number of links connected to an RNC cannot exceed the maximum number of interfaces that can be installed in that RNC;
- The sum of the traffic (in circuits and in bps) from the Node Bs connected to a particular RNC cannot exceed the capacity (in circuits and in bps) of that RNC;
- The sum of the capacities of the links (in circuits and in bps) connected to an RNC cannot exceed its capacity (in circuits and in bps);

Generally the objective function consists to minimize the cost of the access network; other objectives such as reliability can also be considered.
3.5 Topological Network Design

The topological design of cellular networks have long since provided a series of combinatorial optimization problems, from the design of tree and ring networks to meshed network architectures that guarantee a higher level of reliability between every pair of network nodes. The main constraints on such design problems generally are:

- Capacity of the links and equipment.
- Maximum network diameter
- Maximum node degree.
- Uncertain and variable demand.

The primary design problem can generally be specified as follows: Given a set of network nodes, it is required to find the minimum cost topology amongst all possible topologies, which satisfy a set of given constraints. The objective is to balance the overall investment in the network (installation of links and equipment), versus the operational cost. Fault tolerance is one aspect of network design that concerns the ability of network nodes to communicate in the presence of faulty links and/or nodes. Topological network design with reliability constraints is NP-hard.

Network topology design problems are generally very complex and solution approaches can generally be simplified by grouping them into three categories.

- Hierarchical techniques[42].
- Enumerative techniques[43].
Hierarchical approaches for topological network design are based on the idea of partitioning the set of network sites into a number of clusters to satisfy the given design objectives. This method is used to design a network that interconnects \( n \) nodes under the constraints that the network is single node survivable, the diameter of the network is bounded by some specified bound \( d \), and the maximum node degree is bounded by some specified bound \( \Delta \)[45]. A node represents network equipment such as the Node B or an RNC, the diameter bound \( d \) ensures efficient use of node resources and limits the maximum delay, and the node degree bound \( \Delta \) models the limited number of physical ports on a node. The single node network survivability is denoted by \( \rho^N(n,d,\Delta) \), while the single link survivability is denoted by \( \rho^L(n,d,\Delta) \).

![Figure 3.3: Example of the clustering method](image)

Figure 3.3 shows an example of the clustering method applied to a \( \rho^L(9,3,4) \) network, i.e. \( n=9, d=3, \) and \( \Delta=4 \). The network is single link...
survivable since if any link within a cluster fails, a path still exists between any two nodes.

Enumerative techniques search all possible paths between a given source and destination nodes, while selecting the most appropriate subset of links to interconnect them. They are generally very simple to implement but are computationally expensive and so can only reasonably be used for optimizing the design of networks having relatively small number of nodes and/or links. This approach is particularly suited when network reliability is the main objective.

Enumerative algorithms start by enumerating all possible spanning trees in the given network, the cost of a spanning tree is then the sum of the costs of its constituent links and its reliability is the product of the reliabilities of its constituent links. For example, consider the network shown in figure 3.4 (figure from [43]) that has the following specifications:

<table>
<thead>
<tr>
<th>Link</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>2.0</td>
<td>3.7</td>
<td>2.7</td>
<td>2.5</td>
<td>4.0</td>
<td>3.0</td>
<td>3.2</td>
<td>3.5</td>
</tr>
<tr>
<td>Reliability</td>
<td>0.9</td>
<td>0.6</td>
<td>0.8</td>
<td>0.5</td>
<td>0.9</td>
<td>0.7</td>
<td>0.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

The total cost allowed for this network is $\text{Cost}_{\text{max}} = 16$ units.
First, all possible spanning trees of the network are determined, among these one which yields the maximum reliability to cost ratio is selected, as shown in figure 3.5 (a) (figure from [43]). The spanning tree $acef$ is selected as it yields the maximum reliability to cost ratio. The cost of this network is $11.7 \ (2.0 + 2.7 + 4.0 + 3.0)$ and the reliability is $0.4536 \ (0.9 \times 0.8 \times 0.9 \times 0.7)$. Another spanning tree is then added that has the highest distance from the initial one such that the maximum network cost (16) is not exceeded. Based on this criterion, spanning trees are continuously added respecting the network cost limitation. The final optimized network is shown in figure 3.5 (b).
Iterative techniques are perhaps the most popular choice for topological design of networks as they can achieve near optimal network design in a reasonable time. The quality of an iteration method can be judged by its speed of convergence and has this distinct advantage over enumerative techniques. There are a wide variety of iterative techniques reported in the literature [38, 46, 47], these include genetic algorithms (GA), simulated annealing (SA) and tabu search (TS). The SA and GA approaches are explained in some detail in section 3.5.

This thesis addresses a UMTS radio access network design problem. At an abstract level this problem falls under the design of networks with tree topologies. Tree topology design falls under the broader set of problems known as centralized network design problems. In a centralized network communication is to and from a single site [5]. In this scenario sites other than the central site have equipment that is generally not capable of making routing decisions. This implies a tree topology with a single path to the central site and by extension between any pair of nodes.

Terminals may be connected directly to the central site in a star topology or multipoint lines are used where groups of terminals share a tree to the central site [48]. Each subtree rooted in the central site corresponds to a multipoint line. Usually the central site can handle a given fixed amount of information in communication. This results in a restriction on the amount of information flowing in any link adjacent to the central site to that fixed
amount. Other topologies such as ring and mesh can be explored for the UTRAN design but this is beyond the scope of the research. I have chosen to employ iterative techniques in solving the UTRAN topology design problem, firstly because of their wide applicability, and secondly due to the complexity of the design problem. Because of the implicit tree topology of the UTRAN, it is sensible to take a closer look at the design of tree topologies.

3.5.1 Design of Tree Topologies

To enable communication between a pair of network nodes, either directly or indirectly via other network nodes, there must exist at least one path between them. In topology network design problems costs are associated with the usage of a potential link between any two given nodes. These costs can represent various real cost factors such as the laying of a cable or the leasing of a virtual connection. An abstraction from the practical application of the problem can be achieved by the introduction of graphs. Let the graph \( G = (V, E) \) be a connected, undirected graph where \( V \) is the set of network nodes, \( E \) is the set of possible interconnections between pairs of network nodes, and for each edge \((i, j) \in E\) we assign a cost or weight \( k_{ij} \in \mathbb{Q} \), denoting the installation costs of a link between the network nodes.

A second graph \( H = (U,F) \) with \( U \subseteq V \) encodes all required communication paths. An edge \((i, j) \in F\) exists if and only if the topology solution should
contain a path between the two network nodes. In order to design a minimum cost network topology that enables communication between all nodes, a minimum cost subset of the edges \( L \subseteq E \) has to be selected as the network topology. This is the minimum weight spanning tree problem (MWST)[49]. If a connecting path is required between all node pairs \((i, j) \in V\) at least \(|V| - 1\) edges have to be selected and the resulting subgraph \((V, L)\) must be cycle free. These subgraphs are the spanning trees in \(G\) and the minimum cost network topology is a minimum cost spanning tree \(T = (V, L)\).

This can be formulated as an integer constrained problem in the following way. Let \(x_{ij}\) be a 0-1 integer variable indicating whether arc \((i, j)\) belongs to the spanning tree and let \(S \subseteq V\).

\[
\text{Minimize } \sum_{(i,j) \in E} k_{ij} x_{ij}
\]

Subject to \[
\sum_{(i,j) \in E} x_{ij} = N - 1
\]

\[
\sum_{i \in S, j \in S} (x_{ij} + x_{ji}) \geq 1 \quad \forall \text{ Non-empty proper subsets S of nodes}
\]

\[
x_{ij} = 0 \text{ or } 1, \quad \forall (i,j) \in E
\]

The first two constraints ensure that the graph defined by the set \{\((i,j) \mid x_{ij} = 1\}\) has N-1 arcs and is connected. It is therefore a spanning tree. The MWST problem can be solved with the use of a greedy algorithm such as the Prim-Dijkstra algorithm (table 3.2). The algorithm selects repeatedly
the minimum cost edge extending the current tree, starting with a single
vertex, and ending with a tree spanning all vertices.

Table 3.2: Prim-Dijkstra algorithm

<table>
<thead>
<tr>
<th>Dijkstra-Prim algorithm to determine the minimum cost spanning tree $T = (V, L)$ in a graph $G = (V, E)$ with edge cost $k_e$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $L := \emptyset$</td>
</tr>
<tr>
<td>Let $S := {i}$ for some arbitrary $i \in V$</td>
</tr>
<tr>
<td>while $S \neq V$ do</td>
</tr>
<tr>
<td>Let $e = \arg\min \left{k_{ij} \mid ij \in E, \ i \in S, \ j \in V \setminus S \right}$</td>
</tr>
<tr>
<td>$L := L \cup {e}$</td>
</tr>
<tr>
<td>$S := S \cup {j}$ with $e = {i, j}, \ i \in S$</td>
</tr>
<tr>
<td>return $T = (V, L)$ with cost $\kappa(L)$</td>
</tr>
</tbody>
</table>

There are however many variations of the spanning tree depending on the design task at hand. In the case of the UTRAN topology design problem, the aggregation trees have a specified depth limit and there are certain degree constraints for intermediate nodes in each tree level. In each aggregation tree, the transmission flows through special higher capacity nodes upwards to the root node (RNC) through transmission links. In addition the root node also has its own capacity constraints, making the problem computationally hard. Some existing practical approaches [19, 50, 51] in UTRAN topology planning use a generally applicable local search heuristic, such as simulated annealing (SA), which is often combined with greedy heuristic sub-phases. These approaches have the advantage of being able to deal with any kind of cost function. Future 3G networks will require several changes to the RAN to accommodate increased and
varying traffic, among which it is expected that possible changes to the RAN topology will be necessary. The next section assesses the changes to the 3G RAN and how they might affect the choice of topology.

