Automated RRM optimization of LTE networks using statistical learning

Moazzam Islam Tiwana

Ecole Doctorale EDITE

Automated RRM Optimization of LTE networks using Statistical Learning

Thèse présentée pour l'obtention du diplôme de
Docteur de Télécom & Management SudParis

Doctorat délivré conjointement par
Télécom & Management SudParis et l'Université Pierre et Marie Curie - Paris 6

Spécialité :
Informatique, Electronique and Télécom

Par
Moazzam Islam TIWANA

Soutenue le 19 Novembre, 2010 devant le jury composé de :

Prof. Guy PUJOLLE	Président du jury
Prof. Tijani CHAHED	Directeur de thèse
Dr. Bruno TUFFIN	Rapporteur
Prof. Raquel BARCO	Rapporteur
Dr. Berna SAYRAC	Examinateur
Dr. Zwi ALTMAN	Examinateur
Dr. Adam OUOROU	Examinateur

Thèse n° 2010TELE0025

Dedication

To my Parents and Teachers

Acknowledgments

The research work presented in this report was carried out at Orange Labs (France Télécom) in collaboration with Télécom SudParis. I would like to thank all the persons who have helped me in the completion of my thesis. First of all, I would like to express my heartiest gratitude for Dr Berna Sayrac and Dr Zwi Altman, research engineers at Orange Labs, for supervising my thesis. Their great experience, vision and knowledge have always guided me in the right direction. Specially, they had a good idea of the practical feasibilty of various solutions that I proposed during the thesis meetings. This helped me a lot in broadening my horizon. It has been a great opportunity to work in Orange Labs because this company is equipped with lots of professional equipment, backed by the state of the art softwares that has really provided me with a working environment that is indeed rare to find.

I would also like to thank my acamedic supervisor Dr. Tijani Chahed, Assistant Prof. at Télécom SudParis, for his availability and technical guidance for the collaborative work on the thesis. I would like to pay my deepest thanks to all members of my team for their excellent company and technical help during the course of my work, especially Salah Eddine El Ayoubi, Frederic Morlot, Richard Combes and Ridha Nasri . I would also like to thank administrative personnel of the Orange Labs specially Bernadette Dubois, and team managers Arthuro Ortega-Molina and Laurent Marceron for helping me out with my administrative problems. Also, I would like to thank Raquel Barco and Bruno Tuffin for accepting to be the examinators of my thesis report.

Last but not the least a great thanks to my family for their affectionate, motivation and encouragement throughout my studies, whenever it was needed.

Résumé

Le secteur des télécommunications mobiles a connu une croissance très rapide dans un passé récent avec pour résultat d'importantes évolutions technologiques et architecturales des réseaux sans fil. L'expansion et l'hétérogénéité de ces réseaux ont engendré des coûts de fonctionnement de plus en plus importants.

Les dysfonctionnements typiques de ces réseaux ont souvent pour origines des pannes d'équipements ainsi que de mauvaises planifications et/ou configurations. Dans ce contexte, le dépannage automatisé des réseaux sans fil peut s'avérer d'une importance particulière visant à réduire les coûts opérationnels et à fournir une bonne qualité de service aux utilisateurs. Le dépannage automatisé des pannes survenant sur les réseaux sans fil peuvent ainsi conduire à une réduction du temps d'interruption de service pour les clients, permettant ainsi d'éviter l'orientation de ces derniers vers les opérateurs concurrents.

Le RAN (*Radio Access Network*) d'un réseau sans fil constitue sa plus grande partie. Par conséquent, le dépannage automatisé des réseaux d'accès radio des réseaux sans fil est très important. Ce dépannage comprend la détection détection des dysfonctionnements, l'identification des causes des pannes (diagnostic) et la proposition d'actions correctives (déploiement de la solution).

Tout d'abord, dans cette thèse, les travaux antérieurs liés au dépannage automatisé des réseaux sans-fil ont été explorés. Il s'avère que la détection et le diagnostic des incidents impactant les réseaux sans-fil ont déjà bien été étudiés dans les productions scientifiques traitant de ces sujets. Mais étonnamment, aucune référence significative sur des travaux de recherche liés aux résolutions automatisées des pannes des réseaux sans fil n'a été rapportée. Ainsi, l'objectif de cette thèse est de présenter mes travaux de recherche sur la " résolution automatisée des dysfonctionnements des réseaux sans fil LTE (*Long Term Evolution*) à partir d'une approche statistique ". Les dysfonctionnements liés aux paramètres RRM (*Radio Resource Management*) seront particulièrement étudiés.

Cette thèse décrit l'utilisation des données statistiques pour l'automatisation du processus de résolution des problèmes survenant sur les réseaux sans fil. Dans ce but, l'efficacité de l'approche statistique destinée à l'automatisation de la résolution des incidents liés aux paramètres RRM a été étudiée. Ce résultat est obtenu par la modélisation des relations fonctionnelles existantes entre les paramètres RRM et les indicateurs de performance ou KPI (*Key Performance Indicator*). Une architecture générique automatisée pour RRM

a été proposée. Cette dernière a été utilisée afin d'étudier l'utilisation de l'approche statistique dans le paramétrage automatique et le suivi des performances des réseaux sans fil.

L'utilisation de l'approche statistique dans la résolution automatique des dysfonctionnements des réseaux sans fil présente deux contraintes majeures. Premièrement, les mesures de KPI obtenues à partir du réseau peuvent contenir des erreurs qui peuvent partiellement masquer le comportement réel des indicateurs de performance. Deuxièmement, ces algorithmes automatisés sont itératifs. Ainsi, après chaque itération, la performance du réseau est généralement évaluée sur la durée d'une journée avec les nouveaux paramètres réseau implémentés. Les algorithmes itératifs devraient donc atteindre leurs objectifs de qualité de service dans un nombre minimum d'itérations. La méthodologie automatisée de diagnostic et de résolution développée dans cette thèse, basée sur la modélisation statistique, prend en compte ces deux difficultés. Ces algorithmes de la résolution automatisé nécessitent peu de calculs et convergent vers un petit nombre d'itérations ce qui permet leur implémentation à l'OMC (*Operation and Maintenace Center*).

La méthodologie a été appliquée à des cas pratiques sur réseau LTE dans le but de résoudre des problématiques liées à la mobilité et aux interférences. Il est ainsi apparu que l'objectif de correction de ces dysfonctionnements a été atteint au bout d'un petit nombre d'itérations. Un processus de résolution automatisé utilisant l'optimisation séquentielle des paramètres d'atténuation des interférences et de *packet scheduling* a également été étudié.

L'incorporation de la "connaissance a priori" dans le processus de résolution automatisé réduit d'avantage le nombre d'itérations nécessaires à l'automatisation du processus. En outre, le processus automatisé de résolution devient plus robuste, et donc, plus simple et plus pratique à mettre en oeuvre dans les réseaux sans fil.

Abstract

The mobile telecommunication industry has experienced a very rapid growth in the recent past. This has resulted in significant technological and architectural evolution in the wireless networks. The expansion and the heterogenity of these networks have made their operational cost more and more important. Typical faults in these networks may be related to equipment breakdown and inappropriate planning and configuration. In this context, automated troubleshooting in wireless networks receives a growing importance, aiming at reducing the operational cost and providing high-quality services for the end-users. Automated troubleshooting can reduce service breakdown time for the clients, resulting in the decrease in client switchover to competing network operators. The Radio Access Network (RAN) of a wireless network constitutes its biggest part. Hence, the automated troubleshooting of RAN of the wireless networks is very important.

The troubleshooting comprises the isolation of the faulty cells (fault detection), identifying the causes of the fault (fault diagnosis) and the proposal and deployement of the healing action (solution deployement). First of all, in this thesis, the previous work related to the troubleshooting of the wireless networks has been explored. It turns out that the fault detection and the diagnosis of wireless networks have been well studied in the scientific literature. Surprisingly, no significant references for the research work related to the automated healing of wireless networks have been reported. Thus, the aim of this thesis is to describe my research advances on "Automated healing of LTE wireless networks using statistical learning". We focus on the faults related to Radio Resource Management (RRM) parameters.

This thesis explores the use of statistical learning for the automated healing process. In this context, the effectiveness of statistical learning for automated RRM has been investigated. This is achieved by modeling the functional relationships between the RRM parameters and Key Performance Indicators (KPIs). A generic automated RRM architecture has been proposed. This generic architecture has been used to study the application of statistical learning approach to auto-tuning and performance monitoring of the wireless networks.

The use of statistical learning in the automated healing of wireless networks introduces two important difficulties: Firstly, the KPI measurements obtained from the network are noisy, hence this noise can partially mask the actual behaviour of KPIs. Secondly, these automated healing algorithms are iterative. After each iteration the network performance is typically evaluated

over the duration of a day with new network parameter settings. Hence, the iterative algorithms should achieve their QoS objective in a minimum number of iterations. Automated healing methodology developped in this thesis, based on statistical modeling, addresses these two issues. The automated healing algorithms developped are computationaly light and converge in a few number of iterations. This enables the implemenation of these algorithms in the Operation and Maintenance Center (OMC) in the off-line mode.

The automated healing methodolgy has been applied to 3G Long Term Evolution (LTE) use cases for healing the mobility and intereference mitigation parameter settings. It has been observed that our healing objective is achieved in a few number of iterations. An automated healing process using the sequential optimization of interference mitigation and packet scheduling parameters has also been investigated.

The incorporation of the *a priori* knowledge into the automated healing process, further reduces the number of iterations required for automated healing. Furthermore, the automated healing process becomes more robust, hence, more feasible and practical for the implementation in the wireless networks.

Contents

1 Introduction **23**
- 1.1 Background and problem definition . . . 23
- 1.2 Scope and objective of the thesis . . . 24
- 1.3 Original contribution . . . 25
- 1.4 Thesis structure . . . 26

2 Background **28**
- 2.1 Introduction . . . 28
- 2.2 State of the art in troubleshooting in RANs . . . 29
- 2.3 Statistical learning . . . 32
  - 2.3.1 Linear regression . . . 33
  - 2.3.2 Logistic regression . . . 35
- 2.4 Introduction to LTE . . . 36
- 2.5 Overview of LTE system . . . 37
  - 2.5.1 System requirements . . . 37
  - 2.5.2 System architecture . . . 38
  - 2.5.3 Physical layer . . . 39
  - 2.5.4 Self organizing network functionalities . . . 40
- 2.6 Interference in e-UTRAN system . . . 41
  - 2.6.1 System model and assumption . . . 41
  - 2.6.2 Interference model . . . 42
- 2.7 e-UTRAN handover algorithm . . . 44
- 2.8 Conclusion . . . 46

3 Statistical Learning for Automated RRM **47**
- 3.1 Introduction . . . 47
- 3.2 Statistical learning approach . . . 48
  - 3.2.1 Learning phase . . . 48
  - 3.2.2 Forward regression . . . 48
  - 3.2.3 Backward regression . . . 49
- 3.3 Regression Results: Two eNB scenario . . . 50

3.3.1 LTE network simulation scenario 50
3.3.2 Results 51
3.4 Automated RRM system description 59
3.5 Monitoring 59
3.6 Optimization model 61
3.7 Automated RRM: Description and results on LTE 63
3.8 Conclusion 70

4 Automated Healing by Statistical Learning 71
4.1 Introduction 71
4.2 Generic automated healing block diagram 72
4.3 Automated healing of LTE Mobility Parameters using Iterative Statistical Model Refinement 74
4.3.1 Relation to generic automated healing block diagram . 75
4.3.2 Automated healing algorithm description 75
4.3.3 Case study 77
4.4 Statistical Learning in Automated Healing(SLAH): Application to LTE interference mitigation 82
4.4.1 Relation to generic automated healing block diagram . 84
4.4.2 Application of SLAH for LTE intereference mitigation . 86
4.4.3 Case study 90
4.5 Enhancing RRM optimization using a priori knowledge for automated troubleshooting 104
4.5.1 Relation to generic automatd healing block diagram . . 105
4.5.2 A priori knowledge incorporation 105
4.5.3 Automated healing model 109
4.5.4 ICIC automated healing use case 109
4.6 Enhancement of the Statistical Learning Automated Healing (SLAH) technique using packet scheduling 116
4.6.1 Relation to generic automatd healing block diagram . . 118
4.6.2 α-fair scheduler 118
4.6.3 Automated healing algorithm 119
4.6.4 Simulations and results 121
4.7 Application of SLAH for LTE mobility 124
4.7.1 Adaptation of SLAH to mobility parameter 126
4.7.2 Simulations and results 128
4.8 Conclusion 133

5 Conclusion and Perspectives 135
5.1 Conclusion 135
5.2 Perspectives 136

Bibliography **138**

Author's Publications **148**

A LTE interference model **149**

B Pseudo code of the iterative KPI tuning algorithm **151**

List of Figures

2.1 Logistic function 35
2.2 LTE architecture 38
2.3 E-UTRAN (eNB) and EPC (MME and S-GW) 39
2.4 Inter-cell interference coordination scheme 42

3.1 Forward regression test for Handover Margin (HM). 49
3.2 Backward regression test for HM. 50
3.3 Scatter plot between HM and mean BCR of eNB_1, correlation coefficient=0.9. 51
3.4 Scatter plot between HM and Mean DCR of eNB1, Correlation coefficient=0.94713. 52
3.5 Scatter plot between HM and Mean Load of eNB1, Correlation coefficient=0.91135. 52
3.6 Scatter plot between HM and Mean Average Bit Rate of eNB1, Correlation coefficient=0.86923. 53
3.7 Scatter plot between HM and mean BCR of eNB_2, correlation coefficient=0.88. 54
3.8 Scatter plot between HM and Mean DCR of eNB2, Correlation coefficient=0.92941. 54
3.9 Scatter plot between HM and Mean Load of eNB2, Correlation coefficient=-0.9148. 55
3.10 Scatter plot between HM and Mean Average Bit Rate of eNB2, Correlation coefficient=0.84188. 55
3.11 5-fold cross validation error for forward regression. 56
3.12 Backward regression generalization errors. 57
3.13 3D Scatter plot between HM_{12} and mean BCRs of eNB_1 and eNB_2. 57
3.14 3D Scatter plot between HM12 and Mean Load of eNB1 and eNB2. 58
3.15 3D Scatter plot between HM_{12} and Mean ABR of eNB1 and eNB2. 58

3.16 3D Scatter plot between HM12 and Mean DCR of eNB1 and eNB2. 59
3.17 Block diagram for automated RRM. 60
3.18 3D scatter plot between HM_{12} and mean BCRs of eNB_1 and eNB_2, projected on the KPI plane. 60
3.19 3D scatter plot between HM_{12} and mean DCRs of eNB_1 and eNB_2, projected on the KPI plane. 61
3.20 3D scatter plot between HM_{12} and mean ABRs of eNB_1 and eNB_2, projected on the KPI plane. 61
3.21 3D scatter plot between HM_{12} and mean Loads of eNB_1 and eNB_2, projected on the KPI plane. 62
3.22 Admission probability as a function of the traffic intensity with (square) and without (triangle) autotuning. 66
3.23 Average Bit Rate as a function of the traffic intensity with (square) and without (triangle) autotuning. 66
3.24 HM histograms for different arrival rates (a) λ=3 (b) λ=4. . . 67
3.25 HM histograms for different arrival rates (a) λ=5 (b) λ=6. . . 68
3.26 HM histograms for different arrival rates (a) λ=7 (b) λ=8. . . 69

4.1 SLAH block diagram. 73
4.2 The network diagram of the simulated system. 77
4.3 The aggregated network KPIs (mean PPR, mean BCR and mean DCR) as a function of uniform (default) HM. 79
4.4 The KPIs in troublshooting as a function of HM_i after first optimization iteration. 80
4.5 The KPIs in troublshooting as a function of HM_i after second optimization iteration. 81
4.6 The KPIs in troublshooting as a function of HM_i after third optimization iteration. 81
4.7 Convergence of HM_i in automated healing algorithm for different KPI weights, ω_t. 82
4.8 The logistic function. 85
4.9 System Model. 87
4.10 The network diagram of the simulated system. 92
4.11 The variation of the global mean BCR of the network with uniform change in α of all eNBs. 93
4.12 The variation of the global mean FTT of the network with uniform change in α of all the eNBs. 93
4.13 Mean BCR values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ (γ=-0.3). 95

4.14 Mean FTT values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ (a) and for $eNB_{j=14}$, $eNB_{j=15}$, $eNB_{j=23}$ and $eNB_{j=45}$ (b) (γ=-0.3). 96
4.15 KPI of the eNBs in the optimization zone for the reference solution (white) and optimized (black) network conditions; mean BCR (a) and mean FTT (b) (γ=-0.3). 97
4.16 KPIs in descending order for the eNBs in the evaluation zone; mean BCR (a) and mean FTT (b) (γ=-0.3). 98
4.17 Mean BCR values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=15}$ and $eNB_{j=43}$ (γ=0). 100
4.18 Mean FTT values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=15}$ and $eNB_{j=43}$ (a) and for $eNB_{j=14}$, $eNB_{j=22}$, $eNB_{j=23}$ and $eNB_{j=45}$ (b) (γ=0). 101
4.19 KPI of the eNBs in the optimization zone for the reference solution (white) and optimized (black) network conditions; mean BCR (a) and mean FTT (b) (γ=0). 102
4.20 KPIs in descending order for the eNBs in the evaluation zone; mean BCR (a) and mean FTT (b) (γ=0). 103
4.21 *A priori* curves and data points for BCR as a function of $\alpha_{s=45}$. 112
4.22 *A priori* curves and data points for FTT as a function of $\alpha_{s=45}$. 112
4.23 Figure showing the convergence of $\alpha_{s=45}$ with and without *a priori* knowledge incorporated algorithm. 114
4.24 Mean BCR values of $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ along with the corresponding regression curves as a function of $\alpha_{s=45}$. 115
4.25 Mean BCR values of $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ along with the corresponding regression curves as a function of $\alpha_{s=45}$. 115
4.26 KPIs for the eNB in the evaluation zone in descending order for mean BCR (a) and mean FTT (b). 117
4.27 The network diagram of the simulated system. 122
4.28 Mean KPI values and LoR regression curves as a function of α_q for $eNB_{q=1}$, $eNB_{q=2}$, $eNB_{q=3}$ and $eNB_{q=15}$ (a) mean BCR (b) mean ABR. 125
4.29 Mean KPI values and LR regression curves as a function of HM_{sc} for $eNB_{j=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ (a) mean BCR (b) mean ABR. 131
4.30 KPI of the eNBs in the optimization zone for the reference solution (white) and optimized (black) network conditions; mean BCR (a) and mean ABR (b). 132

4.31 KPIs in descending order for the eNBs in the evaluation zone, mean BCR (a) and mean ABR (b). 133

List of Tables

3.1 The Percentage error of KPIs for the forwarde regression test . 56
3.2 Percentage improvement provided by the dynamic scheme. . . 70

4.1 The system level simulation parameters. 78
4.2 Convergence of HM_i during optimization process and the corresponding KPIs of automated healing algorithm 80
4.3 The complete SLAH Algorithm 91
4.4 Phase-I shows the initially chosen α values. Phase-II shows the α values calculated during optimization (γ=-0.3). 94
4.5 Phase-I shows the initially chosen α values. Phase-II shows the α values calculated during optimization (γ=0). 99
4.6 The complete SLAH Algorithm 110
4.7 The system level simulation parameters. 110
4.8 Table showing *a priori* data points 113
4.9 Shows the α values calculated during optimization. 113
4.10 System level simulation parameters 123
4.11 Phase-I shows the initially chosen α_q values for *optimization zone* . Phase-II shows the α_q values calculated during optimization. 124
4.12 The complete SLAH Algorithm 128
4.13 System level simulation parameters 129
4.14 Phase-I shows the initially chosen HM_{jc} values. Phase-II shows the HM_{jc} values calculated during optimization. 130

Acronyms

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first occurs in the text.