3.6 Related Work

Research on the optimization of access networks has focused mainly on designing tree topologies that minimize cost. The tree topology has proven to be the most cost-effective for our design problem; however, it has the distinct disadvantage of being sensitive to any kind of failures. The RNC at the root of each tree represents a single point of failure, and although these failures rarely occur in practice, it may become much more of a problem as future networks continue to grow in size. Juttner et al [19] propose algorithmic solutions that optimizes simultaneously the cost and location of the RNCs as well as their connections to the Node Bs. They first propose a global algorithm which combines Simulated Annealing (SA) with the solution of a specific b-matching problem to create a number of independent trees. Because of the difficulty in applying SA directly to the problem, the b-matching algorithm is introduced and performed within each step of the SA process. The second method uses a Lagrangian relaxation based lower bound computation with branch-and-bound to design a single tree with a predefined RNC node. Wu and Pierre [52] propose a constraint-based optimization model used to find the optimal location of RNCs while simultaneously optimally assigning the Node Bs to the RNCs. Their model follows closely the RNC constraints outlined in section 3.4. Of significance is the fact that their objective function also
includes complex handover cost and the cost of simple handovers. Most of research works do not include these costs, but as we saw in section 2.3 they can affect overall network cost.

More recently, Kim et al [51] provide a mathematical formulation of the access network design problem with constraints on RNC and Node B capacities, along with a lower bounding method. They develop a heuristic algorithm based on simple add/drop and move/exchange procedures, along with the lower bounding method. Another valuable work that is capable of planning the access network recently published in Mehdi Hashemi et al [50]. Mehdi Hashemi et al develop a hybrid ant colony optimization (ACO) algorithm to tackle the problem heuristically. They first make a savings-based greedy algorithm to design the UMTS access network, then perturb the greedy algorithm to make the ACO algorithm for designing the UMTS access network. Finally, they develop the ACO algorithm using a divided and conquer technique based on decomposition ants (D-ants). The main idea in this part is to decompose large problems into a number of smaller disjoint problems that can be solved more efficiently by the ACO algorithm. Faigl et al tackle a related problem of optimizing the underlying transmission network topology (described in section 2.4). The UTRAN topology design is done from a logical viewpoint taking into consideration the logical interfaces between network elements. Transmission topology optimization by comparison includes the underlying network elements such as switches, and considers factors such as the
traffic conditions of the Node Bs and capacities of available transmission links. Faigl et al [53] propose a two phase approach to the problem, first they introduce a tree planning procedure that creates a directed spanning forest using given topology constraints. Secondly, they develop a mixed integer program to assign the link capacities to the graph from the first phase and choose one having minimal cost.

Some researchers have tackled individual aspects of the access network planning problem, but not the problem as a whole. Tian et al [54] tackle the problem of optimizing the physical links connecting the Node Bs to the RNCs. They assume that the layout of the Node Bs and RNCs is given and don’t deal with the problem of finding the number and location of the RNCs. They discuss the use of fixed Wimax as an alternative topology to the traditional T1/E1 links. Similarly, Diallo et al [29] also tackle the assignment of Node Bs to RNCs in the form of the cell to switch assignment problem in UMTS networks.

They go one step further and include in their consideration the assignment of RNCs to MSCs and SGSNs. They propose a mathematical model for the cell assignment problem and a tabu search heuristic to solve it. As indicated earlier, the access network planning problem is similar to the cell to switch assignment in 2G networks. Quintero and Pierre [16] also address the problem of assigning Node Bs to RNCs and propose a multi-population hybrid genetic algorithm with migration to solve the problem.
Similar to Wu and Pierre they consider the notion of complex and simple handovers when assigning Node Bs to RNCs.

3.6.1 Planning Reliable Access Networks

One of the key challenges in the migration to 4G mobile systems is to minimize the failures and their potential impacts in any level of tree-like topology in the access network. For this reason, the issue of network reliability/survivability has received more attention in recent times. Network operators can guarantee reliability only if the networks are equipped with survivability mechanisms that are able to minimize the impact of network failures. For example, a single cable cut causes a network separation into two separate sub-trees leading to the disconnection of one or more nodes. There are several strategies for designing fault tolerant network topologies given that different network topologies yield different values of network reliability.

The reliability of tree-like UTRAN topologies can be increased by inserting new links into the network [55]. This approach provides a more efficient protection strategy, since in this case we create alternative paths to bypass the most failure sensitive parts. In addition, sufficient spare capacity can be allocated in the network infrastructure in advance to help mitigate effects due to system level failures such as loss of links or nodes. These approaches however leads to an increase in network cost due to the additional links and spare capacity, and solutions are needed that provide a good compromise between network cost and reliability. Other
network topologies like self-healing ring and mesh topologies are capable of rerouting traffic around the single point of failure through other existing facilities and switches. In this regard topologies other than the tree have also been explored for connections in the UTRAN.

In a series of papers, Charnsripinyo et al [28, 56, 57] focus on the design of fault-tolerant 3G access network incorporating the effects of user mobility and the packet based nature of 3G traffic. In [23] and [42] they adopt a two phase design methodology to produce a survivable network topology. In the first phase they formulate the problem as a mixed integer programming model and solve to obtain a minimum cost network topology, the second phase augments the network topology from phase one in order to satisfy survivability requirements. They apply this approach to different types of network topologies including a mesh topology in [23] and a tree topology in [42]. In [43] they present a mixed integer-programming model which incorporates reliability in its formulation by generating a multi-ring network design. The significance of their work comes from their inclusion of the effects of user mobility after base station failures.

In a similar approach to Charnsripinyo et al, Eksook and Pronmak [58] present a 0-1 integer programming model for designing optical access network topologies comprised of multi-rings. The objective is to maximize the number of rings used in the access network topology to achieve the highest network reliability. The total cost of the network is used as a
constraint in their design formulation. The choice of objective is of course
down to the network operator as network cost and reliability are conflicting
objectives. In this regard Dharmaraja et al [59] provide an analytical
model to determine reliability and survivability attributes of 3G UMTS
networks.

The UMTS network is modeled using stochastic models such as Markov
chains, semi-Markov process, reliability block diagrams and Markov
reward models to obtain these attributes. They investigate the impact of
failures in UTRAN elements such as Node B and RNC and conclude that
while substantial gains in network reliability can be achieved by
incorporating fault tolerance in the UTRAN architecture, increasing the
number of network elements at any level beyond a maximum number will
not enhance the network reliability. Dharmaraja et al claim that their model
can help to guarantee network connectivity after any failure without over
dimensioning the network.

3.7 Optimization Methods for Access Network Planning
UMTS access networks (UTRAN) have become a critical part of the
telecommunications infrastructure, providing voice, high bit-rate data and
multimedia services to mobile users. The topological design of the UTRAN
has been an important area of research in recent years. Recent studies
are given in references [27, 28, 33, 36]. The design problem is generally
formulated as a constrained optimization problem where the aim is to find
a network topology such that an objective function is optimized, subject to
a set of constraints. In this scenario we are aiming to find the ‘best’ among many possible candidate solutions to the problem. There are an intractably large number of ways to design the topology of the UTRAN, it still remains that the number of possible candidate solutions which meet the main constraints is still too large for us to hope to examine each of them in turn.

The size and complexity of modern UTRAN architectures makes it difficult to apply traditional optimization methods (Integer Programming, Dynamic Programming, etc) to find efficient solutions. Instead, modern approaches called ‘Metaheuristics’ or simply ‘heuristics’, are being applied as they usually take much less effort to develop and are fundamentally easier to apply. Heuristics are highly general in their applicability and can be applied in most situations where there is an acceptable method of scoring or evaluating candidate solutions to a problem. The techniques essentially fall into two groups: local search and population-based search. In the following sections we describe some of these heuristic techniques and their application to the UTRAN topology design problem. We focus on genetic algorithms (GA) which we use to conduct our experiments described in chapter 5.

3.7.1 Local Search

Local search is one of the oldest and simplest metaheuristic methods[60]. It starts at a given initial solution, and, at each iteration the heuristic replaces the current solution by a neighbor that improves the objective function. The search stops when all candidate neighbors are worse than
the current solution (i.e. a local optimum is reached). The main objective of this restricted neighborhood strategy is to speed up the search. Algorithm 3.1 illustrates the template of a local search algorithm:

**Algorithm 3.1**

\[ s = s_0; \quad /* \text{Generate an initial solution } s_0 */ \]

**While** not Termination Criterion **Do**

- Generate \( (N(s)) \); /* Generation of candidate neighbors */
- **If** there is no better neighbor **Then** Stop;
- \( s = s'; \quad /* \text{Select a better neighbor } s' \in N(s) */ \)

**Endwhile**

**Output** Final solution found (local optima)

In general, local search is a very easy method to design and implement and gives fairly good solutions very quickly. One of the main disadvantages of local search however is its convergence towards local optima. Many alternative algorithms have been proposed to avoid becoming stuck in local optima. Simulated annealing (SA) and Tabu Search (TS) are just two such examples. These two heuristic approaches have been used to address various optimization problems in UMTS access network design. Tabu search behaves like a steepest local search algorithm, but it accepts nonimproving solutions to escape from local optima when all neighbors are nonimproving solutions. Usually, the whole neighborhood is explored in a deterministic manner, whereas in SA a random neighbor is selected. Diallo *et al* [29] use tabu search to solve the
problem of assigning Node Bs to RNCs in the UMTS access network. Chamberland [61] also uses tabu search to tackle the problem of updating the access network in order to connect new subscribers and to satisfy the new class of service requirements for the existing subscribers.

Simulated annealing has had a major impact on the field of heuristic search for its simplicity and efficiency in solving combinatorial optimization problems. The SA algorithm simulates the energy changes in a system subjected to a cooling process until it converges to an equilibrium state. It is based on the principles of statistical mechanics whereby the annealing process requires heating and then slowly cooling a substance to obtain a strong crystalline structure. The strength of the structure depends on the rate of cooling metals. If the initial temperature is not sufficiently high or a fast cooling is applied, imperfections are obtained. In this case, the cooling solid will not attain thermal equilibrium at each temperature.

Table 3.3 illustrates the analogy between the physical system and the optimization problem. The objective function of the problem is analogous to the energy state of the system. A solution of the optimization problem corresponds to a system state. The decision variables associated with a solution of the problem are analogous to the molecular positions. The global optimum corresponds to the ground state of the system.
Table 3.3: Analogy between physical system and the optimization problem

<table>
<thead>
<tr>
<th>Physical System</th>
<th>Optimization Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>System State</td>
<td>Solution</td>
</tr>
<tr>
<td>Molecular Positions</td>
<td>Decision Variables</td>
</tr>
<tr>
<td>Energy</td>
<td>Objective Function</td>
</tr>
<tr>
<td>Ground State</td>
<td>Global Optimum Solution</td>
</tr>
<tr>
<td>Metastable State</td>
<td>Local Optimum</td>
</tr>
<tr>
<td>Rapid Quenching</td>
<td>Local Search</td>
</tr>
<tr>
<td>Temperature</td>
<td>Control Parameter $T$</td>
</tr>
<tr>
<td>Careful Annealing</td>
<td>Simulated Annealing</td>
</tr>
</tbody>
</table>

SA proceeds in several iterations. In each iteration, a random neighbor is generated. Moves that improve the cost function are always accepted. Otherwise, the neighbor is selected with a given probability that depends on the current temperature and the amount of degradation $\Delta E$ of the objective function. The probability that such moves are accepted decreases with each iteration. This probability follows the Boltzmann distribution:

$$P(\Delta E, T) = e^{-\frac{f(x')-f(x)}{T}}$$

The probability function uses a control parameter called temperature to determine the probability of accepting nonimproving solutions. Once an equilibrium state is reached the temperature is gradually cooled according to a cooling schedule. Table 3.4 (table from [60]) illustrates the template of the SA. In addition to common design issues for local search heuristics, the main design issues specific to simulated annealing are:
- The acceptance probability-- This is the main element of SA that enables nonimproving neighbors to be selected.
- The cooling schedule-- The cooling schedule defines the temperature at each step of the algorithm

### Table 3.4: Template of Simulated Annealing Algorithm

<table>
<thead>
<tr>
<th>Algorithm 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Cooling schedule.</td>
</tr>
<tr>
<td>s = s₀ ; /* Generation of the initial solution */</td>
</tr>
<tr>
<td>T = T_max ; /* Starting temperature */</td>
</tr>
<tr>
<td><strong>Repeat</strong></td>
</tr>
<tr>
<td>Repeat /* At a fixed temperature */</td>
</tr>
<tr>
<td>Generate a random neighbor s' ;</td>
</tr>
<tr>
<td>( \Delta E = f(s') - f(s) ) ;</td>
</tr>
<tr>
<td><strong>If</strong> ( \Delta E \leq 0 ) <strong>Then</strong> s = s' /* Accept the neighbor solution */</td>
</tr>
<tr>
<td>Else <strong>Accept</strong> s' with a probability ( e^{-\frac{\Delta E}{T}} ) ;</td>
</tr>
<tr>
<td><strong>Until</strong> Equilibrium condition</td>
</tr>
<tr>
<td>/* e.g. a given number of iterations executed at each temperature T */</td>
</tr>
<tr>
<td>T = g(T) ; /* Temperature update */</td>
</tr>
<tr>
<td><strong>Until</strong> Stopping criteria satisfied /* e.g. T &lt; T_min */</td>
</tr>
<tr>
<td><strong>Output:</strong> Best solution found</td>
</tr>
</tbody>
</table>

#### 3.7.2 Population-based

Population-based metaheuristics start from an initial population of solutions. Then, they iteratively apply the generation of a new population and the replacement of the current population. In the generation phase, a new population of solutions is created. In the replacement phase, a selection is carried out from the current and the new populations. This process iterates until a given stopping criteria. The search process is stopped when a given condition is satisfied (stopping criterion).

Population-based metaheuristics include algorithms such as evolutionary
algorithms (EAs), scatter search (SS), and particle swarm optimization (PSO) amongst others. Table 3.5 (table from [60]) illustrates the template of a population-based metaheuristic.

Table 3.5: Template of a population-based heuristic

<table>
<thead>
<tr>
<th>Algorithm 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = P₀; /* Generation of the initial population */</td>
</tr>
<tr>
<td>t = 0;</td>
</tr>
<tr>
<td>Repeat</td>
</tr>
<tr>
<td>Generate(P_t); /* Generation a new population */</td>
</tr>
<tr>
<td>P_{t+1} = Select-Population(P_t \cup P_t'); /* Select new Population */</td>
</tr>
<tr>
<td>t = t + 1;</td>
</tr>
<tr>
<td>Until Stopping criteria satisfied</td>
</tr>
<tr>
<td>Output: Best solution(s) found</td>
</tr>
</tbody>
</table>

Population-based metaheuristics differ in the way they perform the generation and the selection procedures and the search memory they are using during the search. Depending on the strategy used to generate the population, population-based metaheuristics may be classified into two main categories:

- **Evolution based** - solutions composing the population are selected and reproduced using variation operators (e.g., mutation, recombination) acting *directly* on their representations. Evolutionary algorithms (EAs), and scatter search (SS) are examples of this class of metaheuristics.

- **Blackboard based** - the solutions of the population participate in the construction of a shared memory. This shared memory will be the main input in generating the new population of solutions. Ant colonies are a well known example of this class of metaheuristics. A
discussion of genetic algorithms now follows as they are the optimization method chosen for the thesis.

**Genetic algorithms**

Genetic algorithms employ a strategy of imitating evolutionary systems by using the principle of ‘survival of the fittest' to guide the search process[62]. Thus, genetic algorithms present a unique and distinctly different approach to the access network design problem. Genetic algorithms create a population of candidate solutions to the problem and let them evolve over multiple generations to find increasingly better solutions. Each individual in the population is called a chromosome, representing a solution to a given problem. The *fitness* of an individual is a measure of how good the solution represented by the individual is.

To create the next generation, individuals are selected for breeding (cross-over and/or mutation) based on their fitness values. The flowchart of a simple genetic algorithm is presented in figure 3.6 (figure from [44]). The flowchart implements a proportional selection operator and a generational replacement, i.e. the population of the offspring replaces that of the parents completely. Another version uses a steady state replacement and is used to conduct experiments later in the thesis.
In the traditional genetic algorithm, the representation used is a \textit{fixed-length bit string}. Each position in the string is assumed to represent a particular feature of an individual, and the value stored in that position represents how that feature is expressed in the solution. Each string is analogous to genes in biological organisms. The representation being based on bit strings, the difficulty is to discover a good coding of the chromosome, such as variation operators producing feasible offspring and satisfying the constraints of the problem.

\textbf{3.7.3 Constraint Handling in GAs}

An important issue when dealing with constrained optimization problems is the handling of infeasible solutions. There are several approaches proposed in GAs to handle constrained optimization problems. These approaches can be classified into four main groups:

- Methods based on penalty functions (e.g. Static Penalties).
- Methods based on a search of feasible solutions (e.g. repairing infeasible individuals).
- Methods based on preserving feasibility of solutions (e.g. searching the boundary of feasible region).
- Hybrid methods.

Here we will concentrate on penalty functions as the method of choice for dealing with infeasible solutions to our design problem.

**Penalty Functions**

The solution or set of solutions which are obtained as the final result of the genetic algorithm search must necessarily be feasible, that is, satisfy all constraints. Difficulty in satisfying constraints will increase (generally more than linearly) with the number of constraints. The penalty function transforms a constrained problem to an unconstrained one in two ways.

The first is to use the additive form as follows:

\[
eval(\vec{x}) = \begin{cases} 
  f(\vec{x}), & \text{if } \vec{x} \in F \\
  f(\vec{x}) + p(\vec{x}), & \text{otherwise}
\end{cases}
\]

Where \( p(\vec{x}) \) presents a penalty term. If no violation occurs, \( p(\vec{x}) \) will be zero and positive otherwise. Under this conversion, the overall objective function now is \( \eval(\vec{x}) \) which serves as a fitness function in GAs.

The second method uses the multiplicative form as follows:

\[
eval(\vec{x}) = \begin{cases} 
  f(\vec{x}), & \text{if } \vec{x} \in F \\
  f(\vec{x})p(\vec{x}), & \text{otherwise}
\end{cases}
\]

For minimization problems, if no violation occurs, \( p(\vec{x}) \) is one and bigger than one otherwise.
If either the penalty is too large or too small, the problem could be very hard for GAs. A big penalty prevents searching the infeasible region, in which case the GA will converge to a feasible solution very quickly even if it is far from the optimal. A small penalty will cause to spend so much time in searching an infeasible region that the GA would converge to an infeasible solution. In the area of combinatorial optimization, the popular Lagrangian relaxation method[63] employs a variation on this theme for dealing with constrained problems: temporarily relax a problem’s most difficult constraints, using a modified objective function to avoid straying too far from the feasible region.