2G	Second Generation
3G	Third Generation
3GPP	Third Generation Partnership Project
ABR	Average Bitrate
ARQ	Automatic Repeat-request
BCR	Block Call Rate
BLER	Block Error Rate
bps	bits per second
CAC	Call Admission Control
CDMA	Code Division Multiple Access
DCR	Drop Call Rate
EPC	Evolved Packet Core
eNB	eNodeB
e-UTRAN	evolved UMTS Terrestial Radio Access Network
FD	Fault Detection
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FTT	File Transfer Time
GLM	Generalised Linear Model
GPRS	General Radio Packet Service
GSM	Global System for Mobile Communication
HM	Handover Margin
HO	Handover
ICIC	Inter-Cell Interference Coordination
IEEE	Institute of Electrical and Electronics Engineers
KPI	Key Performance Indicator
LHS	Left Hand Side
LiR	Linear Regression

LoR	Logistic Regression
LTE	Long Term Evolution
MAC	Medium Access Control
MBMS	Multimedia Broadcast/Multicast Service
MBSFN	Multicast-Broadcast Single-Frequency Network
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
MLR	Multiple Linear Regression
MME	Mobility Management Entity
MMF	Max-Min Fair
MTP	Maximum Throughput
NAS	Non-access Stratum
NMS	Network Management Station
NN	Neural Networks
NE	Network Element
OFDMA	Orthogonal Frequency Division Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
OMC	Operation and Maintenance Center
PBQ	Power Budget Quantity
PF	Proportional Fair
PPR	Ping Pong Rate
PRB	Physical Resource Block
QoS	Quality of Service
RAN	Radio Access Network
RHS	Right Hand Side
RNC	Radio Network Controller
SAE	System Architecture Evolution
S-GW	Serving Gateway
SOFM	Self Organizing Feature Map
SINR	Signal-to-Interference-plus-Noise Ratio
SLAH	Statistical Learning Automated Healing
SNR	Signal-to-Noise Ratio
SOM	Self Organizing Map
SON	Self Optimizing Network
TDMA	Time Division Multiple Access
TDD	Time Division Duplex
TS	Troubleshooting
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network

Chapter 1

Introduction

1.1 Background and problem definition

With the growth in the size and the complexity of the Radio Access Networks (RANs), the issue of the operational cost of the network is becoming more and more important. As networks increase quickly in size and complexity, it becomes impractical to manually examine all possible sources of fault in each network component. Automated troubleshooting is one of the solutions which can reduce the operational cost. Manual troubleshooting is time consuming, requires much expertise and involves high operational costs. Automated troubleshooting can be carried out by the software tools incorporated within the Operation and Maintenance Center (OMC).

Rapid troubleshooting of failures is a key challenge for telecom operators. The satisfaction of customers indeed requires to minimize the impact of failure events. Efficient troubleshooting processes with rapid fault detection, diagnosis and healing allow fast reaction to problems and the reduction of infrastructure equipment down-time and period of poor Quality of Service (QoS). Hence troubleshooting is clearly a strategic activity for an operator, and has a direct impact on QoS experienced by customers, on network churn reduction and on operator's revenues.

The network operational teams have to solve the performance degradations as quickly as possible. However, efficiently and effectively solving these problems imposes to cope with two challenges. The first one is to develop an automated, rapid and efficient fault management approach to problem solving in real-time even in high-speed networks with possibly a large number of impacted nodes. Moreover today's networks suffer from a wide and volatile set of failure types where the underlying fault proves difficult to detect, analyze and heal. The main reasons are the heterogenity of the underlying

technologies, protocols and terminal implementations, and the ever growing variety of applications. A second challenge is to define accurate but generic diagnosis and healing methods. They should avoid the use of too specific information related to the underlying protocols or wireless technologies that may be different among the different wireless networks.

Troubleshooting comprises the following three tasks: fault detection; cause diagnosis (i.e. identification of the problems' cause from Key Performance Indicators (KPIs) and alarms); and the solution deployment, namely fixing the problem or automated healing. A cause could be a hardware failure, like a broken base-band card in a node B, or a bad parameter value, i.e. transmission power, antenna tilt or a control parameter such as Radio Resource Management (RRM) parameter. The term symptom refers to quality indicators, i.e. quality of service (QoS) perceived by the user or performance indicator characterizing the network functioning and alarms. Alarms can be triggered when there is a material failure or when certain indicators exceed some thresholds. There are numerous parameters and possible hardware failures that could deteriorate the network performance and cause alarms. Furthermore, one fault could often trigger a series of alarms. To achieve the conclusive diagnosis and healing, not only alarms should be taken into account, but also performance indicators. The large number of possible faults, network KPIs and alarms make troubleshooting a complex task. In heterogeneous RANs which are more and more common in the wireless/mobile landscape, troubleshooting becomes even more complex. The automation of troubleshooting in general, will allow mobile operators to alleviate the burden of troubleshooting teams and to shorten the time necessary to identify and heal faults and thus reduce the time in which the network suffers from poor performance.

1.2 Scope and objective of the thesis

Fault management or troubleshooting is an important building block of network operation. Efficient troubleshooting processes allow us to rapidly react to problems, to lower infrastructure equipment down-time and to shorten period with poor QoS provisioning. Hence troubleshooting has direct impact on the operator revenues and on the rate of churn (users switching to competing network operators). Automated troubleshooting is an important functionality of Self Organizing Networks (SON) [1]. This thesis is related to the third step of the troubleshooting process i.e., auotmated healing. It has been assumed that fault has been correctly detected and diagnosed to be caused by a bad RRM parameter setting. First of all, the role of the sta-

tistical learning in the automated RRM is investigated in general. A generic architecture for automated RRM using statistical learning is proposed. The regression techniques are used to derive closed form relationships between the RRM and KPI parameters. These closed form relationships give us an estimate of the required change in an RRM parameter in order to achieve the required QoS objective. The design of auto-tuning heuristics using these relationships has also been investigated.

The automated healing algorithms developped in this thesis use these closed form relationships. They achieve the required QoS objective by selecting the most appropriate values for the network RRM parameters. These algorithms are iterative and for the practical operating wireless networks, the duration of one iteration is one day. Hence, the RRM optimization in this case is different from the standard optimization techniques as here the badly parameterised RRM parameters need to be optimized in a minimum number of iterations to achieve our QoS objective. The use cases of Long Term Evolution (LTE) network studied in this context investigate automated healing for interference mitigation, mobility and packet scheduling parameters.

1.3 Original contribution

This thesis has the following major contributions:

- the proposition of an automated RRM architecture based on statistical learning
- the design of an auto-tuning algorithm using inferences derived from statistical learning and its application to LTE mobility use case
- the development of statistical learning based automated healing scheme and its adaptation to the use cases of LTE mobility, interference mitigation and packet scheduling parameters

In the first contribution, a new statistical learning based automated RRM architecture has been proposed. This architecture and its utility for Self Organizing Network (SON) functionalities like network monitoring and network auto-tuning have been demonstrated in our work that appears in reference [2].

The results of the application of statistical learning to the LTE simualtion data have shown that the relationship between the KPIs and RRM parameters can be modeled as closed form expressions, denoted as the *model*. Using these results, an auto-tuning heuristic has been designed. This heuristic is then applied to mobility parameter of LTE networks to achieve network performance improvements.

A new iterative automated healing methodology based upon statistical learning has been proposed for LTE networks. The algorithm uses statistical *model*, obtained using statistical learning, whose accuracy improves during each optimization iteration. The algorithm has the advantage of converging within few iterations as shown in our work in reference [3].

The automated healing methodolgy presented has then been refined by making it more scalable as the automated healing of two eNBs that are not immediate neighbours can be done simultaneously. The methodology has been named as Statistical Learning Automated Healing (SLAH). The application of the SLAH methodology to the intereference mitigation parameters has been shown in our work in [4] and to the mobility parameters that appears in our work in [6]. The use of SLAH for the sequential automated healing of interference mitigation and packet scheduling parameters has also been addressed.

The SLAH methodogy has been improved for the operating networks by the incorporation of *a priori* knowledge into it. The initial *model* in SLAH methodology is very sensitive to noise in KPIs. The use of noise free *a priori* data ensures that this *model* is derived without noise. Furthermore, the number of the iterations required for the optimization has been further reduced as shown in reference [5].

1.4 Thesis structure

First, in Chapter 2, an extensive state of the art of the troubleshooting in mobile communications is presented. The survey of the troubleshooting is preceded by a brief presentation of the statistical learning techniques that is used in our work. A brief overview of the LTE technology that is used in the automated healing simulation scenarios, is also described.

Chapter 3, explains the statistical learning approach for automated RRM. A first case study of the statistical learning applied to two eNodeBs (eNBs) for a LTE simulation scenario is presented. A generic architecture for automated RRM is presented. The benefit of this automated RRM to the auto-tuning and monitoring of LTE network is demonstrated.

The statistical learning based automated healing is described in Chapter 4. A generic block diagram of this automated healing methodolgy is presented. The effectiveness of the automated methodology is demonstrated by its application to interference mitigation, mobility and packet scheduling parameters. The refinement and enhancement in this automated healing methodology for the use of *a priori* knowledge and packet scheduling is also addressed.

The last chapter summarizes and concludes the work presented. The highlights for the potential directions of the future work are discussed.

Chapter 2

Background

2.1 Introduction

Troubleshooting has been studied recently in the context of 2G and 3G networks. The main focus has been in the fault detection and fault diagnosis of these networks. As the complexity of wireless communication networks increases from day to day, the need for having troubleshooting as a part of automated network management has become critical. Research activity on this topic has been conducted both in industry and academia and has been reported in the literature. The objective of this chapter is to give a comprehensive survey on the research done in the area of troubleshooting of Radio Access Network (RAN), in wireless communications. This chapter covers the troubleshooting work on the technologies prior to LTE, namely 2G and 3G. The details of the statistical learning used in the current thesis are also given. Two types of statistical learning techniques based on: Linear Regression and Logistic Regression are described. This thesis work focuses on the automated healing aspect of the troubleshooting of wireless networks with application to LTE networks as an example. Hence, an introduction to key LTE technical details is also presented.

The structure of this chapter as follows: Section 2.2 explains the state of the art for the previous work on troubleshooting. The details of the statistical learning techniques used in our work are given in Section 2.3. The Section 2.3.1 and Section 2.3.2 describe the linear and logistic regression techniques, respectively. Section 2.4 gives an introduction to the system details of LTE system. Section 2.8 eventually concludes the chapter.

2.2 State of the art in troubleshooting in RANs

In the current wireless networks the troubleshooting tasks are achieved manually. The network experts observe the performance of the network and Network Elements (NE). Whenever there is a fault, the fault diagnosis strategy is based upon checking all possible causes and eliminating the possible causes one by one, untill we single out the actual cause. This procedure involves, not only querying the alarm, performance databases and network configuration, but also requires good knowledge of tools used for displaying information. The efficency of the troubleshooting process depends upon the expertise of the troubleshooting network experts.

First steps in automation of the troubleshooting process have been focused on performance visualization and Fault Detection (FD). Thanks to the methods that achieve efficient visualisation of the network performance, anomaly detection is more easily carried out.

Reference [7] proposes a method to measure the performance of the network in terms of the number of failed operations. This method uses the knowledge of KPI indicators, network architecture and quadratic programming for network analysis. The KPIs are divided into sets to describe the performance of different GSM network subsystems. Simple mathematical models are made for each subsystem based upon KPIs. The training data is used to estimate these models using quadratic programming. The interdependencies between different systems and KPIs are explored by constructing a graph.

Fault Detection (FD) is the first step of troubleshooting. Many studies have been done in this field. These methods are based upon building a behavioural model of an operating network under normal conditions. With the availability of such a model, any deviation from the normal network operating behaviour can be detected as an anomaly. In [8]- [11] a FD method is proposed which uses the Self-Organizing Map (SOM) as its basic principle. A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network that is trained using machine learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. In wireless networks, a behavioural pattern of a cell is a set of KPIs. Hence in network performance analysis, SOM of different cells can be used to find the behavioural similarities between them. The cells having similar behaviour will be located close to one another. In statistics and machine learning, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. A combination of SOM and K-means clustering algorithm can be

used to isolate the the cell clusters with similar behaviours. The behavioural degradation/fault in a cell can be found by calculating its distance with cell cluster having a normal profile.

A neural network algorithm, namely Winner-Take-All, has been used for FD in [12] [13]. In this algorithm, the weight of a single neuron is only changed with each new input vector. The network is trained using KPI vectors collected during the normal network operations. The percentiles of the normality profile that represents a numerical interval to represent the normal behaviour of the system is defined. FD is carried out in the following way: the defined interval for normal behaviour is used to classify a new vector into normal/abnormal by means of hypothesis testing. The value of each new KPI is evaluated and if it does not lie within the defined range of normal behaviour interval then it is termed as abnormal. Once a new KPI has been identified as normal/abnormal, this decision can serve as input to a more complex diagnosis system.

The diagnosis of the problem cause is the second step of the troubleshooting process. There are few references about diagnosis in the RAN of cellular networks. The automated diagnosis has been studied extensively in other fields such as diagnosis of diseases in medicine [14]- [21], troubleshooting of printer failures [22]- [31], diagnosis of faults in satellite communication systems [32]- [35], etc. The identification of the faults in the communication networks has been studied in [36]- [40]. In these cases, the interdependence among various communication network modules plays an important role in fault identification, as failure in one network module may result in supplementary alarms in other network modules. Alarms are the only symptoms of the fault in these scenarios. However, in the case in RANs, the above formulation is not valid. As most often in the case of RANs, the fault is not related to the equipment or module failure but rather to poor network configuration and planning. Furthermore, the alarm information in this case is important but not conclusive, for example, for problems like excessive interference or bad coverage. Hence, the information of KPIs is also required and unlike alarm information, the KPI values are continous. Thus the mathematical modeling for cause diagnosis in RANs is different from the modeling of other communication networks as in the case of RAN each subsytstem has its unique functionalities.

The alarm correlation has been used in [41]- [46] for the diagnosis in the RANs. Alarm correlation [45] [46] consists of intrepreting the conceptual significance of multiple alarm occurences that are generated from one fault. The underlining mechanism for the fault diagnosis involves the reduction of multiple alarm into a single alarm and inhibition of the low priority alarms. When there is a fault in the RAN, it may result in generation of multiple

alarms. Alarm correlation systems reduce multiple alarms into a single and meaningful alarm. Hence, the fault identification process becomes simpler for the operator. For example, when a link fails, up to 100 or more alarms are generated and passed to the Network Management System (NMS). Those alarms should be converted into a minimum number of alarms which clearly pinpoint to the breakdown of a link. In [42]- [44], model-based alarm correlation has been presented. Here, the behaviour of a network device is modelled as a set of formulas. For example, if a certain set of alarms related to a network device are present, it is considered to be faulty. Neural networks are used in [41] [43] as the basis of alarm correlation methodology. Here, the network generated alarms are used as the neural network input layer, while alarms filtered by alarm correlation are represented as the output layer. The weights of the neural network are adapted using the training data.

Despite the fact that the alarm correlation can be used in fault diagnosis, the alarm correlation cannot provide the conclusive diagnosis for the alarms caused by intereference or lack of coverage. In order to diagnose causes of such alarms, additional information about the KPIs needs to be taken into account.

An automatic diagnosis method for RANs which utilizes KPI information has been proposed in [47]. This method uses the Naive Bayesian classifier, using the same reasoning as in [48], for the diagnosis in GSM/GPRS, 3G or multi-systems networks where a diagnosis in GSM/GPRS networks has been proposed. The beta density function models the relations among the symptoms and the causes. This model is tested on data from a live GSM network.

In [49]- [52], Bayesian Networks (BN) using the same reasoning as in [53] [55] [54], are used for automatic diagnosis in cellular networks. Here, the performance indicators are modeled as discrete variables. The trials are done on the data from live GSM/GPRS network. The question of choosing a suitable BN network has been investigated in [56]. The paper in [57] proposes two methods for improving the accuracy of the troubleshooting model. The diagnosis methodology based upon the BN is extended to the diagnosis of UMTS networks in [58] and [59].

Reference [60] a Bayesian Network based diagnosis is proposed that takes into account both alarm correlation and performance data. The proposed method divides the 'hardware' and 'transmission' related causes into several sub causes in order to provide details about problem's location (BTS's subsystem: power unit, transmission unit, signalling or BTS itself).

Reference [61] compares the performance of Causation Bayesian Network and Naive Bayesian Network for wireless network fault diagnosis. It is observed that both types of BNs learns from the same training data, but the

latter has better performance. After the review of the scientific literature it turns out that the above mentioned techniques of fault detection and diagnosis are yet to be applied to the LTE network as it is in the early phases of its deployment.

There does not seem to be a significant work or publications related to the third step of the troubleshooting process i.e., problem solution or automated healing. Hence, in this thesis, the automated healing of wireless networks, particularly LTE networks, has been explored.

2.3 Statistical learning

The main goal of statistical learning theory is to provide a framework for studying the problem of inference, that is of gaining knowledge, making predictions, making decisions or constructing models from a set of data. This is studied in a statistical framework, that is there are assumptions about the statistical nature of the underlying phenomena (in the way the data is generated). Statistical learning offers efficient algorithms that learn from observations. Such algorithms are usually referred to as machine learning algorithms. The aim is to perform a prediction or an estimation regarding a (usually complex) system. To achieve this goal, we dispose two kinds of information: a priori and/or a posteriori. A priori information is usually in the form of some hypotheses or models that synthesizes our past knowledge on the problem at hand. A posteriori information consists of the data that we obtain while observing the system of interest. A machine learning algorithm takes the models/hypotheses and the observations to learn about the nature of the problem and the dynamics of the system. As a result of this learning, it comes up with a set of rules that lead to predictions/estimations. Since the observations are limited, these predictions/estimations are not deterministic but rather probabilistic, i.e. they contain probabilistic bounds or reliability intervals. The problem at hand, i.e. finding RRM parameters that yield a desired/optimum KPI performance, fits perfectly into this framework. The a priori knowledge is provided by the models (propagation model, traffic model, mobility model, network model etc.) used by analytic and/or numeric evaluation methods; and the a posteriori knowledge is provided by network measurements. Thus, it is absolutely plausible to use the statistical learning framework for the auto-correction of RRM parameters.