This chapter described the UTRAN planning problem in some detail and outlined the existing optimization methods and solution approaches to tackling the problem. The next chapter presents the formulation of the UTRAN topology planning problem, the solution approach taken in this thesis, and detailed descriptions of the heuristic algorithms to be implemented.
CHAPTER 4: UTRAN TOPOLOGY PLANNING PROBLEM

4.1 Introduction

In this chapter, the UTRAN topology planning problem is formulated as a capacitated p-median problem. The objective is optimization of link cost, and handover cost, whereas RNC capacity is considered as a constraint. The objective functions for the estimation of costs of the above objectives as well as RNC capacity constraint are modeled. Section 4.2 provides a summary of the objectives of UTRAN topology planning and optimization. In section 4.3 we discuss and formulate the capacitated p-median problem. In section 4.3 we discuss and formulate the specific problem this thesis addresses, while section 4.5 looks at the solution approaches to the design problem. We give an accessible description of the underlying principles of iterative heuristic algorithms and a discussion of basic implementation issues.

4.2 Objectives of UTRAN Topology Planning and Optimization

Due to the increasing complexity of UTRAN architectures, one of the most critical issues regarding the design of UMTS access networks pertains to the assignment of Node Bs to radio network controllers RNCs, which is an NP-hard problem. Hence, for large mobile networks, this problem cannot be practically solved by using exact methods. UTRAN topology planning is a subset of the total wireless network planning, both when creating a new
network or extending an existing one. The goal is to build an access network that will provide a reliable transmission network capable of delivering enough capacity for present needs as well as ensuring seamless expansion in the future. The cost of transmission for Node B access as well as inter-MSC backbone networks is significant; typically twenty to twenty-five percent of infrastructure costs[64]. Its impact on operator profitability is also influenced by the way the network roll out speed and transmission-related downtime affect the customers’ overall perception of service quality.

The access network can be presented either with a logical or a physical view. The logical view is useful in initial network dimensioning and identifies where a connection is needed in order to transport traffic from source to destination. The physical view works together with the logical to help build an optimized network. The total operating cost of a cellular network includes three components [16]:

- The cost of the links between Node Bs and their controlling RNC.
- The monthly amortization cost of installed RNCs.
- The cost of handovers between Node Bs.

The discussion of handover costs is beyond the scope of the thesis and is omitted from the problem formulation. The problem assigning Node Bs to RNCs essentially consists of finding a configuration that minimizes the total network cost. In this thesis solutions to the UTRAN topology planning
problem are tackled by modelling the problem as a capacitated p-median problem. A discussion of the capacitated p-median problem now follows.

4.3 The Capacitated p-median Problem

Informally, the goal of the p-median problem is to determine p facilities in a predefined set with n (n > p) candidate facilities in order to satisfy a set of demands, so that the total sum of distances between each demand point and its selected facility is minimized. Assuming all vertices of a graph are potential medians, the p-median problem can be formally defined as follows. Let \( G = (V, E) \) an undirected graph where \( V \) are the vertices and \( E \) are the edges. The goal is to find a set of vertices \( V_p \subset V \) (median set) with cardinality \( p \), such that: (i) the sum of the distance between each remaining vertex in \( V - V_p \) (demand set) and its selected vertex in \( V_p \) be minimized; and (ii) all demand points are satisfied without violating the capacity restrictions of the median facilities.

Compared to the p-median problem, the capacitated p-median problem has the following additional constraints: (i) each facility can satisfy only a limited number of demands (capacity restriction); and (ii) all demand points must be satisfied by respecting the capacities of the facilities selected as medians. An integer programming formulation of the capacitated p-median follows:

Let \( V = \{1, 2, \ldots, n\} \) be the set of vertices to allocate and also of possible medians. Let \( p \) represent the number of medians to be selected, and let
$q_i$ be the demand of vertex $i$. Let $Q_j$ be the capacity of median $j$, $[d_{ij}]_{n \times n}$ be a distance matrix. Let $[x_{ij}]_{n \times n}$ be an allocation matrix, with $x_{ij} = 1$ if demand point $i$ is allocated to median $j$, and $x_{ij} = 0$, otherwise; $x_{ij} = 1$ if median $j$ is selected and $x_{ij} = 0$, otherwise.

$$v(p) = \text{Min} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}$$  \hspace{1cm} (4.1)$$

Subject to

$$\sum_{j \in V} x_{ij} = 1 \quad i \in V$$  \hspace{1cm} (4.2)$$

$$\sum_{j \in V} x_{ij} = p$$  \hspace{1cm} (4.3)$$

$$\sum_{i \in V} q_i x_{ij} \leq Q_j x_{ij} \quad j \in V$$  \hspace{1cm} (4.4)$$

$$x_{ij} \in \{0,1\}; \ i \in V, j \in V$$  \hspace{1cm} (4.5)$$

Constraint (4.2) imposes that each demand point is allocated to only one median. Constraint (4.3) ensures that $p$ medians are selected. Constraint (4.4) imposes that a total median capacity must be respected, and (4.5) provides the integer conditions.

We can equate the RNCs in the UTRAN planning problem to the facilities (median) in the capacitated p-median problem, and the demand points to the Node Bs. With this assumption we can then formulate the UTRAN planning problem as a capacitated p-median problem.
4.4 Formulation of the UTRAN Planning Problem

The UTRAN topology as described in section 2.3 consists of a set of RNC rooted (multi-constrained) trees, where the constraints come from some technical limitations on the equipment. Due to the complexity of the design we make a few simplifications to the overall problem. These simplifications include the following: (i) A simple hierarchy with one level of concentration nodes (ii) no degree and/or depth constraints on the tree.

Let us consider a UTRAN service area covered by $n$ cells, where a set of $r$ RNCs must be allocated in order to manage the network and other resources. We also assume that the RNCs must be located in existing Node B sites in order to use the existing infrastructure, since it saves costs. It is always fulfilled that $r < n$ and in some cases $r \ll n$. Assume that a Node B serves a single cell and must be connected to a RNC by a link. The Node B sites and the RNC sites have fixed locations and are known. The capacity requirement of each Node B is known and may vary from one Node B to another.

The capacities of all the RNCs and the cost of linking each Node B to an RNC are also known. The main constraints of our design problem are firstly, we need to minimize the distance between the Node B and its associated RNC in order to maximize the reliability of the radio link between them. Secondly, a capacity constraint on the RNC since it can only manage a certain number of Node Bs.
Given the number and location of the Node Bs, then the design task can be divided into two subproblems:

- The optimal location of the RNCs.
- A Node B assignment pattern that minimizes a certain cost function for each of the RNCs selected.

The Node B – RNC assignment with a capacity constraint on the RNC represents a terminal assignment problem [65]. Terminal assignment (TA) is an important issue in telecommunication networks optimization to increase their capacity and reducing the cost of them. The objective of the TA problem is to minimize the link cost to form a network by connecting a given set of terminals to a given set of concentrators. The terminals and concentrators have fixed and known locations. Each concentrator is limited in the amount of traffic that it can accommodate. The capacities of all concentrators and the cost of linking each terminal to a concentrator are also known. The problem is to identify for each terminal the concentrator to which it should be assigned, respecting topological constraints, in order to minimize the total network cost.

In order to define an objective function for the RNC location problem, a TA must be solved in the calculation of the cost function for each RNC selection. For this particular solution we can associate Node Bs in the RNC location problem with terminals in the terminal assignment problem and RNCs in the RNC location problem with concentrators in the terminal assignment. The RNC location problem with the TA for evaluating the set
of Node Bs, is the well known capacitated p-median problem which is known to be NP-Hard [66].

4.4.1 Cost Function Calculation in the RNC location problem: the Terminal Assignment

The cost function used in the TA is adapted from the model used in [16]. This approach is based on a similar type of problem, the cell to switch assignment problem studied in [67]. The proposed model will consider the following assumptions:

- The network consists of \( n \) Node Bs, and \( r \) RNCs;
- The locations of Node Bs and RNCs are known;
- Each Node B is connected with one RNC;
- The connection costs between a Node B and a RNC are known. Such costs include both link and installation;
- The total capacities of the links connected to an RNC cannot exceed its capacity, in terms of circuits, bits per second, and maximum number of interfaces that can be installed on that equipment.

Let \( I = \{1, 2, \ldots, n\} \) be the set of Node Bs,

Let \( J = \{1, 2, \ldots, r\} \) be the set of RNCs;

Let \( c_{ij} \) denote the amortization cost associated to the link between Node Bi and RNC \( j \) \( (i = 1, \ldots, n; \ j = 1, \ldots, r) \).

Let \( INS_j \) express the monthly amortization cost for each installed RNC \( j \) \( (j = 1, \ldots, r) \).
Let \( x_{ij} \) illustrate a binary variable, which is equal to one if Node Bi is related to RNC \( j \); otherwise, \( x_{ij} \) is equal to zero.

The assignment of Node Bs to RNCs is subject to a number of constraints as follows:

The cost function \( Z(\bar{x}) \) of the assignment is expressed as follows:

\[
Z(\bar{x}) = \sum_{i=1}^{n} \sum_{j=1}^{r} c_{ij} x_{ij} + \sum_{j=1}^{r} \text{INS}_j
\]  

(4.6)

The first term of the equation represents the link cost, while the second term represents the amortization cost of the installed RNC. Let \( \lambda_i \) depict the number of calls per unit of time directed to Bi, and the vector \( q = [q_1, \ldots, q_r] \) of RNC capacities. The total load of all Node Bs assigned to RNC \( j \) must be less than the capacity \( q \) of the RNC.

\[
\sum_{i=1}^{n} \lambda_i x_{ij} \leq q_j, \quad \text{for} \quad j = 1, \ldots, r
\]  

(4.7)

Each Node B must be assigned to a single RNC

\[
\sum_{j=1}^{r} x_{ij} = 1, \quad \text{for} \quad i = 1, \ldots, n
\]  

(4.8)

\[
x_{ij} = 0 \text{ or } 1 \quad \text{for} \quad i = 1, \ldots, n \quad \text{and} \quad j = 1, \ldots, r
\]  

(4.9)

\( x_{ij} \) is 1 if Node Bi is related to an RNC \( j \), 0 otherwise.

The total network cost comprises of two components; the link cost between Node Bs and RNC, and the installation cost of the RNC. As a result the assignment problem consists of minimizing the cost function expressed in (4.6), subject to the constraints (4.7)-(4.9). When formulated
in this way, the assignment of Node Bs to RNCs appears to be very similar to the cell to switch assignment problem in 2G networks. This problem has been shown to be NP-Hard [67]. For this reason it is appropriate to use heuristics in order to find good solutions in reasonable computing time. Enumerative searches by contrast examine the entire search space in an exhaustive manner to find the optimal solution, and are more suited to small spaces, corresponding to small sized instances of a problem. For our network described above, consisting of \( n \) Node Bs and \( r \) RNCs, \( r^n \) solutions must be examined.

### 4.4.2 Modelling RNC Location Problem as Capacitated p Median

Let us assume that each of the \( N \) nodes in the network can act as a Node B or as an RNC. For any given solution to the RNC location problem there will be \( M \) nodes serving as RNCs and \( K = N - M \) nodes which are Node Bs. For this particular solution we can associate Node Bs in the RNC location problem with terminals in the terminal assignment problem and RNCs in the RNC location problem with concentrators in the terminal assignment. We have therefore that a given solution to the RNC location problem is an instance of the terminal assignment which can be solved using any algorithm for the terminal assignment existing in the literature. The solution for the terminal assignment has associated with it a cost function value given by (4.6). The RNC location problem then consists of finding the location of the RNCs into the nodes of the network which makes this cost function minimum.
By this definition, the RNC location problem is equivalent to a capacitated p-median problem[66] which has been addressed before in the literature.

The mathematical formulation of the RNC location problem as a capacitated p-median problem is as follows:

Let \( I = \{1, \ldots, N\} \) be the set of nodes in the network, and \( M \) be the number of nodes which will be selected as RNCs. Find a binary vector \( y \) such that:

\[
S = \{ j \text{ such that } y_j = 1 \}, \quad Z = \min \left( \sum_{i \in I - S} \sum_{j \in S} c_{ij} x_{ij} \right) \tag{4.10}
\]

Subject to

\[
\sum_{j \in S} \lambda_i x_{ij} \leq q_i y_i \quad j \in S \tag{4.11}
\]

\[
\sum_{j \in S} x_{ij} = 1 \quad i \in I - S \tag{4.12}
\]

\[
\sum_{j \in S} x_{ij} = M \tag{4.13}
\]

\[
y_i \in \{0,1\} \quad j \in I \tag{4.14}
\]

\[
x_{ij} \in \{0,1\} \quad i \in I - S, \quad j \in S \tag{4.15}
\]

Constraint (4.11) imposes that a total RNC capacity must be respected.

Constraint (4.12) imposes that each Node B is allocated to only one RNC.

Constraint (4.13) ensures that \( M \) RNCs are selected, and constraints (4.14)-(4.15) provide the integer conditions. Observe also that this definition includes the resolution of a terminal assignment problem for each value of vector \( y \).
4.5 Solution Approaches

Due to the difficulty of the problem described in Section 4.2, it is not possible to guarantee the optimal solution in reasonable running times. Therefore, techniques that give near optimal solutions within acceptable run times are needed. Two methods that find a sub-optimal solution without searching the entire solution space are genetic algorithms [60], simulated annealing (SA) [60], which approximates the solution of very large combinatorial optimization problems. Following the divided structure of the RNC location problem into two subproblems, our algorithms are based on a global–local search technique.

First, we use a choice of GA (cGA or ssGA) for choosing which nodes serve as RNCs; second, simulated annealing is used to solve the associated TA and obtaining a value of the cost function. We also explore various combinations of the GA and SA algorithms for solving the given problems. The considered network topologies are generated and optimized by using these combinations. For the purpose of this work cellular genetic algorithm (cGA) and steady state genetic algorithm (ssGA) were implemented. The implemented algorithms are modified versions of software developed at the Networking and Emerging Optimization group (NEO) at the University of Malaga[68]. The GAs work by encoding a population of non-binary strings, representing a possible selection of RNCs among the nodes which form the network. In this particular case the length of each individual representing a solution to the access network
design problem, is equal to the number of nodes in the network. This concept is explained in more detail later in the thesis.

4.5.1 Cellular Genetic Algorithms (cGA)

The cGA creates a random initial population, evaluates it, and then the variation operators are applied to all the individuals of the population in each generation. Figure 4.1 (figure from [69]) shows the neighborhood structure of the cellular GA. The idea of neighborhoods is explained further later in this section.

![Figure 4.1: Structure of a cellular GA](image)

The algorithm replaces the old population with the new one, evaluates this new population and then checks the stopping condition. The cGA makes this loop until the stopping condition is satisfied. Table 4.1 shows the pseudocode for the cGA used in the design. The algorithm runs until either it finds an optimal solution to the problem or it terminates after a certain number of generations (MAX_STEPS) without finding an optimal solution.
Table 4.1: Pseudocode for cellular GA

```
proc Rep_Cycle (ga)
    for s<-1 to MAX_STEPS do
        for x<-1 to WIDTH do
            for s<-1 to HEIGTH do
                list<-Calculate_neighbors(ga,position(x,y));
                parent1<- Select (list);
                parent2<- Select (list);
                Cross (ga.Pc,list[parent1],list[parent2],indiv_aux.chromosome);
                Mutate (ga.Pm,indiv_aux.chromosome);
                indiv_aux.fitness<- ga.Evaluate(Decode(indiv_aux.chromosome));
                Insert_New_Indiv (position(x,y),indiv_aux, [if better | if worse],
                    ga,population_aux);
            end_for
        end_for
    ga.population<-population.aux;
    Makes_Statistics (ga);  
    end_for
end_proc Rep_Cycle;
```

Encoding and variation Operators in cGA

**Selection**- this model only makes the selection on the neighbors. It generates a temporal population which replaces to the worst individuals in the old population. They can replace always or only if they are better. The population is represented as a two-dimensional (2-D) toroidal grid shape of size WIDTH x HEIGHT in which nodes are placed the individuals. A single individual exists at each grid point and interacts only with its neighbors on nearby grid points. There are two selection methods considered in the cGA; linear rank selection, and roulette wheel selection. Both of which are variants of local selection[70]. In linear rank selection for example, the neighborhood is sorted according to the objective values. The selection of an individual depends only on its position in the individuals rank and not on the actual objective value. The linear rank selection algorithm defines the Target Sampling Rate (TSR) of an individual x as:
\[ TSR(x) = \text{Min} + (\text{Max} - \text{Min}) \frac{\text{rank}(x)}{(N - 1)} \]

Where \( \text{rank}(x) \) is the index of \( x \) when the neighborhood is sorted in increasing order based on fitness, and \( N \) is the neighborhood size. The following constraints are also imposed:

\[ 0 \leq TSR(x), \quad \sum TSR(x) = N, \quad 1 \leq \text{Max} \leq 2, \quad \text{Min} + \text{Max} = 2 \]

The TSR is effectively the number of times an individual should be chosen as a parent for every \( N \) sampling operations.

**Reproduction** - reproduction allows us to obtain offspring from one or more parents. This involves two methods of recombination and mutation. Recombination is a reproduction operator which forms a new chromosome by combining parts of each of two parent chromosomes. This combination of parent chromosomes is usually done by selecting cross over points in each parent, splitting the chromosomes at these points, and linking those portions of chromosomes to create new ones. Table 4.2 illustrates the crossover operators in a binary string chromosome. The cGA model used includes a choice of Single Point Crossover (SPX), Arithmetic Crossover (AX), Uniform Crossover (UX), and Two Point Crossover (TPX).

<table>
<thead>
<tr>
<th>Table 4.2: Crossover operator in binary string chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Before Crossover</strong></td>
</tr>
<tr>
<td>0011</td>
</tr>
<tr>
<td>1110</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>After Crossover</strong></td>
</tr>
<tr>
<td>0011</td>
</tr>
<tr>
<td>1110</td>
</tr>
</tbody>
</table>
The mutation operator forms a new chromosome by making small alterations to the values of genes in a copy of a single, parent chromosome. The cGA model uses binary mutation which selects bits in the parent and inverts it with a given probability.

**Replacement** - replacement schemes are used by the cGA to determine how the new individuals will be assimilated into the population. There are a number of schemes that can be employed but only two (replace worst individual, crowding) are really useful here. The offspring replaces either the worst individual of the neighborhood or replaces the individual of the neighborhood with closest chromosome information. The cGA model used allows for a choice of methods.