2.3.1 Linear regression

The statistical learning technique that is chosen to be applied on our problem is the *regression*. In regression, the relation between a response variable (dependent variable) and one or more explanatory variables (independent variables) is sought. The response variable is formulated as a function of the explanatory variable(s), corresponding *coefficients* (constants) and an error term. In the present case, the regression will relate an RRM parameter to a set of KPIs. The aim is to estimate the coefficients that yield a best fit, generally in the least square sense. The error term explains the unpredictable part in the estimation. Regression is widely used in statistical learning. In our problem, it is preferable to work with more than one KPI for a satisfying RRM parameter optimisation. Furthermore, depending upon the nature of relationship between RRM parameters and the KPIs, linear regression model is one of the models used in the regression analysis. This brings us to a *linear regression* model that can be formulated as follows: Let y_i denote the i_{th} sample of the response variable $\tilde{y}$, x_i the i^{th} sample of the explanatory variable $\tilde{x}$, β_k the k^{th} coefficient and ϵ_i the error in the ith sample. Then, we can write the linear regression expression for the i^{th} sample as follows:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \qquad i = 1, 2...I \qquad k = 0, 1 \tag{2.1}$$

Considering all of the samples $i = 1, 2....I$, where I is the total number of training samples, the linear regression equation takes the below form:

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_I \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_I \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \end{bmatrix} \tag{2.2}$$

which can also be written using the vector-matrix notation:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{2.3}$$

The criterion used to find the best fit (i.e. the coefficients) is generally the *least squares* criterion where the sum of the squares of the Euclidean distance between each sample and its estimate $\hat{y}_i = \beta_0 + \beta_1 x_i$ is minimized:

$$\beta^* = argmin_\beta \sum_{i=1}^{I} (y_i - \hat{y}_i)^2 \tag{2.4}$$

$\epsilon = y_i - \hat{y}_i$ is the i^{th} error term and empirical variance of the error (also know as the mean squared error) can be calculated as:

$$\sigma^2 = \frac{\sum_{i=1}^{I} \epsilon_i^2}{I - 1} \tag{2.5}$$

σ^2 is one of the metrics to judge the quality of the regression estimation. Therefore, the least squares estimates are given as

$$\beta^* = (X'X)^{-1}X'y \tag{2.6}$$

2.3.1.1 Coefficient of Determination (R^2)

Coefficient of determination is often used to judge the adequacy of a regression model. It is the proportion of variability in a data set that is accounted for by the statistical model.
The "variability" of the data set is measured through different sums of squares. The total sum of squares (proportional to the sample variance) is given as;

$$SS_{tot} = \sum_{i=1}^{I} (y_i - \overline{y})^2$$

where $\overline{y}$ is mean of observed values y_i i.e., $\overline{y} = \frac{\sum_{i=1}^{I} y_i}{I}$. The regression sum of squares, also called the explained sum of squares is given as:

$$SS_{reg} = \sum_{i=1}^{I} (\hat{y}_i - \overline{\hat{y}})^2$$

where $\overline{\hat{y}}$ is mean of predicted values $\hat{y}_i$ i.e., $\overline{\hat{y}} = \frac{\sum_{i=1}^{I} \hat{y}_i}{I}$. The sum of squares of residuals, also called the residual sum of squares is given as:

$$SS_{err} = \sum_{i=1}^{I} (y_i - \hat{y}_i)^2$$

The most general definition of the coefficient of determination is:

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}} \tag{2.7}$$

The statistic R^2 should be used with caution, because it is always possible to make R^2 unity by simply adding enough terms to the model. For example, we can obtain a perfect fit to I data points with a polynomial of degree $I-1$. In addition, R^2 will always increase if we add a variable to the model, but this does not necessarily imply that the new model is superior to the old one. R^2 does not measure the appropriateness of the model, since it can be artificially inflated by adding higher order polynomial terms in x to the model. Finally, even though R^2 is large, this does not necessarily imply that the

regression model will provide accurate predictions of future observations. Many regression users prefer to use adjusted R^2 statistics:

$$R^2_{adj} = 1 - \frac{SS_{err}/\left(I-2\right)}{SS_{tot}\left(I-1\right)} \tag{2.8}$$

As $SS_E/\left(I-2\right)$ is the error or residual mean square and $SS_{tot}/\left(I-1\right)$ is a constant, the R^2_{adj} will only increase when a variable is added to the model if the new variable reduces the error mean square.

2.3.2 Logistic regression

The second statistical learning technique used in our automated-healing work is the Logistic Regression (LoR) [91] [87]. The LoR establishes the statistical model by extracting the functional relations between the KPIs and the RRM parameters. It fits the data into a functional form denoted as *logistic function*

$$f(z) = \frac{1}{1 + exp^{-z}} \tag{2.9}$$

where z can vary from $-\infty$ to ∞ and $f(z)$ from 0 to 1 (see Figure 2.1). One can see from Figure 2.1 that $f(z)$ has the advantage of describing the saturation effects in its extremities as often encountered in KPIs in a communication network.

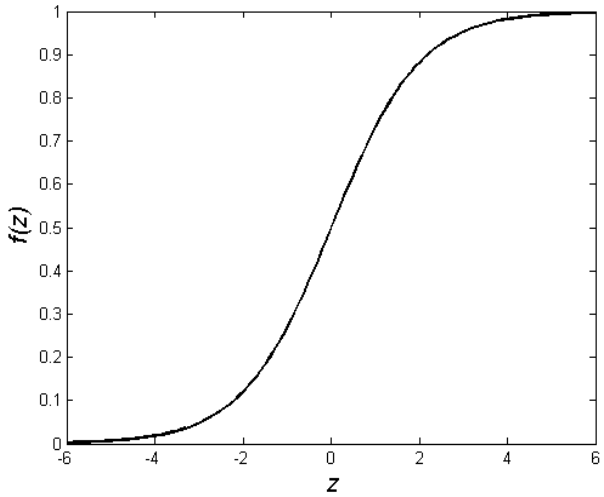


Figure 2.1: Logistic function

The LoR model is defined as follows. Denote by Y_1,Y_2,...,Y_n independent variables following a Binomial distribution. Let p_i be the mean value of Y_i, namely $p_i = \mathbb{E}[Y_i] = P(Y_i = 1)$. p_i can be expressed in terms of the explanatory variable x_i via the logistic function as

$$p_i = f(z_i) \tag{2.10}$$

where z_i being a variable representing the contribution of all explanatory variables, $z_i = \beta_0 + \beta_1 x_i$; β_0 is the intercept and β_1 is the regression coefficient of the explanatory variable x_i. The output of the logistic function is limited from 0 to 1. So, in our work, each time we apply the LoR on a particular type of KPI, the KPI values are normalised with their maximum value as normalising factor. After the system model estimation, the estimated KPI values obtained from the model are denormalised. Here $p_i = \mathbb{E}[Y_i]$ represents the mean normalized KPI value in the eNB.
The application of the logit transformation to (2.10) provides a linear relationship between logit(p_i) and the explanatory variables

$$\text{logit}(p_i) = log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i \tag{2.11}$$

The β_j coefficients in (2.11) can be estimated using the maximum likelihood method [88].

2.4 Introduction to LTE

3GPP (3rd Generation Partnership Project) organization has defined the requirements for an evolved UMTS Terrestrial Radio Access Network (e-UTRAN) [62]. The evolution of 3G UTRAN is referred to as the 3GPP Long Term Evolution (LTE). In 2006 3GPP started to standardise LTE in Release 8, with the following disruptive changes compared to the previous releases: Orthogonal Frequency Division Multiplex Access (OFDMA) instead of Wide Band Code Division Multiplex (CDMA) flat architecture instead of a hierarchical architecture is proposed. Release 8 standardisation has been completed in march 2008. However, since march 2008 the specifications is being corrected continuously.

Different working groups are involved in defining the architecture and the technology of the radio access and the core network [64]. In the framework of the working group 3GPP TSG-RAN WG3, there have been discussions and studies on the use of self-configuration, self-tuning and self-optimization in the e-UTRAN system [65]. In the first phase of the network optimization/adaptation, neighbour cell list optimization and coverage and capacity

control have been proposed [65].
The purpose of this section is to present an essential overview of LTE technology.

2.5 Overview of LTE system

The objective of the LTE is to introduce a new mobile-communication system that will meet the needs and challenges of the mobile communication industry in the coming decade [68] [67]. It is characterized by a flat architecture; a new radio access technology with an OFDM (Orthogonal Frequency Division Multiplexing) based physical layer; and considerably enhanced performance with respect to current 3G networks, including delays, high data rates and spectrum flexibility. The LTE technology is specified by 3GPP and is developed in parallel with the evolved HSPA. Unlike the evolved HSPA comprises of a smooth evolution of 3G networks, LTE is fully based on packet switched transmissions with IP based protocols and will not support circuit switched transmissions. The LTE radio access can be deployed in both paired and unpaired spectrum, namely it will support both frequency- and time-division based duplex arrangements. In Frequency Division Duplex (FDD) downlink and uplink transmission are carried out on well separated frequency bands whereas in Time Division Duplex (TDD) downlink and uplink transmissions take place in different non-overlapping time slots. A special attention is given in LTE to efficient multicast and broadcast transmission capabilities. This transmission is denoted as the Multicast-Broadcast Single-Frequency Network (MBSFN).

2.5.1 System requirements

3GPP has defined ambitious performance targets for the LTE system, and the important ones are summarized below [62] [67]. At the base station, one transmit and two receive antennas are assumed and at the mobile terminal side, one transmit and maximum two receive antennas are assumed.

- Peak data rate of 100 Mbit/s and 50 Mbit/s in downlink and uplink transmissions respectively in a 20 MHz bandwidth.
- Improvement of mean user throughput with respect to HSPA Release 6: 3-4 times in downlink; 2-3 times in uplink; and 2-3 times in cell-edge throughput measured at the 5th percentile.

- Significantly improved spectrum efficiency: 2-4 times that of Release 6, achieved for low mobility, between 0 to 15 km/h, but should remain high for 120 km/h, and should still work at 350 km/h.
- Significant reduction of user and control plane latency with a target of less than 10 ms user plane round-trip time and less than 100 ms for channel setup delay.
- Spectrum flexibility and scalability, allowing to deploy LTE in different spectrum allocations: 1.25, 1.6, 2.5, 5, 10, 15 and 20 MHz.
- Enhanced Multimedia Broadcast/Multicast Service (MBMS) operation.

2.5.2 System architecture

The requirements of reducing latency and cost have led to the design of simplified network architecture, with a reduced number of nodes. The RAN has been considerably simplified. Most functions of the RNC in UMTS have been transferred in the LTE to the eNodeBs (eNB) that constitute now the RAN part, and denoted as the e-UTRAN. The e-UTRAN consists of eNBs interconnected with each other by means of the X2 interface (see Figure 2.2). The eNBs are also connected by means of the S1 interface to the Evolved

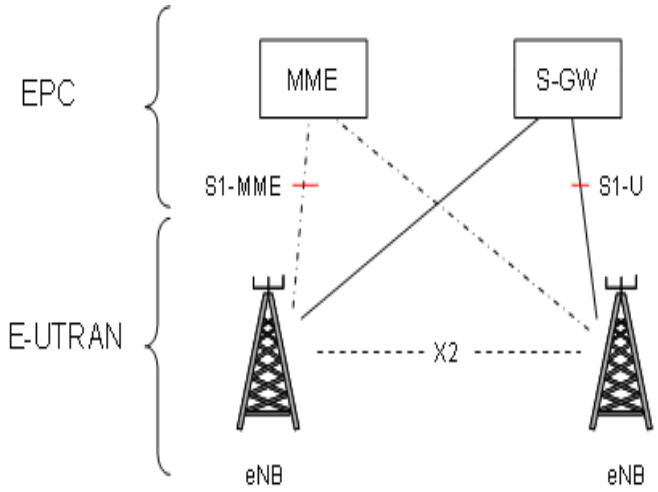


Figure 2.2: LTE architecture

Packet Core (EPC), and more specifically, to the Mobility Management Entity (MME) via the S1-MME interface, and to the Serving Gateway (S-GW)

via the S1-U interface. The S1 interface supports a many-to-many relation between MMEs / Serving Gateways and eNBs. Among the functions of the eNBs are RRM functions, such as radio admission control, radio bearer control, connection mobility control, dynamic resource allocation (scheduling) to the User Equipment (UE) in both uplink and downlink; IP header compression and encryption of user data stream; routing of user data towards the Serving Gateway (S-GW); scheduling and transmission of paging messages; and scheduling and transmission of broadcast information [63]. The MME is responsible for the following functions: distribution of paging messages to the eNBs; security control; idle state mobility control; SAE bearer control; and ciphering and integrity protection of Non-Access Stratum (NAS) signalling. The term SAE, or System Architecture Evolution has been given by 3GPP to the evolution of the core network, and was finally denoted as the EPC. The Serving Gateway is the mobility anchor point. The different functions of the eNB, MME and the S-GW are depicted in Figure 2.3.

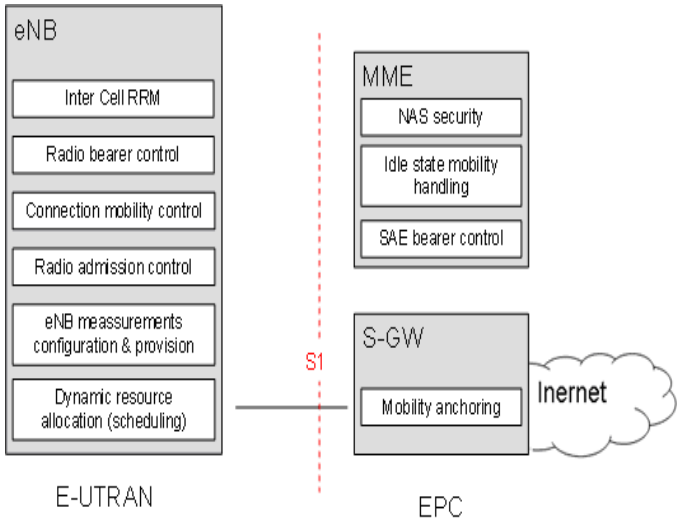


Figure 2.3: E-UTRAN (eNB) and EPC (MME and S-GW)

2.5.3 Physical layer

LTE uses OFDMA (Orthogonal Frequency Division Multiple Access) as the downlink transmission scheme [64] [69]. OFDMA uses a relatively large number of narrowband subcarriers, tightly packed in the frequency domain. The subcarriers are orthogonal, hence without mutual interference. The OFDMA scheme can be rendered robust to time-dispersive channel by the cyclic-prefix

insertion, namely the last part of the OFDM symbol is copied and inserted at the beginning of the OFDM symbol. Subcarrier orthogonality is preserved as long as the time dispersion is shorter than the cyclic-prefix length. To achieve frequency diversity, channel coding is used, namely each bit of information is spread over several code bits. The coded bits are then mapped via modulation symbols to a set of OFDM subcarriers that are well distributed over the overall transmission bandwidth of the OFDM signal [70]. In the uplink, LTE uses the Single-Carrier FDMA (SC-FDMA: Frequency Division Multiple Access) transmission scheme. This scheme can be implemented using a DFTS-OFDM, namely an OFDM modulation preceded by a DFT (Discrete Fourier Transform) operation. It allows flexible bandwidth assignment and orthogonal multiple-access in the time and frequency domains.
The OFDM transmission scheme allows dynamically sharing time-frequency resources between users. The scheduler controls at each instant to which user to allocate the shared resources. It can take into account channel conditions in time and frequency to best allocate resources. According to channel variations, in addition to choosing the mobiles to be served, the scheduler determines the data rate to be attributed to each link by choosing the appropriate modulation. Hence rate adaptation can be seen as part of the scheduler. In the downlink, the smallest assignment resolution of the scheduler is 180 kHz during 1 ms which is called a resource block. Any combination of resource blocks in a 1 ms interval can be assigned to a user. In uplink, for every 1 ms, a scheduling decision is taken in which mobile terminals are allowed to transmit during a given time interval, on a contiguous frequency region, with a given attributed data rate. Scheduling in LTE is a key element to enhance network capacity.
To enhance the RAN performance, fast hybrid ARQ (Automatic Repeat-reQuest) with soft combining is used to allow the terminal to rapidly request retransmissions of erroneous transport blocks [64]. From the first release, LTE supports multiple antennas in both eNB and the mobile terminal. Multiple antennas are among the features that allow the LTE to achieve its ambitious targeted performances, including multiple receive antennas, multiple transmit antennas, and MIMO (Multiple-Input Multiple-Output) for spatial multiplexing.

2.5.4 Self organizing network functionalities

Within 3GPP Release 8, LTE considers Self Organizing Network (SON) functions. Some of the SON functions have already been standardized and others are in still being studied. SON concerns both self-configuration and self optimization processes. Self configuration process is defined as the process where

newly deployed nodes are configured by automatic installation procedures to get the necessary basic configuration for system operation [63]. The determination of automatic neighbour cell relation list [71] [72]is an example of self-configuration process that is being standardized in LTE Release 8. Self-optimization process is defined as the process where user equipment and eNB measurements are used to autotune the network.

2.6 Interference in e-UTRAN system

In this thesis, one of the applications of proposed automated healing methodology is to heal the interference mitigation parameters. This motivates us to present in this chapter an interference model for the e-UTRAN. The interference is given based on a system model which includes the eNBs distribution and the propagation model [73]. The interference model is used in a second step by the network level simulation.

2.6.1 System model and assumption

In this section, we analyze only the interference in the downlink. For the uplink, the same concept should be followed. In downlink, each terminal reports an estimate of the instantaneous channel quality to the cell. These estimates are obtained by measuring on a reference signal, transmitted by the cell and used also for demodulation purposes. Based on the channel-quality estimate, the downlink scheduler grants an arbitrary combination of 180 kHz wide resource blocks in each 1 ms scheduling interval. Since the time scale of scheduling is very small, we will not take into account the scheduling process in the interference model and in the system level simulations. Only propagation loss and shadow fading, namely channel variations over large time scales are considered. However, small-scale variations (multi-path fading) are considered in the link level simulations which serve as an input to the present study. The link level simulations return a link curve which represents the throughput as a function of the received Signal to Interference plus Noise Ratio (SINR). The *Okumara-Hata* propagation model is used in the 2 GHz band. The attenuation L is given by $L = l_0 d^{\gamma} \zeta$, where l_o is a constant depending on the used frequency band, d is the distance between the eNB and the mobile, γ is the path loss exponent and ζ is a log-normal random variable with zero mean and standard deviation σ representing shadowing losses.