**Neighborhood type** - The neighborhood of an individual is a set of individuals placed close to it on the grid, and whose shape has a great impact on the behaviour of the cGA. There are two neighborhood types represented in the cGA model; linear and compact. The linear shape is assigned to neighborhoods defined as all points reachable in ≤n steps taken in a fixed axial direction (North, East, West, and South) from the central grid point. The compact shape contains the closest n-1 points to the central grid point. All the examples shown in Table 4.3 are available for selection in the cGA model used.
4.5.2 Steady State Genetic algorithm (ssGA)

This algorithm generates a new individual in every step. The new individual is inserted in the population with his parents, and always replaces to the worst individual (or only replaces if it is better than the worst). This strategy is particularly useful when the representation of a solution is distributed on several individuals, possibly the entire population. The steady state replacement generates a population where the individuals are subject to large variations of lifespan measured in number of generations and thus in number of offspring. Table 4.4 shows the pseudocode for the implementation of the steady state algorithm.

Table 4.4: Pseudocode for ssGA

```plaintext
[ssGA] proc Rep_Cicle (ga)
    for s<-1 to MAX_STEPS do
        parent1<- Select (ga.population);
        parent2<- Select (ga.population);
        Cross (ga.Pc,parent1,parent2,indiv_aux.chromosome);
        Mutate (ga.Pm,indiv_aux.chromosome);
        indiv_aux.fitness<-ga.Evaluate(Decode(indiv_aux.chromosome));
        Insert_New_Indiv (ga,indiv_aux, [if better | if worse]);
        Makes_Statisticals (ga);
    end_for
end_proc Rep_Cicle;
```
4.5.3 Local Search using Simulated Annealing

This is a combinatory search guided method. This algorithm starts with an initial solution and attempts to locate the best solution in a finite number of possible solutions (neighborhood) of a problem. The current individual, which represents a partial solution, is modified to provide another individual with less cost. A neighborhood is simply a set of solutions that are found by applying an appropriate transformation (move) to the current solution.

The algorithm starts with an initial solution in which all the equipment is installed (i.e. all Node Bs and RNCs are installed). If this is a feasible solution we calculate its cost and store it as the current solution. In each iteration, the algorithm then explores the neighborhood of the current solution by choosing a random neighbour and calculates its cost, then decides whether it accepts this new solution or rejects it. This iteration is repeated until a certain stop condition is reached. The local search heuristic is presented in Table 4.5. The neighborhood is explored by making any of the following types of moves.

- Change the type of a node (a Node B or an RNC) that is already installed;
- Add a node (a Node B or an RNC) that reduces the most the cost function;
- Remove a node ((a Node B or an RNC) that is already installed.
Table 4.5: Pseudocode for SA

```plaintext
// Simulated Annealing
actual = new_structure();  // creates new structure
k = 0                      // initializes the counter
Tk = Calculate_Temperature(T0);
while Tk > T_FINAL do
    k = k + 1;
    new = Generate_structure(actual);  // generate neighbor of "actual"
    if Energy(new) > Energy(actual) then  // Energy is similar to fitness
        actual = new;
    else
        p = Calculate_Probability(k, Tk);
        if p > random() then actual = new;
        else refuse_structure(new);
    end_if;
end_if;
if k > L_MARKOV then
    k = 0;                            // update temperature
    Tk = Calculate_Temperature(Tk, T0);  //Calculate Tk=(T0* (e^ ( (c-1)*k) ) )
end_if;
end_do;
```

In using the SA algorithm to solve our design problem, only the best fitness of an individual is computed on each loop. SA is relatively easy to implement and also produces high quality solutions regardless of the choice of the initial configuration. SA transitions from an approximately random search process to a greedy, deterministic method controlled by the changing temperature parameter and are thus able to navigate complex, multimodal search spaces and avoid local-optima.

4.6 Implementation of the Algorithms

There are a number of steps that have to be determined as a pre-requisite for implementing any of the solution approaches outlined. These steps are now described for the different algorithms used.
4.6.1 cGA-based Terminal Assignment

The first step of the GA design involves choosing a representation for the problem. A simple notation was adopted to represent the encoding of Node Bs and RNCs based on the design used by Quintero and Pierre [16]. We use a non-binary string (chromosome) of length $T$, say $s_1, s_2, s_3, \ldots, s_T$, where the value of $s_i$ (gene) represents the RNC to which Node Bi is assigned. In this formulation the genes represent the Node Bs and the integer they contain represents the RNC to which the Node B of row $i$ is assigned as shown in figure 4.2.

![Figure 4.2: Chromosome representing an assignment of Node B-RNC](image)

The length of a chromosome is therefore equal to the number of Node Bs in the network $n$, and the maximal value of a gene represents the maximal number of RNCs $m$. In figure 4.2 for example, Node B 1 is assigned to RNC 2 and Node B 7 is assigned to RNC 3, and so on. This representation conforms to the implementation of a simple genetic algorithm whereby a chromosome is built on a one gene for one object basis. The length of a gene depends on the number of RNCs used in the network design. If for example we have a design where we use say 32 RNCs in the network, then the length of a gene would be 5, the number of bits required to represent 32. The length of a chromosome in bits therefore would be:

$$\text{CHROM_LENGTH} = \text{NUM_GENES} \times \text{GENE_LENGTH}$$
**Fitness function** - The most popular approach among genetic algorithm researchers to handle constraints is to use penalty functions. To evaluate the feasibility of solutions a penalty is incorporated into the cost function of equation 4.6 to cope with constraints. Instead of ignoring infeasible regions and concentrating only on feasible ones, unfeasibly bred strings are allowed to join the population, but at a price. For every infeasible string that joins the population, a penalty term incorporated into the cost function is activated. This reduces the infeasible strings strength relative to other strings in the population. The penalty function follows two main principles:

- The fitness functions use graded functions whereby two infeasible strings are not treated equally. The penalty is a function of the distance from feasibility[71].
- The best infeasible string cannot be better than the weakest feasible string.

The fitness function used for the assignment problem consists of two parts:

1. The cost function (4.6)

   \[ Z(\bar{x}) = \sum_{i=1}^{n} \sum_{j=1}^{r} c_{ij} x_{ij} + \sum_{j=1}^{r} INS_j \]

2. A penalty function used to penalise infeasible strings, where the weight of the penalty reflects the degree of violation of the RNC capacity constraint. A flowchart of the operation of the fitness function is shown in figure 4.3.
The penalty function consists of the sum of two parts:

- The first is the product of the number of Node Bs and the maximum distance on the grid. This will ensure that the best infeasible solution will always have a fitness value greater than the worst feasible solution.

- The second part is the product of the excess load on the RNC and the actual number of RNCs overloaded. This term is used to differentiate the degree of infeasibility among strings.

If we have a problem instance:

Node Bs: \( l_1, l_2, \ldots, l_n \)

RNCs: \( r_1, r_2, \ldots, r_m \)

And the Node Bs and RNCs are placed on a Euclidean grid, i.e. \( l_i \) has coordinates \( (l_{ij}, l_{ij}) \) and \( r_j \) has coordinates \( (r_{ij}, r_{ij}) \)
A feasible solution to the assignment problem is:

A vector $\vec{x} = x_1, x_2, \ldots, x_n$ where $x_i = j$ means that the $i$th Node B is assigned to RNC $j$ such that $1 \leq x_i \leq m$. And

$$c_{ik} = \text{round} \left( \sqrt{(l_{i,j} - r_{i,j})^2 + (l_{i,j} - r_{i,j})^2} \right) \text{ for } 1 \leq i \leq n$$

i.e. the cost of a link is the result of rounding the distance between the Node B $i$ and RNC $j$.

Although the associated link cost can be based on distance, delay, capacity, etc., in the numerical examples we use Euclidean distance.

4.6.2 cGA-based RNC location problem

A solution to this problem is represented similarly as in the cGA-based terminal assignment using integer vector $\vec{x} = x_1, x_2, \ldots, x_n$. The decision vector, $y$, is implicitly encoded. When evaluating a chromosome, $x$, we first find $y$ then if the node load is zero, it is not installed and its installation cost is not added to the objective function value. Otherwise its installation cost is added to the objective function value. This follows from our discussion in section 4.4.2 whereby the RNC location problem has an implicit terminal assignment. In our experiments we allow infeasible solutions in the population and penalise them similar to that of the terminal assignment described in section 4.4.1. The vector $y$ is then determined from the best solution at the termination of the algorithm.
CHAPTER 5: EXPERIMENTAL SETUP AND RESULTS

5.1 Introduction
This chapter presents the setup of our experiments and the results achieved with the applied genetic and simulated annealing algorithms. Comparisons are made with other existing solutions in the literature such as Quintero and Pierre’s hybrid genetic algorithm [16]. It is important to observe that in practical UTRAN planning applications the exact number and location of the RNCs is usually known in the beginning. This reduces the planning problem to that of an assignment of Node Bs to RNCs with constraints mainly on the capacity of the RNC. The need to find optimal location of RNCs generally exists when the network is being expanded as may be the case with future 3G networks. This is a considerably more difficult problem than the one we are addressing. Particular attention is paid to the case where the RNC locations are fixed and known.

5.2 Network Configuration
The physical location of the Node Bs and the RNCs are given on a Euclidean grid whose length and width are almost equal as shown in figure 5.1. It is assumed that sites (represented as circles) are distributed uniformly across the RNC area and carry roughly the same amount of traffic. The cost of a link between a node and an RNC is proportional to the distance that separates the two.
Problem instances are set up and can be called by the algorithms at execution. The data sets used comprise of problem instances taken from Salcedo-Sanz et al [72]. Following the setup used in [72], the experiments are split into two parts:

- The first part to test the performance of the heuristics used for the TA.
- The second part to test the performance of the heuristics used for the RNC location problem.