2.6.2 Interference model

In e-UTRAN system, user signals are orthogonal in the same cell thanks to the OFDMA access technology. As a consequence there is no intra-cell interference. On the other hand, the same frequency band can be used by a given (central) cell and by some other neighbouring cells. This generates inter-cell interference which limits the performance of the e-UTRAN system. In the downlink for instance, inter-cell interference occurs at a mobile station when a nearby eNB transmits data over a subcarrier used by its serving eNB. The interference intensity depends on user locations, frequency reuse factor and loads of interfering cells. For instance, with a reuse factor equals 1, low cell-edge performances may be achieved whereas for reuse factor higher than 1, the cell-edge problem is resolved at the expense of resource limitation. To make an optimal trade-off between inter-cell interference and resource utilization, different interference mitigation schemes are proposed in the standard [64].
One of the techniques for interference mitigation is the Inter-Cell Interference Coordination (ICIC). It is a scheduling strategy in the frequency domain that allows increasing the cell edge data rates. Basically, ICIC implies certain (frequency domain) restrictions to the uplink and downlink schedulers in a cell to control the inter-cell interference. By restricting the transmission power of parts of the spectrum in one cell, the interference seen in the neighbouring cells in this part of the spectrum is reduced. This same part of the spectrum can then be used to provide higher data rates for users in the neighbouring cells. This mechanism is called also partial (or fractional) frequency reuse because the total available bandwidth is reused in all cells with power restrictions on certain subbands (Figure 2.4).

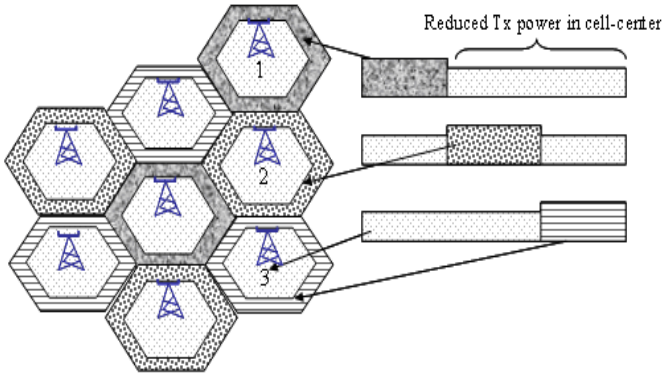


Figure 2.4: Inter-cell interference coordination scheme

In the fractional frequency reuse scheme used in this thesis, an admitted

user gets resource blocks from the portion of the bands as a function of its received signal quality in the cell. One band that is allocated to mobiles with the worst signal quality and is denoted as a protected band or, metaphorically, as an cell edge band. A user with poor radio condition is often situated at the cell edge, but could also be closer to the base station and experience for example a deep in the shadow fading. Hence, the graphical representation is only a logical one. When the cell edge band is full the remaing unassigned users are assigned to cell center band.

Assuming that the spectral band is composed of C resource blocks, one third of the band is reserved for the cell edge users and the rest is for cell centre users.
The eNB transmit power in each cell-edge resource block equals the maximum transmit power P. To reduce intercell interference, the eNB transmit power in the cell-center band must be lower than P. Let εP (where $\varepsilon \leq 1$) be the transmit power in the cell-center band.
The interference should be determined for two different users according to their positions: the cell-center user and the cell-edge user. Let m_c and m_e be two users connected to a cell k. The mobile m_c uses the central band whereas m_e uses the cell-edge band. Let Λ denote the interference matrix between cells, where the coefficient $\Lambda(i,j)$ equals 1 if cells i and j use the same cell-edge band and zero otherwise.
For cell-edge user m_e, the interference comes from signal to users in the cell center of the closest adjacent cells and from the cell-edge user in other cells. The mobile m_e connected to the cell k and using one resource block in the cell-edge band, receives an interfering signal from a cell i equals

$$I_{i,m_e} = \Bigg((1 - \Lambda(k,i)\beta_i^c \varepsilon P_i + \Lambda(k,i)\beta_i^e P_i) \, \frac{G_{i,m_e}}{L_{i,m_e}} \tag{2.12}$$

where P_i is the downlink transmit power per resource block of the cell i. G_{i,m_e} and L_{i,m_e} are respectively the antenna gain and the path loss between cell i and the model station m_e. The factor β_i^c (respectively β_i^e) is the probability that the same resource block in the center-cell band (respectively the cell-edge band) is used at the same time by another mobile connected to the cell i.
Using analysis given in Annexe A, the total interference perceived by user m_e is the sum of all interfering signals

$$I_{m_e} = \sum_{i \neq k} \tilde{\Lambda}_e(k,i) \chi_i \frac{P_i G_{i,m_e}}{L_{i,m_e}} \tag{2.13}$$

the term $\tilde{\Lambda}_e(k,i) = 3\left((1 - \Lambda(k,i)) \, \frac{1-\alpha_i}{2} \varepsilon + \Lambda(k,i)\alpha_i\right)$ is interpreted as a new

interference matrix, denoted here as the fictive intereference matrix for cell-edge users. The factor α_i is defined as the proportion of traffic served in the cell-edge band off cell i.
For the cell-center user m_c, the interference comes from users in the cell-edge and cell-center of closest adjacent cells and also from the cell-center and cell-edge users in other cells. Similarly, to the cell-edge users, the mobile m_c connected to the cell k and using one resource block in the cell center band, receives an interfering signal equal to

$$I_{m_c} = \sum_{i \neq k} \tilde{\Lambda}_c(k,i) \chi_i \frac{P_i G_{i,m_c}}{L_{i,m_c}} \tag{2.14}$$

Here, the fictive interference matrix in the cell-center band is given by

$$\tilde{\Lambda}_c(k,i) = 3 \left(\frac{1}{2} \left(1 - \Lambda(k,i)\right) \left(\frac{1-\alpha_i}{2} \varepsilon + \alpha_i \right) + \Lambda(k,i) \frac{1-\alpha_i}{2} \varepsilon \right) \tag{2.15}$$

The download SINR is then given by

$$SINR_m = \frac{P_k G_{k,m}}{L_{k,m} \left(I_m + N_{th}\right)} \tag{2.16}$$

In equation (2.16), the subscript m stands for m_e if the mobile considered is a cell-edge user and m_c for the cell-center user. N_{th} is the thermal noise per resource block. For more details about the LTE interference model, reader is invited to see the Annexe A.
In e-UTRAN system, an adaptive modulation and coding scheme is used [69] [70]. So, the choice of the modulation depends on the value of the SINR through the perceived Block Error Rate ($BLER$). The decrease of the $SINR$ will increase the $BLER$, forcing the eNB to use a more robust (less frequency efficient) modulation. The latter may have negative impact on the communication quality. For instance, a lower modulation efficiency results in a lower throughput and a larger transfer time for elastic data connections. The throughput per resource block for each user is determined by link level curves. The user physical throughput is N_m times the throughput per resource block, where N_m represents the number of resource blocks allocated to the user m.

2.7 e-UTRAN handover algorithm

Mobility in e-UTRAN is based on hard handover rather than on soft handover as in UMTS. The mechanism of hard handover has been used in 2nd

generation GSM networks and has shown to be efficient for mobility management. In a hard handover, the user keeps the connection to only one cell at a time, breaking the connection with the former cell immediately before making the new connection to the target cell. The basic concept of handover as in GSM is likely to be implemented in e-UTRAN except for the handover preparation phase, which requires new mechanisms.
The reason for abandoning soft handover is related to the extra-complexity involved in its implementation, and the fact that it is not suitable for inter-frequency handover. Furthermore, soft handover handicaps system capacity in highly loaded network condition and with high number of users in soft handover situation. To guarantee seamless and lossless hard handover in e-UTRAN, the handover triggering time should be as low as possible [64]. In order to study e-UTRAN hard handover, some assumptions for the Call Admission Control (CAC) and resource allocation are made. In the CAC algorithm, a user can be admitted to the network only when the following conditions are fulfilled:

- Good signal strength: the mobile selects the cell that offers the maximum signal. If this signal is lower than a specified threshold then the mobile is blocked because of coverage shortage. This condition is in fact a selection criterion. It is noted that in 3GPP, there is no specification for LTE cell selection and reselection.

- Resource availability in the selected cell: the mobile can be granted physical resources in terms of resource blocks between a minimum and maximum threshold. When the signal strength condition is satisfied, the eNB checks for resource availability. If the available resource is lower than a minimum threshold, the call is blocked.

Hard handover is performed in this study using a similar algorithm to the one used in GSM: while in communication, the mobile periodically measures the received power from its serving eNB and from the neighbouring eNBs. The mobile, initially connected to a cell k, triggers a handover to a new cell i if the following conditions are satisfied:

- The Power Budget Quantity (PBQ) is higher than the handover margin:
$$PBQ = P_i^* - P_k^* \geq HM(k,i) + Hysteresis \tag{2.17}$$
where P_k^* is the received power from the eNB k expressed in dB; $HM(k,i)$ is the handover margin between eNB k and i. The $Hysteresis$ is a constant independent of the eNBs and mobile stations and is fixed in this study to 0. Here, we define $HM(k,i)$ as the outgoing HM of

eNB k to eNB i. Conversely, $HM(i,k)$ can be termed as incoming HM of eNB k from eNB i.

- The received power from the target eNB must be higher than a threshold. This is the same condition as in the CAC process.
- Enough resource blocks are available in the target eNB.

The last condition requires information exchange between eNBs because the original cell has to know a priori the load of the target cell; otherwise the handover is blind and the communication risks to be dropped. In an inter-eNB handover procedure, the source eNB is responsible for performing handover preparation to the target eNB based on measurement report transmitted by the mobile.

2.8 Conclusion

This chapter not only provides an overview of the previous work on the troubleshooting to wireless networks but also the background material used in the thesis for the automated healing in LTE networks . At the beginning of the chapter, a survey of the previous work and the literature in this domaine for the GSM and UMTS networks has been made. The fault detection using self Organizing Map (SOM) and Neural Networks (NN) is reviewed. The work on the fault diagnosis using Bayesian Networks and alarm correlation method has also been described. The problem solution or the automated healing aspect of the troubleshooting remains unexplored and has motivated the present thesis.

An overview of the statistical learning techniques used in our work is also presented. In the begining, the linear regression has been used in our work because of its simplicity. Later, logistic regression is used because of its ability to model the saturation in the KPI values.

Chapter 3

Statistical Learning for Automated RRM

3.1 Introduction

The purpose of this chapter is to introduce statistical learning approach for the automated Radio Resource Management (RRM). Assume we have access to a database of Key Performance Indicators (KPIs) and the corresponding RRM parameters. By using simple statistical learning techniques such as regression, one can extract from the data closed form expressions, denoted here as the *model*, that approximate the functional relations between KPIs and the RRM parameters. Once the *model* is available, it can be used to devise efficient automated RRM algorithms. Two examples of such algorithms in the context of LTE mobility are considered. The first one is a monitoring process. The *model* extracted from the data is used to improve the monitoring process and, when necessary, to guide the expert in modifying the mobility parameter. The second case study uses the *model* to devise an efficient auto-tuning algorithm for the mobility parameter.
The chapter is organized as follows: Section 3.2 introduces the statistical learning approach for the *model* extraction. Section 3.3 details the results of the statistical learning approach applied to 2 neighbouring eNBs for a LTE simulation scenario. Section 3.4 describes the general block diagram of the proposed automated RRM scheme. Sections 3.5 explains how the extracted *model* can be used in monitoring. Section 3.6 and 3.7 explain the application of this *model* to auto-tuning processes, followed by concluding remarks in Section 3.8.

3.2 Statistical learning approach

The section introduces the important phases of the statistical learning approach. The first step in this approach is the construction of a training database that contains KPI data with the corresponding RRM parameters. After the construction of the database, the closed form expressions that relate the RRM parameter to the KPIs, known as the *model*, are calculated using regression.

3.2.1 Learning phase

The first step in the statistical learning approach is to construct a training set on which the learning is done. Generally, this training set is constructed from the observations. However, the quality of the training set is a major issue in learning, because a bad-quality training set causes insufficient learning that leads to bad-quality prediction/estimation. To begin with, a good training set must contain sufficient amount of data points. Secondly, these data points must more or less cover the whole region of interest. In other words, there must be sufficient amount of information diversity in the training set. The greatest challenge regarding our problem is that these requirements are very rarely fulfilled by network measurements for an operational network since we have to test a large variety of parameter configurations (including relatively bad ones) on a network that is supposed to give satisfactory QoS to the customers. In most of the cases, we have data with limited parameter settings (if not with a unique parameter setting) that limits the generalization (or extrapolation) capacity of the predictor. One of the solutions is to construct a training database by simulation samples over the whole region of interest and extract tendencies or relative behaviours instead of absolute ones. Once tendencies or relative behaviours are found, a corrective mapping between simulations and measurements to find absolute values that can be directly applied to the real network. In any case, the reliability of the predictions depends as well on the accuracy of the simulations as the quality of the measurements.

3.2.2 Forward regression

In forward regression, the KPIs are tried to be estimated as a function of the RRM parameter p:

$$KPI_j = f_j(p) \tag{3.1}$$

where J is the number of KPIs of interest (the number of response variables). Note that here, there are J regressions and each regression has just one

explanatory variable. Since we consider linear functions (f_j 's are linear), the above equation can be re-written as:

$$\begin{bmatrix} KPI_1 \\ KPI_2 \\ \cdots \\ KPI_J \end{bmatrix} = \begin{bmatrix} \beta_{10} & \beta_{11} \\ \beta_{20} & \beta_{21} \\ \cdots & \cdots \\ \beta_{J0} & \beta_{J1} \end{bmatrix} \begin{bmatrix} 1 \\ p \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \cdots \\ \epsilon_J \end{bmatrix} \tag{3.2}$$

Note that β_{j0} represents the bias for KPI_j which may be different for each KPI and which depends upon the unknown factors like propagation, interference factor etc.

Forward regression helps us to see how a change in a certain RRM parameter influences several different KPIs. In this way, we can determine the effect of modifying the RRM parameter on different KPIs. Figure 3.1 depicts the block diagram of the forward regression test that determines the regression quality. The KPIs yielded by simulations and those yielded by the regression equation for a pre-determined value of the RRM parameter are compared.

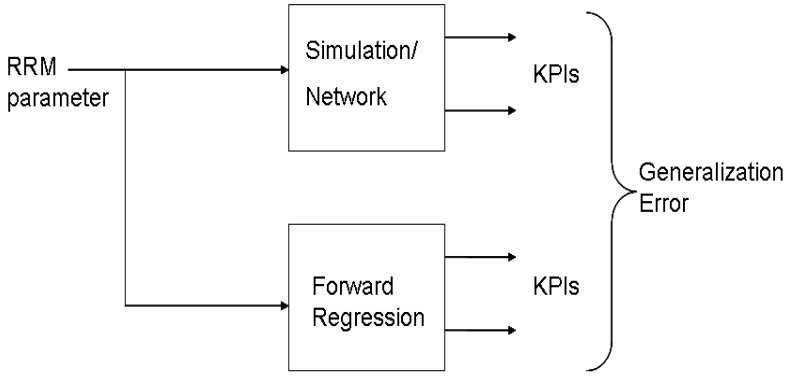


Figure 3.1: Forward regression test for Handover Margin (HM).

3.2.3 Backward regression

Backward regression tries to estimate the RRM parameter p as a function of the KPIs:

$$p = g(KPI_1, KPI_2, ..., KPI_J) \tag{3.3}$$

or in multiple linear regression formulation:

$$p = \beta_0 + \beta_1 KPI_1 + \beta_2 KPI_2 + ... + \beta_J KPI_J + \epsilon \tag{3.4}$$

Backward regression helps us to determine which RRM parameter yields a given KPI vector. In this way, we can determine the exact value of the RRM parameter for a desired KPI performance. Figure 3.2 depicts the block diagram of the backward regression test that determines the regression quality. The RRM parameter that produces the set of KPIs through the simulator and the RRM parameter obtained through the backward regression that uses these KPIs are compared.

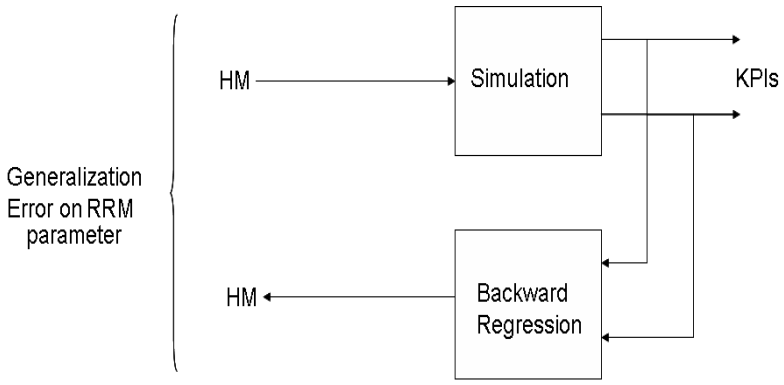


Figure 3.2: Backward regression test for HM.

3.3 Regression Results: Two eNB scenario

3.3.1 LTE network simulation scenario

We are interested in the downlink access of a LTE system. The simulations are carried out using a MATLAB simulator described in [89]. The traffic simulated is a FTP traffic with a file size of 5700 Kbits for download. Call arrivals are generated using the Poisson process and the communication duration of each user depends on its bit rate. The propogation model used is Okumara-Hata model. The bandwidth used is 5 MHz. The maximum number of Physical Resource Blocks (PRBs) in an eNB, i.e. the capacity, is fixed to 20 PRBs (eNB capacity can be fixed by the operator). The minimum and maximum number of chunks that can be allocated to each user are 1 and 3 respectively. The resources are allocated on the first-come first-served basis. The simulator performs correlated Monte Carlo snapshots with time steps of one second to account for the time evolution. At the end of each time step of one second, new mobile positions are updated, HO events are processed, new users are admitted according to the access conditions and some other users leave the network (end their communications or are dropped). 90% of the

mobiles are in motion with a speed of 15 meters per second and rest are still. The simulations are run for 3500 time steps and the KPIs are averaged on the interval between 1000 and 3500 seconds to account for transient effects. The RRM parameter setting is fixed during the simulation duration of 3500 time steps.

3.3.2 Results

The regression analysis is done on a two eNB scenario using Linear Regression. The learning database is generated by setting the arrival rate of the mobiles in the network to 1 arrival per second. The RRM parameter considered in this analysis is the LTE mobility parameter Handover Margin (HM), explained in Section 2.7. The HM from eNB_1 to eNB_2 (HM_{12}) is varied from 2dB to 11.95dB in steps of 0.05dB, while the HM value from eNB_2 to eNB_1 (HM_{21}) is simultaneously decreased from 10dB to 0.05dB in steps of 0.05dB in order to enhance the effect of the HM on the KPIs. Scatter plots of Block Call Rate (BCR) of eNB_1 (BCR_1) and of eNB_2 (BCR_2) are shown in Figures 3.3 and 3.7 respectively, as functions of HM_{12}. It can be seen that the relationship of both BCRs with HM_{12} is linear. The strength of this linear relationship is also indicated by the Pearson's correlation coefficient [92]. Scatter plots for Drop Call Rate (DCR), Load (L) and Average Bit Rate (ABR) of both eNBs exhibit similar behaviours as shown in Figures 3.4 to 3.10.