Several limitations on RNC capacity exist, and at least one of the following must be taken into account and the most demanding selected:

- Maximum number of cells [68]
• Maximum number of Node Bs under one RNC.
• Maximum Iub throughput.
• Amount and types of interfaces (e.g., STM-1, E1) [73].

I have chosen the maximum number of Node Bs under one RNC as the limiting factor on the RNC capacity. The number of RNCs needed according to the number of Node Bs to be connected can therefore be calculated as follows:

\[
\text{numRNCs} = \frac{\text{numNodeBs}}{\text{nbRNC} \times \text{fillrate}}
\] (5.1)

Where \(\text{numNodeBs}\) is the number of Node Bs in the area to be dimensioned; \(\text{nbRNC}\) is the maximum number of Node Bs that can be connected to one RNC; and \(\text{fillrate}\) is a margin used as a backoff from the maximum capacity. It is normal to assume a fillrate in the region of 90 percent as the maximum capacity is not usually achieved in practice.

### 5.3 Algorithm configuration

Defining a correct configuration file is the most important step for solving the optimization problems. The configuration file contains all information about the algorithms (cGA, ssGA, and SA) and the way in which the problem will be solved. The configuration file is divided in four parts:

• Main Parameters-- In this part there is information on the configuration parameters that affect all the different algorithms. For example we can choose which statistics (best fitness, worst fitness, etc) will be calculated during execution of the algorithms.
• Problem Parameters— Here we configure information specific to the type of problem that we are going to solve. We can for example indicate the maximum fitness value expected for the optimum solution, or set the length of a chromosome based on the chosen representation.

• Algorithm Parameters— This part defines the type of algorithm to use. We can select from simulated annealing or genetic algorithm. We then set parameters specific to the chosen algorithm, for example the crossover, mutation, and replacement operators used in genetic algorithm.

• Miscellaneous Parameters— Additional parameters that can alter the way the algorithms behave during execution. For example we can set a Boolean parameter that allows the probability of crossover and mutation rates to change dynamically during execution of the algorithm instead of letting them remain static throughout. We can also control the maximum start temperature and the minimum final temperature for simulated annealing.

The configuration file is shown below:

```
# MAIN PARAMETERS :

SYNC_PORT =1500       # Port to synchronize the island
NUM_ISLANDS =4         # Total number of islands
HOST_1 =127.0.0.1
HOST_2 =127.0.0.1
HOST_3 =127.0.0.1
HOST_4 =127.0.0.1
PORT_1 =6101
PORT_2 =6102
PORT_3 =6103
PORT_4 =6104
```
After the execution, the TA problem produces a data file that contains the Node B and RNC positions and the links. The data file is then converted to an image either using java or by plotting in Matlab. If using Java, in the
image the little green boxes indicates the RNCs and the overloaded RNCs
are represented by a little red box. Each Node B is represented by a blue
point. Finally, a black line between a RNC and Node B indicates the link.

5.4 Application Examples
This section is divided into two different parts. The first part is devoted to
test the performance of the proposed heuristics for the TA, presented in
Section 4.4.1. The second part will show the performance of the
algorithms for the RNC Location problem. All the algorithms used were
coded in java. A combination of java and Matlab was used to generate
plots of the results.

5.4.1 Experiments for the Terminal Assignment
To test the different algorithms for the terminal assignment we run a set of
experiments where their performance and computational time will be
evaluated. A 100 x 100 grid was set up, where a set of N= 100 Node Bs
and M= 5 RNCs will be placed. The algorithms are applied to ten different
collections of the 100 Node Bs to allow different cases in the distribution of
the Node Bs and RNCs. The data problem instances were taken from [74].
Each RNC has the same capacity value of 12, and the number of RNCs
for the ten problem instances remains at 5.

Table 5.1 shows an example of the format of the data file used to input the
ten collections of 100 Node Bs and 5 RNCs. This example is taken from
one of the distributions used in the experiments and shows only the first 20
Nodes in the network because of space limitations. The table shows the
coordinates of the Node Bs and RNCs, and their demands and capacities respectively. Note that the RNC coordinates chosen are identical to some of the Node B locations as it is a requirement to place the RNCs in existing Node B sites.

Table 5.1 demands and capacities for one instance of TA

<table>
<thead>
<tr>
<th>Node #</th>
<th>Node B x-coord</th>
<th>Node B y-coord</th>
<th>Node B demand</th>
<th>RNC x-coord</th>
<th>RNC y-coord</th>
<th>RNC capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>78</td>
<td>2</td>
<td>04</td>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>35</td>
<td>4</td>
<td>37</td>
<td>78</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>07</td>
<td>61</td>
<td>4</td>
<td>92</td>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>04</td>
<td>21</td>
<td>5</td>
<td>50</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>68</td>
<td>68</td>
<td>5</td>
<td>07</td>
<td>61</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>05</td>
<td>32</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>56</td>
<td>39</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>77</td>
<td>17</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>92</td>
<td>21</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>53</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>31</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>38</td>
<td>96</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>43</td>
<td>60</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>25</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>62</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>25</td>
<td>97</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>13</td>
<td>45</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>65</td>
<td>08</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>91</td>
<td>38</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>12</td>
<td>45</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2 shows the network configuration for the TA test problem for the distribution given in table 5.1 above. The full configuration of the number of Node Bs N=100 and the number of RNCs M=5 is shown as represented in Matlab. The Node Bs are represented by circles (o), while the crosses (+) represent the RNCs.
The demand of the Node B has been randomly chosen with values between 2 and 6. This was chosen because we are using the maximum number of Node Bs under one RNC as the capacity constraint on the RNC. The capacities of the RNC can then be calculated from the demands on the Node Bs. Finally the cost matrix has been computed as the Euclidean distance between each pair of Node B-RNC. It has to be noted that there is no specific mechanism in any of the algorithms when ties are encountered in computing the cost. This selection of Node B-RNC pair is done at random and may influence the chances of finding feasible solutions of quality. The computational time of the cGA and SA algorithms was also analyzed. This was done by varying the number of Node Bs (M) in the network from twenty five to one hundred and fifty. Ten experiments were done for each value of M, obtaining the execution time in seconds for
each algorithm. All the experiments were carried out on an Intel core i3 processor (2.13 GHz). The results are shown in figure 5.2 for the different algorithms tested.

![Graph showing CPU time/s in the TA simulations](image)

**Figure 5.3: CPU time/s in the TA simulations**

Each point along the curves was obtained by taking an average running time of the ten experimental runs for the different algorithms. It can be observed that the best results were obtained using the SA while the cGA took a considerably longer time. For the cGA we were able to find good solutions for relatively small instances of M. For larger instance of N=100 the algorithm terminated with mixed results, showing both feasible and infeasible solutions. Figure 5.4 shows an infeasible solution to the TA problem (shown earlier in figure 5.2) using the cGA algorithm. The RNCs are show in green while overloaded RNCs are shown in red. The total cost of the assignment is shown as well as the penalty sum for violating capacity constraint on the RNC. Figure 5.5 shows a feasible but suboptimal solution for the same test instance.
Figure 5.4 Infeasible solution to TA using cGA

Figure 5.5 Feasible solution to TA using cGA
The SA algorithm was more successful and generated better solutions for $M$ up to 100. Figure 5.5 shows the best solution found for the TA problem instance shown earlier in figure 5.2 using the SA algorithm. While this is a good feasible solution it does not appear from the fitness value to be close to the optimal solution.

![Figure 5.6: TA solution using simulated annealing](image)

The ssGA algorithm was also tested for solutions to the TA problem producing results not dissimilar to that of the cGA.

### 5.4.2 RNC Location using combined CGA and SA

In order to test the effectiveness of the cGA and SA in the RNC location problem we use problem instances generated as in [72]. There are six
RNC location instances, with different values for N and M. Two are small size networks (instances 1 and 2), two are medium size networks (instances 3 and 4) and the remaining two are large size networks (instances 5 and 6). Table 5.2 shows the main characteristics of the instances tackled. Following the configuration in [72], the demands of the Node Bs were randomly chosen between 1 and 6, and the capacities of the RNC varied between 15 and 22.

Table 5.2: Main characteristics of RNC location instances

<table>
<thead>
<tr>
<th>Instance #</th>
<th>Node Bs</th>
<th>RNCs</th>
<th>Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>3</td>
<td>100 x 100</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>4</td>
<td>100 x 100</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>6</td>
<td>100 x 100</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>8</td>
<td>200 x 200</td>
</tr>
<tr>
<td>5</td>
<td>80</td>
<td>10</td>
<td>200 x 200</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>12</td>
<td>200 x 200</td>
</tr>
</tbody>
</table>

The RNC locations are obtained by applying a clustering method. This algorithm assumes a minimum of 2 clusters and a maximum of p clusters where p is the number of nodes in the network. From the set of nodes the first RNC is chosen at random, the clustering algorithm then appoints the node farthest away from the first RNC as the second RNC. The algorithm then determines the coordinates of other RNCs based on a distance concept whereby all nodes are within j units of their hub or until all nodes are themselves RNCs. The parameters used for the cGA in simulations include a population size of 64 individuals, 200 generations, crossover
probability $P_c = 0.6$ and mutation probability $P_m = 0.01$. The cGA is used to locate the RNCs while the SA is used to perform the assignment as an implicit step in the RNC location. This arrangement is preferred because as we saw in section 5.4.1, the SA algorithm outperformed the cGA when applied to the TA problem. For smaller instances of the RNC location problem we were able to generate satisfactory results and plot them in Matlab.

Figure 5.6 shows the results obtained for instance # 2. In figure 5.6 (a) we see the 20 nodes for instance #2, distributed throughout the coverage area. 4 RNCs need to be located from amongst the 20 nodes, assuming that an RNC can be placed in any location. The remaining 16 (20-4) locations are therefore that of the Node Bs. In figure 5.6 (b) we see the locations chosen to place RNCs (represented by the +) and the grouping of the Node Bs around their chosen RNC (represented by the different colours).
Finally, figure 5.6 (c) shows the optimized solution showing the Node B to RNC assignment with interconnecting links. In the case of larger instances we were unable to produce any feasible solutions to the RNC location problem. For all the solutions found there were a number of RNCs overloaded.

Figure 5.7 Optimized solution for RNC location in instance #2
The fact that the SA algorithm performed reasonably well on its own for the terminal assignment, suggests that the combined approach with the cGA was not effective enough. One suggestion is that the SA could have been used as an operator within the cGA as opposed to executing the algorithm after the cGA had finished. Another consideration is the use of the penalty function in the cGA to handle constraints. If either the penalty is too large or too small, the problem could be very hard for GAs. A big penalty prevents searching the infeasible region, in which case the GA will converge to a feasible solution very quickly even if it is far from the optimal. A small penalty will cause to spend so much time in searching an infeasible region that the GA would converge to an infeasible solution.

5.5 Choice of optimization method

Compared to SA, GAs are widely considered to be superior due to their range of applications. At first glance, the SA approach is faster, as evidenced by our experimental results. The GA approach, on the other hand, is slower but ultimately leads to better performance; many local minima are present due to the tendency of other algorithms to become stuck in these local solutions. In particular, escaping from local optima is a very unlikely ability found in steepest descent methods.

Simulated Annealing is most effective when the state space to be searched is possibly small. Each state should also have lot of (meaningful) neighborhoods, allowing reaching the optimum in as few steps as possible. In addition, the cost of a state should be determined relatively
quickly. None of these criteria could be fulfilled easily in case of our planning problem. The most obvious idea for the state space would be the set of all feasible connections. When we vary connections between neighbouring edges by simply changing an edge, we violate the criteria that the state should have lots of meaningful neighborhoods, due to the topological constraints imposed by the UTRAN design problem. The state space is also very large when we are considering all connections in the calculation of the cost of a solution.

The main problem of the previous implementation of our algorithms was its complexity, because it produced large chromosomes despite the integer representation used. For instances where the number of Node Bs used was above 100, the chromosome length approached 1000 bits, that gave some high execution times and the solution was very difficult to find. A significant contributing factor was also the arbitrary placement of the nodes before the start of the experiments. A better representation for the chromosome is definitely needed in the case of the GA.

To achieve a minimum cost topology for the UTRAN planning problem, the design should be simplified and translated into an optimization routine with respect to the technological and economic constraints. In this chapter we presented a formulation designed to optimize this process. The selection of the heuristic algorithms for the optimization has an important impact on the final performance of the solution approach. The comparison of the different methods employed is difficult since they all have different
parameters to configure, and the final performance depends on the tuning
done. The optimization experiments show that the analysed heuristics
produce different performance depending on the running time.
SA achieved the best performance for the terminal assignment and
outperformed the cGA, which was expected. We were only able however
to find solutions with either heuristic for relatively small network sizes.

The results of the scenarios analyzed in this chapter highlight the
complexity of the design process, as the cellular operator needs to
estimate a number of parameters, but also shows the benefits of the
algorithmic approach, which can provide valuable help to the network
designer even in small network scenarios.
CHAPTER 6: CONCLUSION AND FURTHER WORK

6.1 Conclusion

This chapter explains the impact the UTRAN topology design problem has on the overall cost of a cellular network. We review the approaches taken to attempt to solve this problem and discuss improvements, changes and altogether different approaches that might be taken to obtain much better results. UMTS network planning and optimization is very complex and there is no general formulation of it. Complexity is generally overcome by defining simplifications which are modelled as mathematical programming problems. Due to the complexity of the design problem addressed in the thesis we had to make many simplifications and assumptions. This without question had a considerable impact on the eventual results.

Elements being considered as part of the evolution towards 4G include a packet-optimised radio access network with reduced latency, higher user data rates, improved system capacity and coverage, all at a reduced cost for the operator. In real-life networks the topology is governed by a number of factors: existing traffic distribution, estimated traffic growth, link distance, infrastructure cost, and so on. The hierarchical tree-/ring-based access network topology was well suited for voice-centric low-bandwidth services. The dynamics of proposed 4G wireless networks, with highly variable traffic characteristics and changing network requirements, make
such a solution rather inflexible and cost-ineffective. In this light, we visualize the hierarchical architecture of the RAN being completely replaced by a horizontal one, where Node Bs communicate with each other and play a more ‘intelligent role in backhauling traffic to the radio network controllers. The network will consist of many short multihop links, forming a mesh among the network elements, implying perhaps a completely new topology is needed for the UTRAN. Our research has explored the use of tree-like topologies to determine their suitability for future 4G networks. It can be argued that while the tree topology is the most cost effective, there are significant difficulties when it comes to introducing more radio nodes into the network for the purpose of expansion.

Cost optimization is itself a complex task in which the network planner has to carefully weigh the available alternatives to the mobile operator. The saving in transmission links for example must more than offset the investments in equipment with higher network capabilities and cannot be made at the expense of transmission network availability. These are conflicting interests and a delicate balancing of resources is needed to optimize the network. It is for this reason that many combinatorial optimization problems occur in telecommunications and a motivating factor in choosing the topic for research. One of the main conclusions that can be drawn from the work is that the performance of any iterative heuristic is closely related to its interaction with the problem in general and the
elements of the problem in particular. The algorithm parameters that can be closely related to the problem elements are the key entities in deciding the performance of the algorithm and the quality of solution. This was very apparent in the UTRAN design problem where even with the simplifications and assumptions made, was still very difficult to find a solution.

One example of this difficulty in identifying the right parameters for the problem at hand is in the RNC location problem. This was one of the key issues in the research and its location was dependent on different factors, such as transmission cost, and location of Node Bs. In our experiments the RNCs were distributed either near the Node Bs or centralized in a few major locations. Whether centralized or distributed, there was no established pattern that favoured one over the other. One can conclude that the RNC topology must be determined from case to case. The main reason for distributed RNC topology is to reduce the need for transmission resources. It is, however, not clear that distributed RNC topology automatically leads to reduced need of transmission resources.

6.2 Contributions

In this section the main contributions of this research in the area of UTRAN topology optimization is presented.

- This thesis presented various strategies for application of iterative heuristics in relevance to the UTRAN topology planning problem.
• The work undertaken has provided a basis for designing, modeling and implementation of commonly used heuristics (genetic algorithms, simulated annealing) in dealing with the UTRAN design problem.

• Presented a modification to the original problem by formulating the problem as a capacitated p-median problem. In doing so we may have inadvertently increased the complexity of the original design problem. This may have had a direct impact on some of the very modest results we obtained with our experiments.

• Compared the proposed iterative heuristic implementations in terms of their convergence, run times, and the quality of solutions produced.

6.3 Further work

In moving forward one of the key issues to address is that of finding better representations for various parameters of the network design. As an example the modelling of the traffic on the Iub link between Node B and RNC can be more realistic to better reflect a more realistic situation in a mobile network. At the same time handover costs which will play a significant part in future 4G networks as the system becomes more congested need to be considered in the calculation of the network cost. In our work we did not consider both hard and soft handover costs. In order to find a consistent set of call volume and handoff rates, we can adopt the widely used Jackson model for traffic handling [75].
In a Jackson network cells are treated as infinite server nodes, call originations as customer arrivals, handoff probabilities as node transition probabilities, and call holding times in each cell as the average service requirement of a customer at the corresponding node. This can then be solved to give the queue lengths, which translates into call volume for the cells, and the rate of customers moving between cells, which gave handoff rates. By extension we can then take the call volume for the cells and use the limiting factor on the RNC as the Iub throughput. As discussed in section 5.2 there are different limiting factors on RNC capacity; this approach gives us more flexibility in our design.

The issue of network reliability although briefly mentioned was not dealt with in sufficient detail in the thesis. In large networks such as UMTS with thousands of radio links, reliability will always be an issue, especially given the nature of the wireless environment. Associated with each type of connection is reliability, or equivalently, a stationary availability. This reliability has a range from 0 (never operational) to 1 (perfectly reliable). It is assumed that reliability comes at a cost. Therefore, a more reliable connection type implies a greater unit cost. The trade-off between cost and reliability is not linear. An increase in reliability causes a greater than equivalent increase in cost; often a quadratic relationship is assumed. As the UTRAN increases in size and complexity it is worth investigating the impact on reliability of the network. We could incorporate reliability calculations with relative ease into the UTRAN topology design problem. A
two phase approach can be taken for tree topologies where having obtained a minimum cost topology as the first phase; we extend our design to a second phase where we provide fault tolerance. The objective of the second phase would be to minimize the total cost of spare capacity for network restoration to satisfy survivability requirements. Due to the complexity of incremental topology design problems we can consider link restoration as the technique to guard against link failure.

In general, some more performance analysis can be done on this design approach, as it has produced modest results. In addition, other network topologies may be considered and integrated in the design as a future work, in which other topologies besides the tree may be configured.
REFERENCES