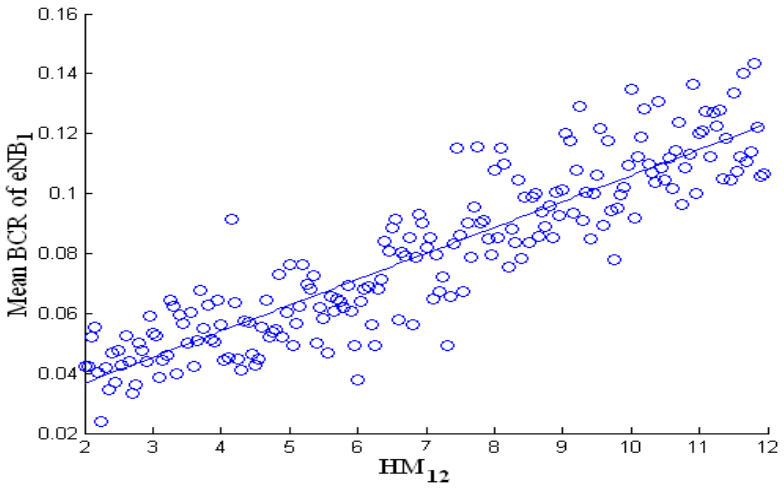


Figure 3.3: Scatter plot between HM and mean BCR of eNB_1, correlation coefficient=0.9.

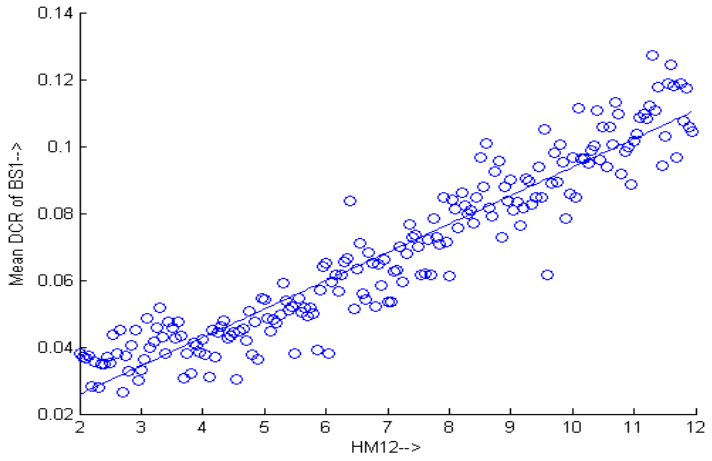


Figure 3.4: Scatter plot between HM and Mean DCR of eNB1, Correlation coefficient=0.94713.

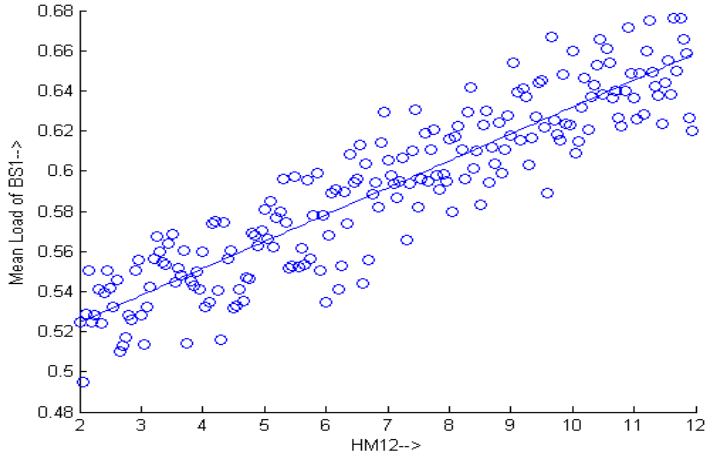


Figure 3.5: Scatter plot between HM and Mean Load of eNB1, Correlation coefficient=0.91135.

Once the learning database is constructed, the regression equations relating the RRM parameter to the KPIs are calculated followed by a 5-fold cross-validation that judges the quality of the regression. In 5-fold cross-validation, the data is divided into five subsets with four data subsets selected for training and one subset used for calculating the validation error of the regression. In this way, all possible combinations of four data subsets are chosen for training with one leftover for validation. The regression equations

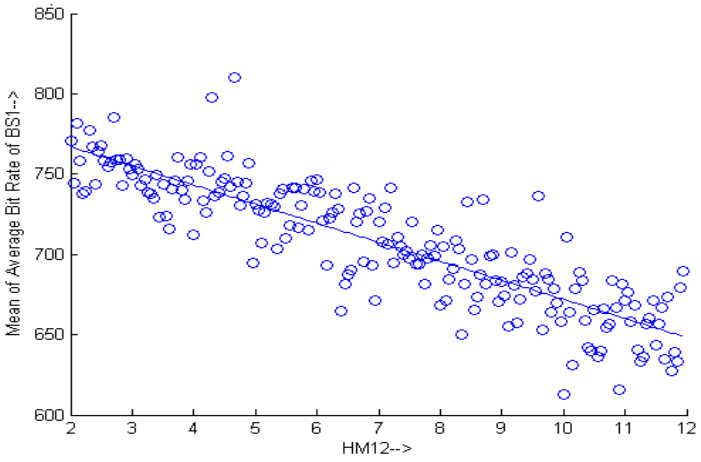


Figure 3.6: Scatter plot between HM and Mean Average Bit Rate of eNB1, Correlation coefficient=0.86923.

corresponding to the combination with the least validation error are selected.

The regression equations calculated for the KPIs of eNB_1 and eNB_2 are as follows:

$$BCR_1 = 0.0191 + 0.00873 * HM_{12} \tag{3.5}$$

$$DCR_1 = 0.00787 + 0.00773 * HM_{12} \tag{3.6}$$

$$L_1 = 0.493 + 0.0143 * HM_{12} \tag{3.7}$$

$$ABR_1 = 784 - 11.6 * HM_{12} \tag{3.8}$$

$$BCR_2 = 0.109 - 0.00701 * HM_{12} \tag{3.9}$$

$$DCR_2 = 0.0935 - 0.00591 * HM_{12} \tag{3.10}$$

$$L_2 = 0.665 - 0.0154 * HM_{12} \tag{3.11}$$

$$ABR_2 = 675 + 11.01 * HM_{12} \tag{3.12}$$

The average 5-fold cross-validation error of this regression analysis is 13.35% for BCR_1, 11.29% for DCR_1, 2.42% for L_1, 2.11% for ABR_1, 17.62% for BCR_2, 11.48% for DCR_2, 2.55% for L_2 and 2.24% for ABR_2. The bar graph showing 5-fold cross-validation error for all the KPIs is given in Figure 3.11

As for the second test, the block diagram of 3.1 is used. A value of $HM_{1,2}$ that is not in the training database is taken. On the one hand, KPIs are determined by simulations and on the other hand, by the regression

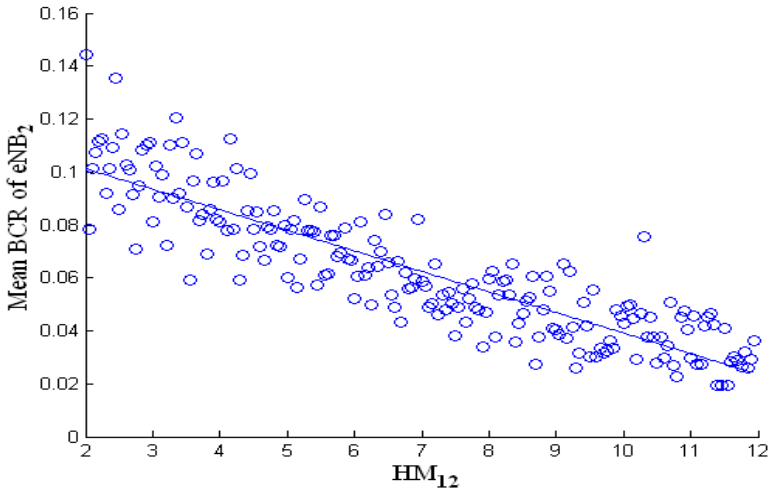


Figure 3.7: Scatter plot between HM and mean BCR of eNB_2, correlation coefficient=0.88.

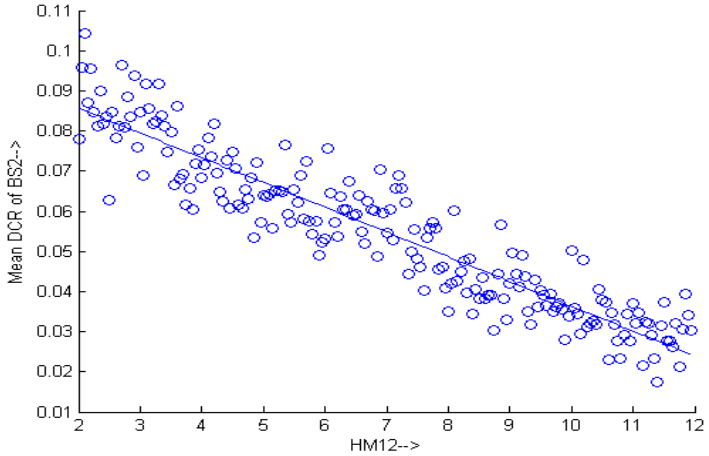


Figure 3.8: Scatter plot between HM and Mean DCR of eNB2, Correlation coefficient=0.92941.

equations. Finally, the percentage error between the two set of KPIs are calculated. Table 3.1 depicts the percentage errors of the KPIs for a value of $HM_{1,2} = 7.33$dB (note that due to the symmetric variation of the HMs $HM_{2,1}$ is 12dB-7.33dB=4.67dB).

As can be observed from Table 3.1, the forward generalization error remains below 6% for all the KPIs. It means that it is possible to use the regression equations instead of the simulations up to an error level of 6% for

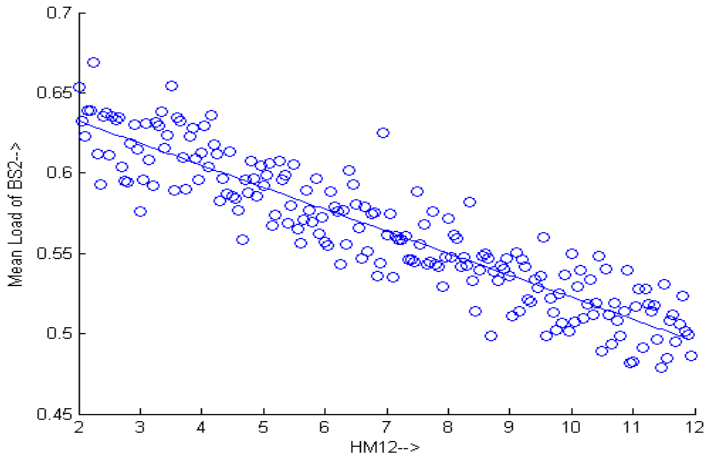


Figure 3.9: Scatter plot between HM and Mean Load of eNB2, Correlation coefficient=-0.9148.

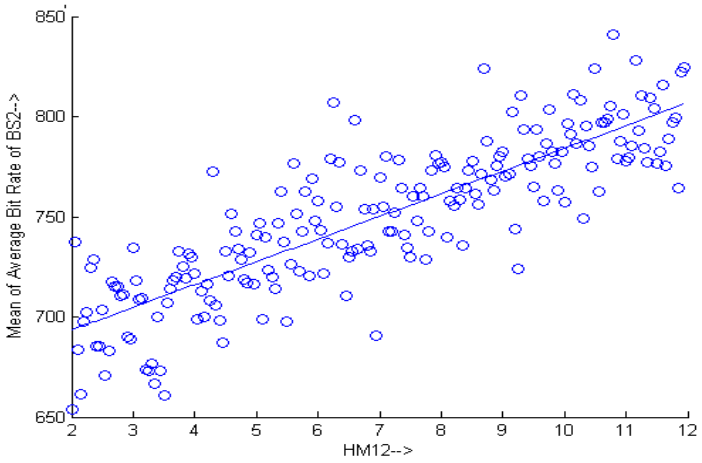


Figure 3.10: Scatter plot between HM and Mean Average Bit Rate of eNB2, Correlation coefficient=0.84188.

the two eNB scenario.

The backward regression is also tested on the LTE system, yielding the following regression equation:

$$\begin{aligned} HM_{12} &= 7.71 + 9.11 * BCR_1 + 34.3 * DCR_1 \\ &\quad +46.7 * L_1 - 0.00591 * ABR_1 - 13.01 * BCR_2 \\ &\quad -16.7 * DCR_2 - 39.7 * L_2 + 00.00281 * ABR_2 \end{aligned} \tag{3.13}$$

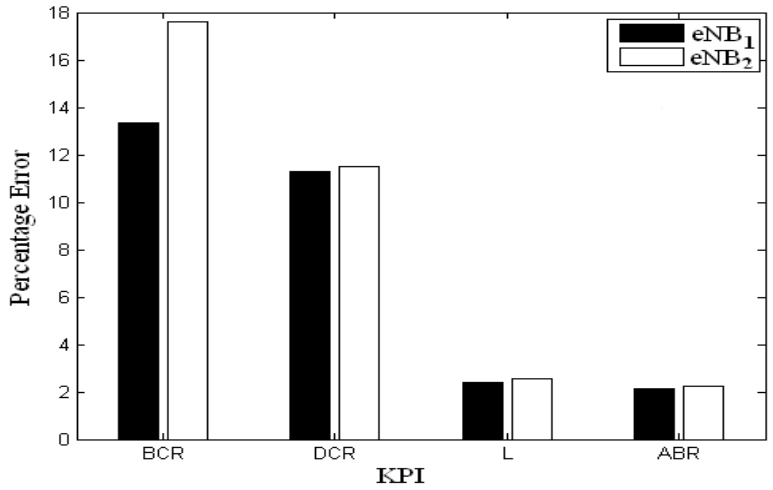


Figure 3.11: 5-fold cross validation error for forward regression.

KPI	Simulation Value	Regression Value	% error
BCR_1	0.0869	0.0834	4.0
DCR_1	0.0695	0.0708	1.91
L_1	0.6079	0.5965	1.88
ABR_1	740.0427	703.6917	4.91
BCR_2	0.0567	0.0594	4.70
DCR_2	0.0558	0.0527	5.60
L_2	0.5402	0.5595	3.58
ABR_2	750.5576	753.5668	0.40

Table 3.1: The Percentage error of KPIs for the forwarde regression test

Note that the backward regression equation is also linear in all the KPIs. The 5-fold cross-validation error of this equation is 6.52%. Figure 3.12 shows the percentage error of the backward regression test (depicted in Figure 3.2):

We see that the percentage error in the backward regression test remains below 8% for all significant values of HM_{12}.

To see the effect of traffic, 3D scatter plots of BCR are plotted in Figure 3.13, with BCR_1 on the x-axis, BCR_2 on the y-axis and HM_{12} on the z-axis, for traffic values of 0.5, 0.75, 1 and 1.5 arrivals/s. One can see the change of slope with traffic increase. Scatter plots of other KPIs show similar trend.

In the case of load, it is evident from Figure 3.14 that with increasing traffic values; the corresponding curves are shifted in the higher load regions but on the other hand the change in the slope of the curves is not very obvious.

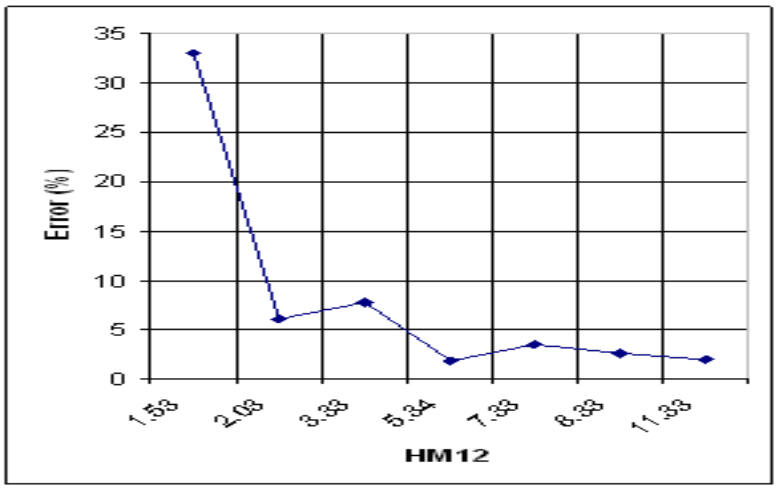


Figure 3.12: Backward regression generalization errors.

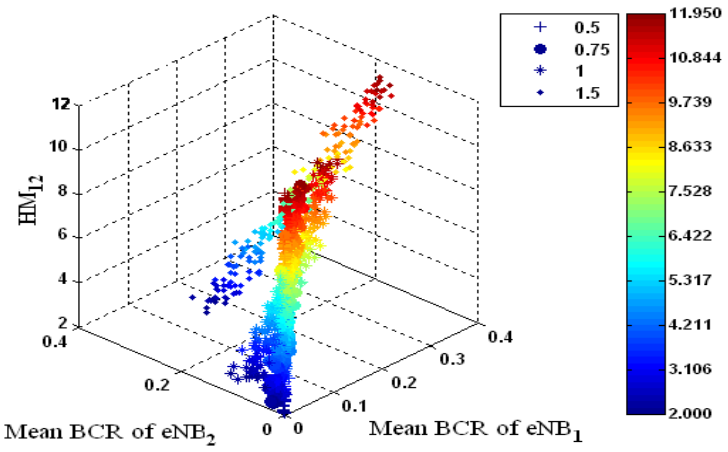


Figure 3.13: 3D Scatter plot between HM_{12} and mean BCRs of eNB_1 and eNB_2.

For the case of average throughput it is apparent from Figure 3.15 that increasing traffic values results in the corresponding traffic curves moving to the low average throughput region and we also observe the change in the slope of the curves. This decrease in the average throughput is due to the fact that the total number of available PRBs is divided between the users with increasing traffic. As a result, the average throughput decreases.

In the case of Drop Call Rate (DCR), we can see that increasing traffic values, as in Figure 3.16, results in the corresponding curves moving to the

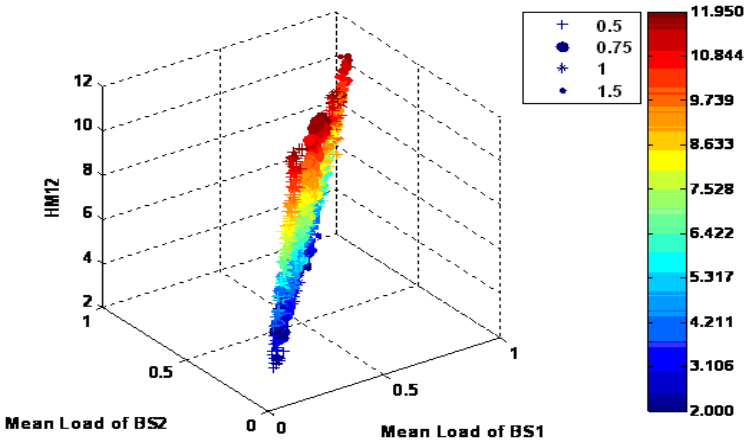


Figure 3.14: 3D Scatter plot between HM12 and Mean Load of eNB1 and eNB2.

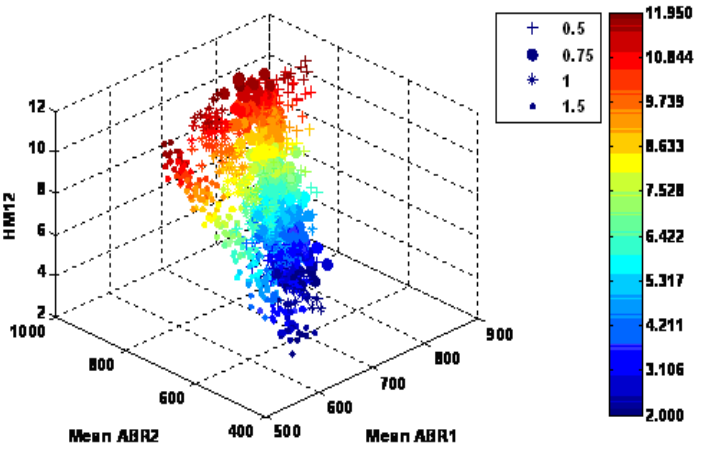


Figure 3.15: 3D Scatter plot between HM_{12} and Mean ABR of eNB1 and eNB2.

higher DCR region. However, the change in the slope is not very obvious. We can also observe for the individual curves that increasing the HM value between eNB1 and eNB2, results in an increase in mean DCR1 and decrease in mean DCR2.

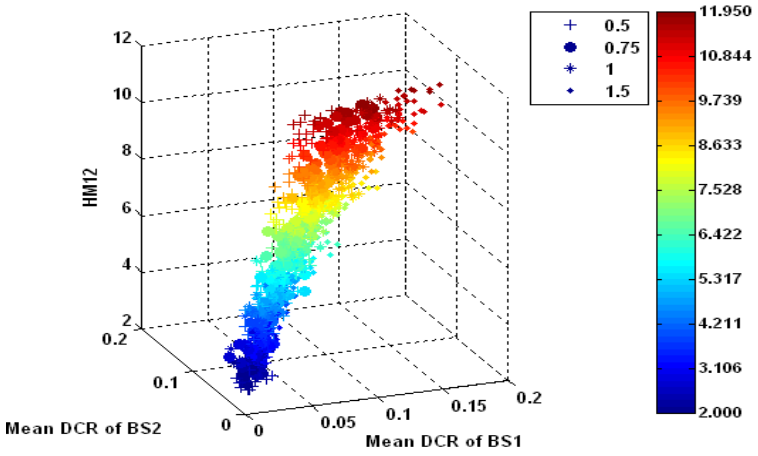


Figure 3.16: 3D Scatter plot between HM12 and Mean DCR of eNB1 and eNB2.

3.4 Automated RRM system description

The block diagram of the proposed automated RRM scheme is shown in Figure 3.17. The database of the KPIs and the corresponding RRM parameters is generated by the network or a simulator. This Database is then used by the Model Extraction block which uses statistical learning to estimate the system *model*. This *model*, alongwith the network KPIs, are then used by the RRM Auto-Tuning block to find a new RRM parameter value. The new RRM parameter value is injected into the Network. The *model* is also used by the monitoring block which, depending upon the current RRM parameter in the network, and the KPIs, locates the current QoS working point. The quantitative measures for the change in the RRM parameter for improvement in KPI performance can then be proposed.

3.5 Monitoring

The proposed statistical approach can be applied to network monitoring. The scatter plots of Figures 3.18 to 3.21 depict the statistical relationship between various KPIs of neighbouring eNBs, having the KPI of eNB_1 on the x-axis, the KPI of eNB_2 on the y-axis and the RRM parameter (HM_{12}) on the z-axis which is projected on the xy-plane. These scatter plots permit us to:

1. determine the QoS operating point and observe its position with respect

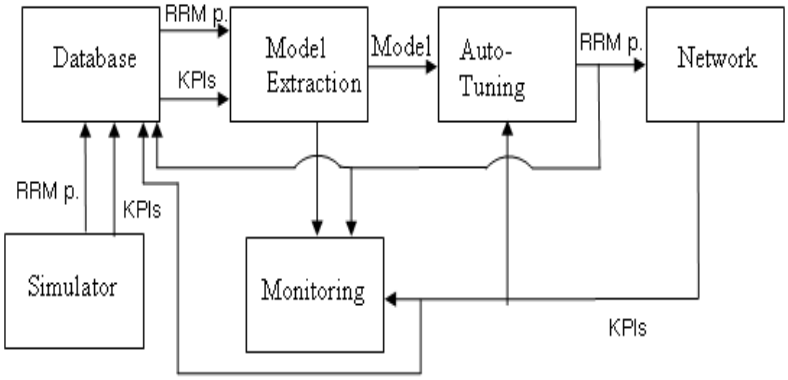


Figure 3.17: Block diagram for automated RRM.

to alarm thresholds,

2. solve a QoS problem by using the knowledge of how the operating point will move in different KPI plots.

Hence, the monitoring will provide a visible information on the network performance of the working point. Also, using equation (3.1), we can get the quantitive measure about the change in the RRM parameter required to achieve a desired QoS objective.

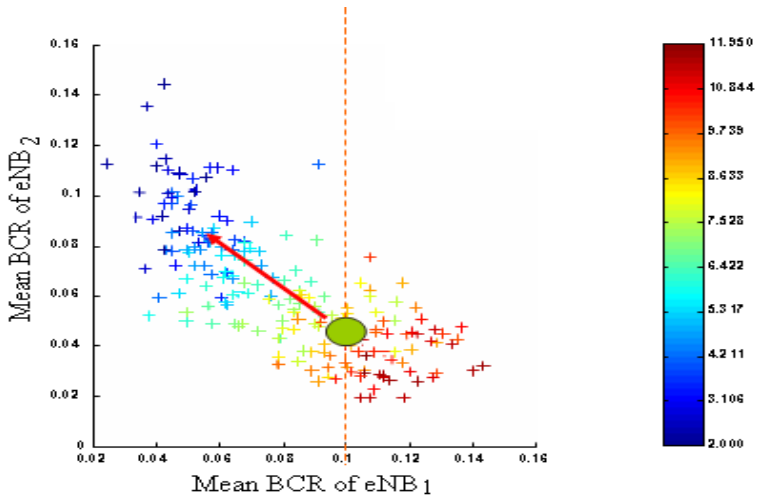


Figure 3.18: 3D scatter plot between HM_{12} and mean BCRs of eNB_1 and eNB_2, projected on the KPI plane.

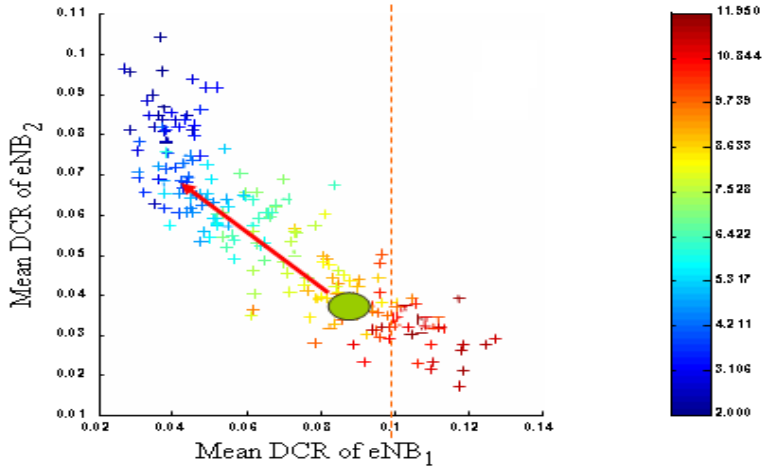


Figure 3.19: 3D scatter plot between HM_{12} and mean DCRs of eNB_1 and eNB_2, projected on the KPI plane.

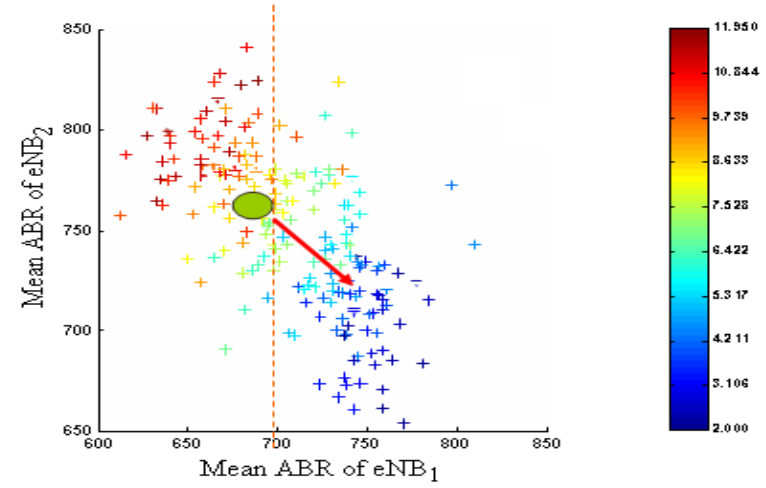


Figure 3.20: 3D scatter plot between HM_{12} and mean ABRs of eNB_1 and eNB_2, projected on the KPI plane.

3.6 Optimization model

The information gained through the statistical analysis approach detailed above can be used to derive heuristics for an iterative optimization algorithm that improves the KPI performance of a radio network by using the statistical relationships between the RRM parameters and the KPIs. In the case the training database is constructed from simulations, we can not use

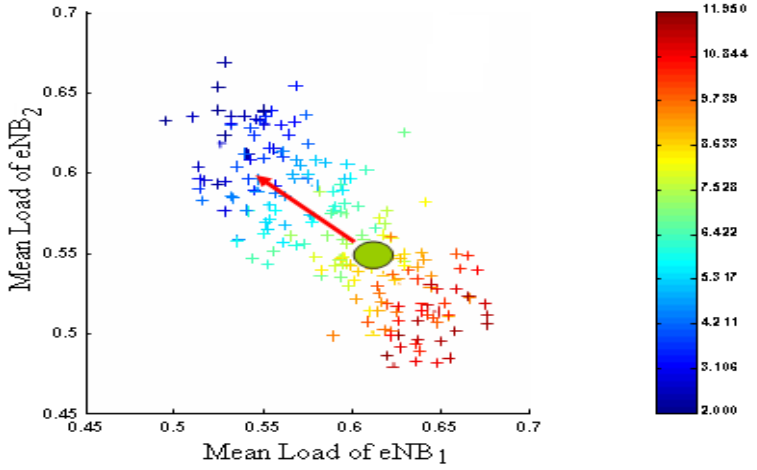


Figure 3.21: 3D scatter plot between HM_{12} and mean Loads of eNB_1 and eNB_2, projected on the KPI plane.

the statistical relationships exactly as they are, due to the inevitable discrepancy between the measurements and the simulations. As stated in the beginning of Section 3.2.1, it is possible to 'correct' the simulations before going into the learning phase (this correction can be done by supervised learning schemes or data assimilation techniques). Another alternative is to perform the statistical analysis with the simulation data, and to use the main trends of the results instead of using the exact regression expressions. For the mobility problem at hand, the trend that we can use is the linearity of the relationship between the chosen KPIs and the HM. The parameters of the linear relationship, namely the slope and the intercept, are not known exactly since they are determined by real measurements. With this consideration, we have conceived a dynamic-iterative algorithm that varies the HM between each couple of eNBs according to the observed KPIs. At each iteration, the algorithm updates the HM of eNB couples by considering two phenomena:

- performance improvement of the eNB with controlled degradation of the neighbouring eNB,
- performance improvement of the neighbouring eNB with controlled degradation of the eNB.

For determining the variations of the HM in these two phenomena, the linearity between the HM and the KPIs is assumed.

3.7 Automated RRM: Description and results on LTE

Using the statistical analysis explained in previous sections, we propose a dynamic auto-tuning algorithm that enhances the KPI performance of the eNBs by varying their outgoing HMs. The heuristic used in the algorithm modifies the outgoing HMs of bad-performing (resp. good-performing) eNBs with those neighbours whose KPI performance is good (resp. bad). In other words, it trades the KPI performance between neighbouring eNBs by adjusting the HMs according to the statistical analysis results. However, we can not always construct the *model* with real network data and resort to working with simulation data. Despite this obstacle, we can still benefit from the statistical analysis by using the functional form of the regression (linear in this case).

Consider an eNB eNB_i with a KPI value KPI_i, and its neighbouring eNB eNB_j with KPI_j. Let ν_{KPI} and HM_{ij} denote the the target KPI value and the outgoing HM of eNB_i towards eNB_j, respectively. Without loss of generality, consider a KPI of type BCR or DCR the allowable region of which lies below ν_{KPI}.

If KPI_i (resp. KPI_j) is above ν_{KPI} and KPI_j(resp. KPI_i) is below ν_{KPI}, we can decrease (resp. increase) HM_{ij} so that KPI_i (resp. KPI_j) decreases toward ν_{KPI} and KPI_j (resp. KPI_i) increases towards ν_{KPI}. The aim is to find the amount of increase/decrease in HM_{ij}, ΔHM_{ij}, that brings KPI_i (resp. KPI_j) to a maximum value equal to ν_{KPI} without causing KPI_j (resp. KPI_i) to rise above ν_{KPI}. Applying the calculated ΔHM_{ij} values to all eNB couples in an iterative manner, we expect to converge to an overall KPI performance where KPIs that are above ν_{KPI} are brought *down* to ν_{KPI} at the expense of the other KPIs that are brought *up* to ν_{KPI}. From the regression results presented in Section 3.3, we know that the increase/decrease in the KPIs will be linearly proportional to the increase/decrease in HM_{ij}. Furthermore, we can also say that the slope of this linear proportionality is determined to a great extent by the HandOver (HO) flux i.e., the number of mobiles making HO from eNB_i to eNB_j (ϕ_{ij}). Then we can write for eNB_j:

$$|\nu_{KPI} - KPI_j| \propto |\phi_{ij}|\,|\Delta HM_{ij}| \tag{3.14}$$

where ΔHM_{ij} is the increase/decrease in HM_{ij} to bring KPI_j to ν_{KPI}. A similar equation can be written for eNB_i. The linear proportionality can easily be extended to other neighbours. Considering another neighbour,

eNB_k, we can express the proportinality between eNB_j and eNB_k as:

$$\frac{|\Delta HM_{ij}|}{|\Delta HM_{ik}|} = \frac{|\nu_{KPI} - KPI_j|}{|\nu_{KPI} - KPI_k|} \frac{|\phi_{ik}|}{|\phi_{ij}|} \triangleq \eta_{jk} \tag{3.15}$$

Equation (3.15) yields the ratio of HM modifications between 2 eNBs, for a given ratio of KPI changes. Having a proportionality equation for all neighbouring eNB couples is not sufficient to find the exact values of $|\Delta HM_{ij}|$s. The additional equation comes from the boundary conditions: the overall effect of modifying HM_{ij} for all the neighbouring eNBs of eNB_i must be limited in order not to cause too much increase/decrease in KPI_i: because every neighbouring eNB having a KPI performance that can be traded with that of eNB_i, contributes to the overall change in KPI_i. Therefore, the proposed algorithm puts an upper-bound on the sum of $|\Delta HM_{ij}|$ over all neighbours eNB_i. The upper bound is not fixed but is calculated as a function s of the current *operating point* (HM_{ip}, KPI_i). HM_{ip} is the outgoing HM of eNB_i towards a *pivotal* eNB (eNB_p) which is chosen as the neighbour having the most extreme KPI performance (with respect to the threshold ν_{KPI}). The function s that limits the maximum HM modification is chosen as a product of 2 sigmoid functions one of which is a function of HM_{ip} and the other being a function of KPI_i. The former allows large and upper-bounded step sizes in $|\Delta HM_{ij}|$ when KPI_i is far away from ν_{KPI} and vice versa. In a similar manner, the latter allows large and upper-bounded step sizes in ΔHM_{ij} when HM_{ip} is far away from its extreme values and vice versa. In the case where $KPI_i < \nu_{KPI}$, the function s limits the overall *increase* in KPI_i via the limitation in HM values:

$$s\left(HM_{ip}, KPI_i\right) = \frac{a_{HM,1}}{1+\exp(\frac{HM_{ip}-b_{HM,1}}{c_{HM,1}})} \frac{a_{KPI,1}}{1+\exp(\frac{KPI_i-b_{KPI,1}}{c_{KPI,1}})} \tag{3.16}$$

In the case where $KPI_i > \nu_{KPI}$, the function s limits the overall *decrease* in KPI_i via the limitation in HM values:

$$s\left(HM_{ip}, KPI_i\right) = \frac{a_{HM,2}}{1+\exp(-\frac{HM_{ip}-b_{HM,2}}{c_{HM,2}})} \frac{a_{KPI,2}}{1+\exp(-\frac{KPI_i-b_{KPI,2}}{c_{KPI,2}})} \tag{3.17}$$

The coefficients a, b and c are adjusted so that the step-size of the cumulative increase/decrease in HM_{ij} stays between pre-defined limits [0.1,1.0]. Combining equations (3.15), (3.16) and (3.17), we can solve for ΔHM_{ip} as:

$$\Delta HM_{ip} = s\left(HM_{ip}, KPI_i\right) \frac{1}{1+\sum_{k\neq p} \frac{1}{\eta_{pk}}} \tag{3.18}$$

and then for all other eNB_j $(j \neq p)$ as:

$$\Delta HM_{ij} = \frac{\Delta HM_{ip}}{\eta_{pj}}. \tag{3.19}$$

In the auto-tuning algorithm, (3.18) et (3.19) are applied simultaneously to each couple of neighbouring eNBs whose KPIs can be traded. The network runs with that fixed HM setting for a duration of ΔT at the end of which the average KPIs are collected to calculate the new set of HMs to be used in the next iteration. The initial starting point is chosen as $HM_{ij} = 6dB$ for all geographical neighbours and $HM_{ij} = 11dB$ for all non-geographical neighbours. The complete pseudo-code for the auto-tuning algorithm is given in Annexe B.

The LTE network scenario

The auto-tuning heuristic described above has been simulated for a 45 eNB network. In contrast to the semi-dynamic simulation mode for the 2 eNB network scenario where HM setting is fixed for a simulation, here we have made simulations in the dynamic mode i.e., after every 50 seconds HM values are updated based on the mean BCR of the previous 50 seconds. The update frequency of the auto-tuning heuristic is $\Delta T = 50s$. The FTP download file size for the simulations is kept as 8Mbits. The rest of the network configuration details are kept the same as in Section 3.3.
Figure 3.22 shows the gain brought about in access probability (i.e., 1-BCR) by the optimization heuristic using auto-tuning mode as compared to the mode without auto-tuning i.e., fixed HM values (=6 dB for all eNBs).

Similarly, Figure 3.23 shows the gain brought about in the ABR of the mobiles in the network.

It is worth looking at the distribution of the HM parameters at the end of the dynamic simulations. Figures 3.24(a) to 3.26(b) depicts the histograms of the HM parameters at the end of the 16 minutes simulations for different values of the network call arrival rate λ. We can observe that for low and high values of λ, the distribution of HM parameters still have a peak at 6dB (the initial distribution). This is due to the fact that the iterative algorithm modifies the HM only when the KPI of one of the eNBs is good (bad) and the other is bad (good), but does not modify the HM when both are good (bad). Since for low (high) values of λ, most of the KPI pairs are good (bad), the HM parameters are rarely modified and the final distribution conserves its peak at 6dB.
Table 3.2 summarizes the percentage improvements in global KPIs like Access

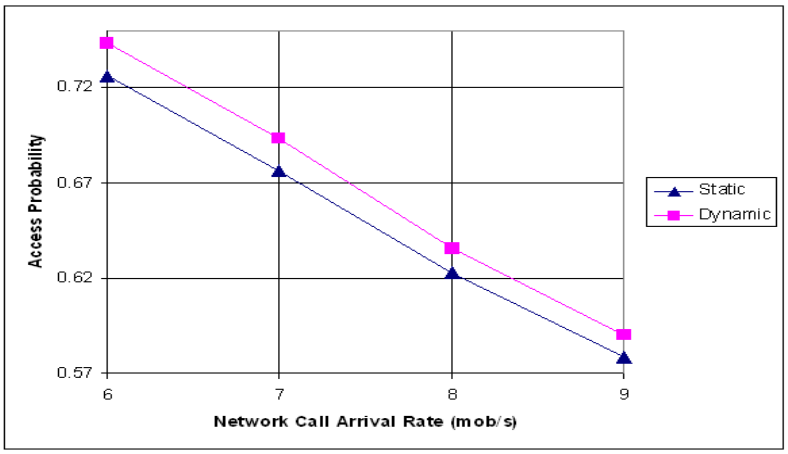


Figure 3.22: Admission probability as a function of the traffic intensity with (square) and without (triangle) autotuning.

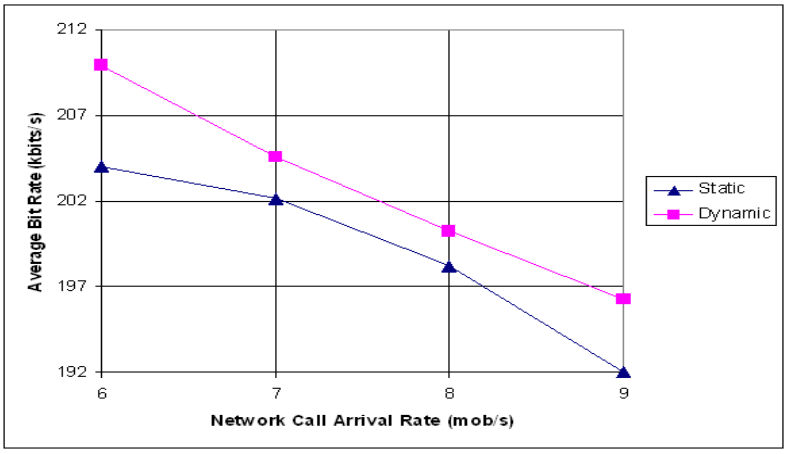
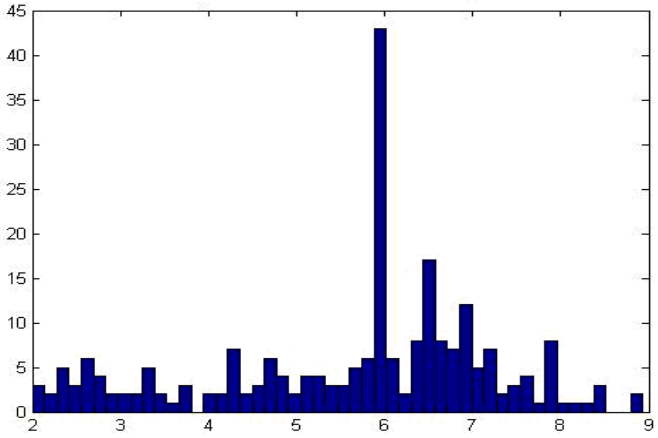


Figure 3.23: Average Bit Rate as a function of the traffic intensity with (square) and without (triangle) autotuning.

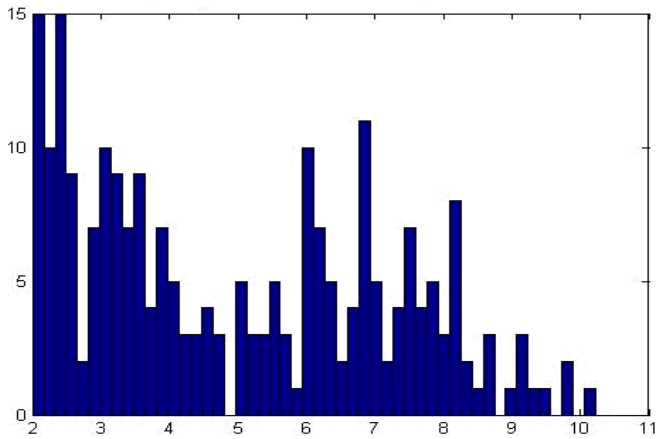
Probability (1-BCR), Maintain Probability (1-DCR) and Average Bit Rate. Note that the maximum gain that can be obtained from varying the HM parameter is in the order of 2 to 3%.

Keeping this fact in mind, we can state that the proposed iterative algorithm provides the available KPI improvement.

One should note that this simple example is based on uniform traffic. Hence only small differences are expected from one eNB to another and there is not much to improve when optimising RRM parameters. However, real networks are more heterogeneous and optimizing HM should bring higher

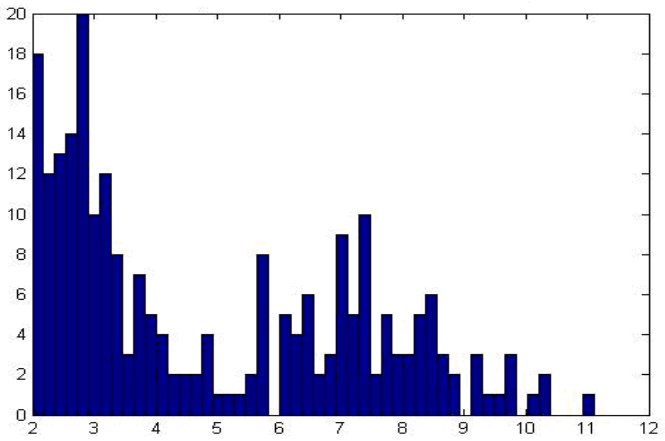


(a)

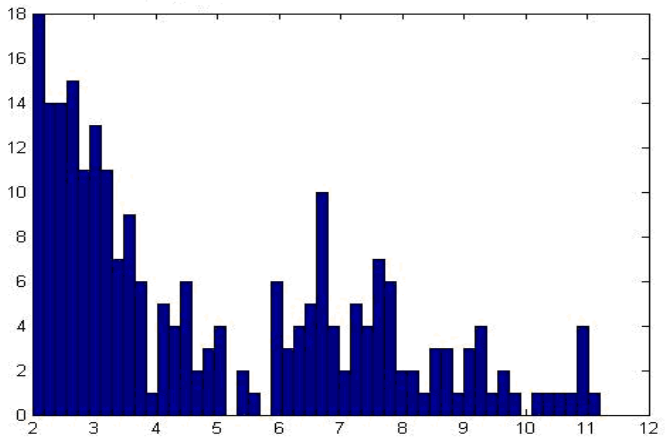


(b)

Figure 3.24: HM histograms for different arrival rates (a) λ=3 (b) λ=4.

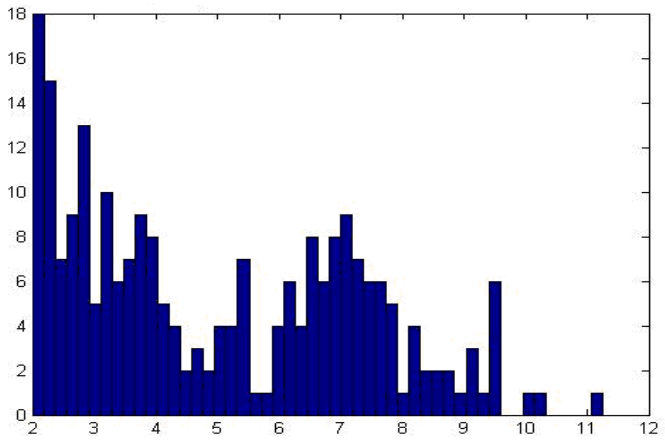


(a)

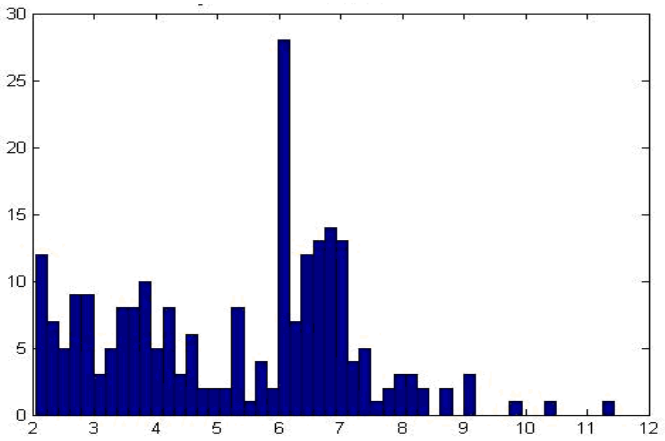


(b)

Figure 3.25: HM histograms for different arrival rates (a) λ=5 (b) λ=6.



(a)



(b)

Figure 3.26: HM histograms for different arrival rates (a) λ=7 (b) λ=8.

%	Δ Access Prob.	Δ Maintain Prob.	Δ Ave Bit Rate
$\lambda = 3$	0.51	1.58	0.99
$\lambda = 4$	0.75	0.39	2.24
$\lambda = 5$	1.53	1.29	0.62
$\lambda = 6$	2.38	0.22	2.88
$\lambda = 7$	2.53	0.90	0.22
$\lambda = 8$	2.11	1.27	1.00
$\lambda = 9$	2.04	0.38	2.22
$\lambda = 10$	1.16	1.94	0.90
$\lambda = 11$	0.35	1.36	0.31
$\lambda = 12$	1.23	0.05	1.60
$\lambda = 13$	2.84	1.30	1.07

Table 3.2: Percentage improvement provided by the dynamic scheme.

improvements in reality.

3.8 Conclusion

This chapter has presented a statistical learning approach for extracting a *model* from data with applications to monitoring, optimization and auto-tuning network functionalities. The *model* provides closed form expressions that approximate the functional relations between KPIs and RRM parameter. The *model* can help anticipating the behaviour of a network sub-system to new values of RRM parameters. A case study of eUTRAN HO algorithm has been considered. The functional relations between the HM parameter and KPIs have first been established via a regression analysis. It has been shown how the obtained *model* can be used for monitoring and auto-tuning of the network. The proposed approach can be easily extended to other problems of auto-tuning and self-healing.

Chapter 4

Automated Healing by Statistical Learning

4.1 Introduction

This chapter presents the automated healing based on statistical learning for Long Term Evolution (LTE) networks. The automated healing uses the closed-form expressions between Radio Resource Management (RRM) and Key Performance Indicator (KPI) parameters, obtained using statistical learning. The statistical learning technique used is the regression. Initially, Linear Regression (LiR) is used due to its simplicity. However, for certain cases it is not adapted and the Logistic Regression (LoR) is used because of its capability to model the saturation effects in the behaviour of KPIs corresponding to extreme RRM parameter values. The closed-form expressions obtained using regressions are termed as the *model*.

As explained earlier in the introduction, this thesis investigates the 3rd step of troubleshooting i.e., the problem solution or healing. It has been assumed that the cause of the degraded performance of a problematic eNB has been diagnosed to be due to a bad RRM value settings. This methodology aims at locally optimising the RRM parameters of the cells with poor performance in an iterative manner. The optimization process uses the *model* information from the problematic eNB and its first tier neighbours to calculate the optimized RRM parameter value. The *model* improves iteratively as new KPIs corresponding to the optimized RRM parameter value, known as the RRM-KPI pairs, are calculated during each iteration. The main advantage of this methodolgy is the small number of iterations required to achieve convergence and the QoS objective.

In the beginning of the chapter, a relatively simple automated healing

scheme based on the iterative statistical *model* refinement is presented. Later this idea is refined into an automated healing methodology known as Statistical Learning Automated Healing (SLAH) which is more scalable and efficient. SLAH is used in the healing of LTE interference mitigation, mobility and packet scheduling parameters. It is further enhanced using the *a priori* knowledge to make it more robust and practical for the operating networks.

The remainder of this chapter is organized as follows. Section 4.2 describes the generic block diagram for automated healing based on statistical learning. Section 4.3 details the automated healing algorithm based on the idea of iterative statistical *model* refinement with application to automated healing of LTE mobility parameters. Section 4.4 introduces SLAH methodology with its application to intereference mitigation in LTE using Intercell Interference Coordination (ICIC). The SLAH methodology is enhanced using the *a priori* knowledge in Section 4.5. Section 4.6 presents the application of SLAH to LTE healing using sequential modification of interference mitigation and packet scheduling parameters. The application of SLAH in order to troubleshoot LTE mobility parameters is explained in Section 4.7. Eventually, Section 4.8 concludes this chapter.

4.2 Generic automated healing block diagram

As indicated above, we assume that the fault cause has been diagnosed to be due to bad RRM parameter setting; our focus is only on the problem solution phase. The generic block diagram based upon statistical learning is presented in Figure 4.1.

As it can be seen, the system *model* is composed of four blocks:

- **Initialization block:**
  The initialization block can have either of two functions
  - switch position 1: In the absence of the *a priori* RRM-KPI pairs, it provides the initial RRM parameter values to the faulty eNB and its first tier neighbours in the Network/Simulator block in order to generate the corresponding KPIs. These generated RRM-KPI couples are used by the Statistical Learning block.
  - switch position 2: To provide noise-free *a priori* RRM-KPI pairs to Statistical Learning block for the initial *model* estimation.

  The default switch position is 1, the switch position 2 is only used in Section 4.5.

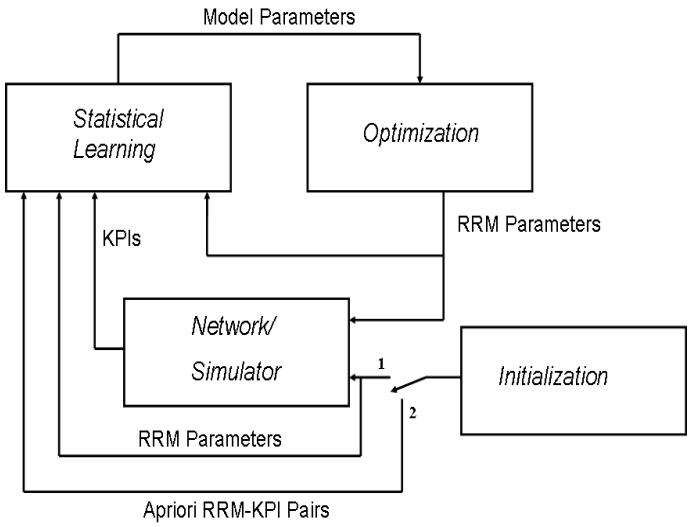


Figure 4.1: SLAH block diagram.

- **Network / Simulator block:**
  The Network/Simulator block represents the real network or the network simulator. It captures (case of real network) or calculates (case of network simulator) a set of KPIs of an eNB and of its neighbors for each new RRM parameter value introduced by the Initialization or the Optimization block. These KPIs allow to assess the performance of the eNB and of its neighbours. They are forwarded to the Statistical Learning block.

- **Statistical Learning block:**
  The Statistical Learning block processes the data comprising RRM-KPIs couples, using the statistical learning, to extract closed-form functional relationships relating the KPIs to the RRM parameters, denoted as the *model*.

- **Optimization block:**
  The Optimization block calculates the optimal RRM value using the current *model* in order to achieve the required QoS objective. The optmization objective can be achieved by maximizing a utility function or reducing a cost function.

The automated healing model assumes that the KPIs are well behaved functions, namely they are not multi-modal functions of the RRM parameter. This assumption allows to capture the functional form of the KPIs using regression techniques. The automated healing process is iterative. At each

iteration, a new RRM parameter value is proposed by the Initialization block during the initialization iterations and by the Optimization block during the optimization iterations to update the RRM setting of the eNBs of interest in the Network/Simulator block. The performance of the faulty eNB and of its neighbours with this new RRM setting is assessed by the Network/Simulator block through a set of KPI values obtained at the end of the measurement period, typically one day. And hence, a data point comprising an RRM parameter value and the corresponding KPIs is obtained. This data point together with the previously obtained data points are used by the Statistical Learning block to refine the *model* which is then used by the Optimization block to generate the RRM parameter value of the next iteration. It is noted that the data is noisy due to the random nature of the traffic and of the radio channel, but also due to imprecisions in measurements. And thus, as the iterations progress, on the average, the *model* precision improves and is used by the Optimization block to find a better value for the RRM parameter. This approach allows optimization to converge in few number of iterations and allows fast reaction to problems in the network.

4.3 Automated healing of LTE Mobility Parameters using Iterative Statistical Model Refinement

It has been assumed the fault cause of a problematic eNB or its problematic neighbours has been diagnosed as bad RRM value settings. This methodology aims at locally optimising the RRM parameters of the cells with poor performance in an iterative manner. Linear Regression (LiR) is the statistical learning used to obtain closed-form relations known as the *model*. As mentioned in the Chapter 3, the LiR is used due to its simplicity and due to the linear relationship of KPIs with LTE RRM mobility parameters. This *model* is then used by the optimization engine to calculate the optimized RRM parameters for these cells. The *model* is refined iteratively as new KPIs corresponding to the optimized RRM parameter value, known as RRM-KPI pair, is calculated during each iteration. The required QoS objective and the convergence are achieved in small number of iterations. An automated healing application scenario involving mobility in LTE networks is considered. Numerical simulations illustrate the benefits of our proposed scheme.

4.3.1 Relation to generic automated healing block diagram

Referring to the generic block diagram in Figure 4.1, all the other details are the same except the Optimization and the Statistical Learning block which need further explanation. These two blocks are explained as follows:

- **Statistical Learning block:**
  The Statistical Learning block processes the data comprising RRM-KPIs couples using the LiR method to extract the *model*. LiR is a general method for estimating/describing the association between a continuous response (dependent) variable and one predictor (explanatory variable) [91]. It has been used to write the KPI (response variable) as a function of RRM parameter x (explanatory variable) as follows:

  $$KPI_j = f_j(x) \qquad j = 1, 2, ...J \tag{4.1}$$

  where J is the number of different KPIs of interest and $f_j(x)$ is linear. LiR is chosen due to its simplicity and its validity in most engineering problems [90].

- **Optimization block:**
  The Optimization block calculates the optimal RRM value using the current statistical *model* by taking the weighted average of RRM parameter values corresponding to the maximum allowable thresholds of the KPIs under consideration. This is possible because the KPIs of an eNB and its first tier neighbouring eNBs have a conflicting tendency with respect to change in RRM parameter and also because the functional form of the relationship between the RRM parameters and KPIs, captured using LiR, is linear. Hence, the weighted average gives the RRM parameter value which satisfies the required QoS constraint. The weights are chosen with respect to the relative importance of the KPIs under consideration for automated healing. It is noted that the data is noisy due to the random nature of the traffic and of the radio channel, but also due to imprecisions in measurements.

4.3.2 Automated healing algorithm description

We now consider the adaptation of the automated healing algorithm to the mobility parameter of the LTE network by optimizing the Handover Margin (HM) parameter of the degraded eNBs. The mobility model for LTE is explained in 2.7

Let eNB_c (c standing for *central*) or its first tier neighbouring eNBs T_1 experience a degraded performance. We, again, assume that the cause of the degraded performance has been diagnosed and that it is related to a bad mobility parameter HM_i. We have: $HM_{cj} = HM_{kj} = HM_i$, $j \in T_1$ and $k \in T_2$, where T_2 are the second tier neighbours of eNB_c. Also, $HM_{jc} = HM_{jk} = HM_{max} - HM_i$, where HM_{max} is the maximum HM value.

Two KPIs are next used in the automated healing process: Block Call Rate (BCR) and Drop Call Rate (DCR). Our optimization objective is to determine the optimized HM_i as

$$HM_i = \frac{\sum_{\kappa \in \left\{BCR_c, DCR_c, \overline{BCR}, \overline{DCR}\right\}} \omega_\kappa HM_\kappa}{\sum \omega_\kappa} \tag{4.2}$$

where $\overline{BCR}$ and $\overline{DCR}$ are the mean BCR and mean DCR of T_1 eNBs, ω_κ is the weight given to KPI κ, denoted as KPI_κ and HM_κ is the HM value corresponding to the maximum allowable threshold for KPI_κ, determined using the relationship:

$$KPI_\kappa = f_\kappa(HM_i) \tag{4.3}$$

where f_κ is obtained using LiR on vector P_k consisting of k data points. The kth data point p_k is given as:

$$p_k = (HM_i, BCR_c, DCR_c, \overline{BCR}, \overline{DCR})_k \tag{4.4}$$

Now according to the requirements in [91], the initial number of data points necessary to obtain regression coefficients must be two or greater. The initial number of data points, before applying LiR and consequently optimization, is chosen as three. One extra data point (more than the minimum required data points) is chosen to compensate the effect of noise in the KPIs. The smaller the number of points, the more sensitive the regression is to noise. Furthermore, a badly estimated initial *model* of f_κ can cause the optimization problem to be erroneous and the new points found by the optimization module to get stuck in a non-optimal region.

The complete automated healing algorithm is as below:

Initialization:
1. Compute an initial set P_k of k data points by applying initially chosen HM_i values one by one to network/simulator and obtaining the corresponding KPI_κ values
Repeat until convergence:
2. Compute the statistical *model* f_κ using P_k
3. Compute the new HM_i value using (4.2)
4. Apply HM_i in the network/simulator and observe corresponding KPI_κ values. Compute new data points using (4.4)
5. Update P_{k+1} : $P_{k+1} = P_k \cup p_{k+1}$
6. *k=k+1*
End Repeat

4.3.3 Case study

4.3.3.1 Simulation scenario

A LTE network comprising 19 eNBs in a dense urban environment is depicted in Figure 4.2.

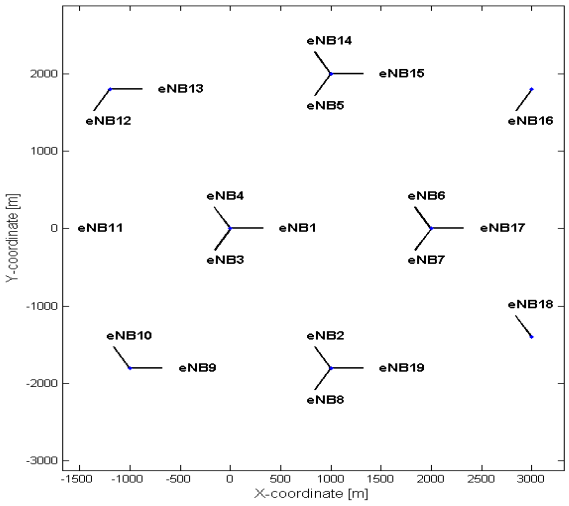


Figure 4.2: The network diagram of the simulated system.

We consider downlink transmissions. The simulation parameters are listed in Table 4.1. A MATLAB LTE simulator described in [89] has been used.

Parameters	Settings
System bandwidth	5MHz
Cell layout	19 eNBs, single sector
Maximum eNB transmit power	32 dBm
Inter-site distance	1.5 to 2 KM
Subcarrier spacing	15 kHz
PRBs per eNB	15
Path loss	L=128.1 + 37.6 $\log_{10}(R)$, R in kilometers
Thermal noise density	-173 dBm/Hz
Shadowing standard deviation	6 dB
Traffic model	FTP
File size	5700 Kbits
PRBs assigned per mobile	1 to 4 (First-come, first-serve basis)
Mobility of mobiles	90%
Mobile speed	15 m/s
HM_{max}	12dB

Table 4.1: The system level simulation parameters.

The simulator performs correlated Monte Carlo snapshots with time steps of one second to account for the time evolution of the traffic. At the end of each time step of one second, new mobile positions are updated, Handover (HO) events are processed, new users are admitted according to the conditions of access and some other users leave the network (end their communications or are dropped). The simulations are run for 3300 time steps, with a fixed HM value, and the KPIs are averaged using the interval between 500 and 3300 seconds to account for transient effects.

Reference Solution

An optimal default value for HM is chosen as 6dB for all eNBs in the network and will serve as the reference (default) solution. This reference solution will be used as a starting point for the automated healing process. The default HM value is determined by varying it uniformly from 0.05 to 12 in steps of 0.15 for all the eNBs. For each HM value, the network performance is assessed in terms of the mean Ping Pong Rate (PPR), mean BCR and mean DCR. If these three KPIs are aggregated as shown in Figure 4.3, we observe that the global optimum HM value occurs around HM =6dB. Hence, the

value of $HM = 6$dB is selected as the reference (default) HM value for eNBs in the network.

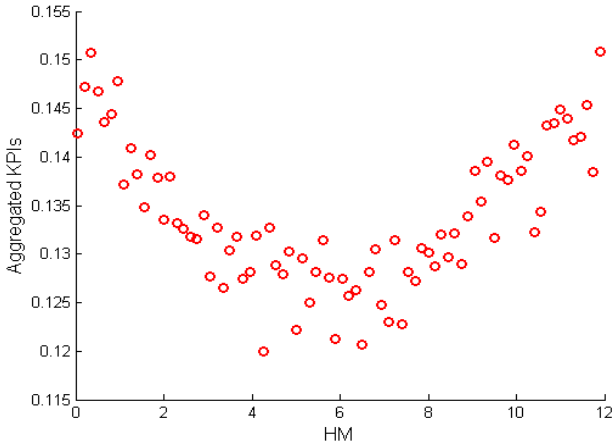


Figure 4.3: The aggregated network KPIs (mean PPR, mean BCR and mean DCR) as a function of uniform (default) HM.

4.3.3.2 Automated healing Scenario I : Equal weights to KPIs

With reference to the network with 19 eNBs shown in Figure 4.2, eNB_1 is the central eNB which we denoted by eNB_c. Its 6 first tier eNBs T_1 are given as eNB_2, eNB_3, eNB_4, eNB_5, eNB_6 and eNB_7. While the 12 second tier neighbours T_2 are eNB_8, eNB_9, eNB_{10}, eNB_{11}, eNB_{12}, eNB_{13}, eNB_{14}, eNB_{15}, eNB_{16}, eNB_{17}, eNB_{18} and eNB_{19}.

We consider the case where equal importance is given to all the KPIs, i.e., ω_t=1, $t \in \{BCR_c, DCR_c, \overline{BCR}, \overline{DCR}\}$. Initially, the system is working with default $HM_i = 6$dB. Table 4.2 shows the convergence of these parameters starting from the three initial points (Phase I), corresponding to $HM_i = 6, 4, 8$dB, and seven iterations of the optimization algorithm (Phase II). Figures 4.4, 4.5 and 4.6 show the KPIs as a function of HM_i after the first, second and third iterations, respectively. As explained earlier, the KPI curves are estimated from corresponding KPI data points using LiR.

We set the thresholds for BCR and DCR to 8% and 4%, respectively. We observe from Table 4.2 that obtained $\overline{BCR}$ and $\overline{DCR}$ values corresponding to the initial working point exceed these thresholds.

Now, we apply the automated healing algorithm and collect the initial data points corresponding to HM_i=6,4 and 8dB (Phase-I). We can see that

	HM_i(dB)	BCR_c	DCR_c	$\overline{BCR}$	$\overline{DCR}$
Phase I	6	0.040	0.016	0.82	0.041
	4	0.012	0.010	0.095	0.056
	8	0.063	0.028	0.069	0.038
Phase II	8.49	0.055	0.032	0.067	0.031
	8,60	0.062	0.023	0.065	0.031
	8.64	0.063	0.025	0.064	0.030
	8.56	0.063	0.033	0.059	0.028
	8.41	0.044	0.021	0.064	0.029
	8.47	0.068	0.024	0.064	0.035
	8.47	0.068	0.024	0.066	0.037

Table 4.2: Convergence of HM_i during optimization process and the corresponding KPIs of automated healing algorithm

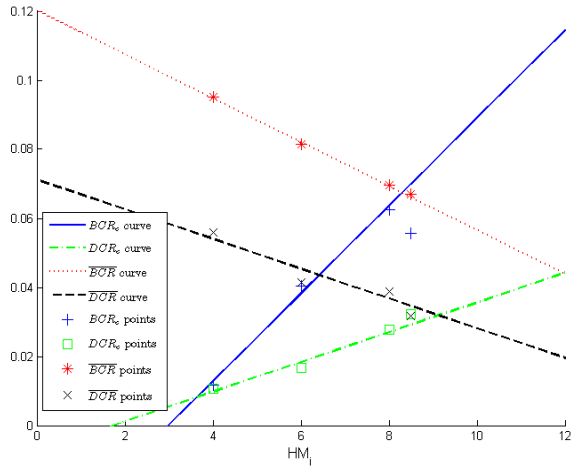


Figure 4.4: The KPIs in troublshooting as a function of HM_i after first optimization iteration.

after the first iteration of the optimization process (when phase-II starts) we get the next HM_i=8.49dB. This iteration corresponds to Figure 4.4.

After the second and third optimization iterations (shown in Figures 4.5 and 4.6, respectively), the HM_i values converges to 8.64. The corresponding $\overline{BCR}$ improves from 8.2% to 6.4% and $\overline{DCR}$ improves from 4.15% to 3%, which are within the maximum allowable thresholds of 8% and 4%, respectively. BCR_c and DCR_c of the central eNB are also within the thresholds. Eventually, we can see from Table 4.2 that even over the next 7 optimization

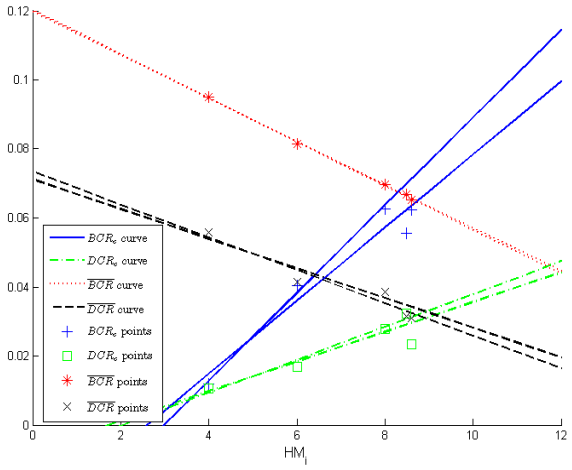


Figure 4.5: The KPIs in troublshooting as a function of HM_i after second optimization iteration.

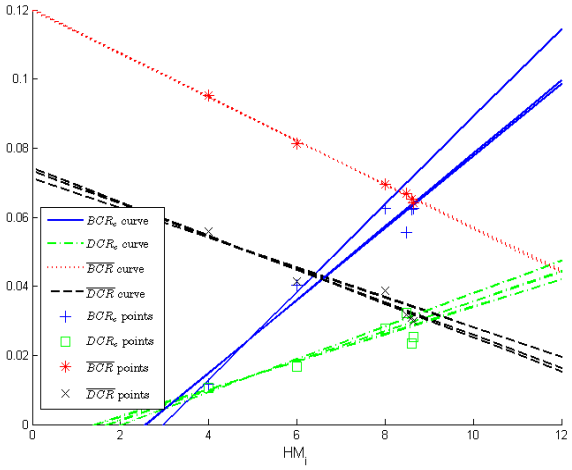


Figure 4.6: The KPIs in troublshooting as a function of HM_i after third optimization iteration.

iterations during Phase-II, there is not much variation in the HM_i value.

4.3.3.3 Automated healing Scenario II : Different weights to KPIs

The automated healing methodology is now verified using different ω_t values, $t \in \left\{BCR_c, DCR_c, \overline{BCR}, \overline{DCR}\right\}$ as given below:

1. ω_{BCR_c}=1, ω_{DCR_c}=1, $\omega_{\overline{BCR}}$=1, $\omega_{\overline{DCR}}$=1
2. ω_{BCR_c}=2, ω_{DCR_c}=1, $\omega_{\overline{BCR}}$=1, $\omega_{\overline{DCR}}$=1
3. ω_{BCR_c}=3, ω_{DCR_c}=1, $\omega_{\overline{BCR}}$=1, $\omega_{\overline{DCR}}$=1
4. ω_{BCR_c}=1, ω_{DCR_c}=1, $\omega_{\overline{BCR}}$=1, $\omega_{\overline{DCR}}$=2
5. ω_{BCR_c}=1, ω_{DCR_c}=1, $\omega_{\overline{BCR}}$=1, $\omega_{\overline{DCR}}$=3

For each set of w_t weights, we apply our automated healing algorithm to the network shown in Figure 4.2. We repeat the scenario eight times and store each time the seven optimization iterations as above.

We now take the variance between the eight HM_i value calculated during each iteration as the measure of the convergence. Figure 4.7 shows that after the third iteration, HM_i converges to an optimum fixed value.

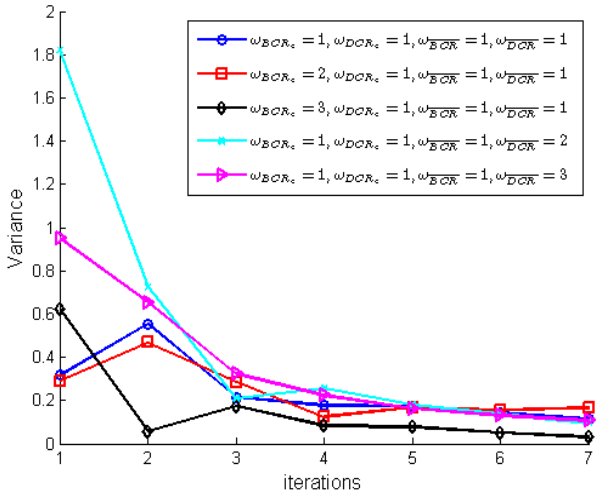


Figure 4.7: Convergence of HM_i in automated healing algorithm for different KPI weights, ω_t.

4.4 Statistical Learning in Automated Healing(SLAH): Application to LTE interference mitigation

In this section a new automated healing methodology has been introduced which we named Statistical Learning Automated Healing (SLAH). Similar to

previous section, it uses statistical learning to derive the *model*. However, the statistical learning technique used in this case is Logistic Regression (LoR). The merit of LoR is that it can account for saturation effects that often appear in KPIs such as blocking rate, dropping rate etc. This *model* is then processed by an optimization engine so as to calculate the optimized RRM parameters which improve the KPIs of a degraded cell. The process is iterative and converges to the optimum RRM parameter value in few iterations, which makes it suitable for operational wireless networks.

In fact, once the problem has been identified, there can be various ways which can be used to improve the performance of the cells that exhibit a degraded performance. The so-called *steered optimization technique* is one such method introduced in [85]. It uses an interference matrix approach to identify the most coupled base stations with the problematic cell in a cellular (UMTS) environment. Indeed, the knowledge of the interaction between any couple of stations in terms of interference, macrodiversity and load difference allows to, first, accurately identify sectors with poor performance and, second, to suggest corrective measures [86]. The method makes use of local information from an eNB and its neighbors; the resulting automated healing is thus a local optimization process.

Based on this interference matrix approach and local optimization technique, we have introduced in [4] a new automated healing methodology, which we termed Statistical Learning Automated Healing (SLAH), and which uses, as its name indicates, statistical learning to derive the functional relationships between the Radio Resource Management (RRM) parameters, for instance interference mitigation parameter, and the Key Performance Indicators (KPIs), such as Block Call Rate (BCR), file transfer time, etc.

In a typical conventional problem resolution process, the optimization expert analyzes KPIs and then proposes a new RRM parameter which is applied to the problematic eNB. The eNB operates with the new parameter during a period long enough, typically a day, to have statistically significant results that allow to assess the eNB performance. This optimization process is reiterated during several days, typically between one to two weeks. Hence a (near) optimal solution should be reached in a small number of iterations. The difficulty for devising automated healing algorithms are twofold. First, optimization heuristics often require hundreds of iterations and more time to converge. Second, measured counters and KPIs are inherently noisy. Noise can originate from limited measurement accuracy, but also from traffic fluctuations, varying propagation conditions etc. The effect of RRM parameter modification on KPIs can be masked partially by unobserved effects thus introducing uncertainty in the relation between KPIs and RRM parameters.

To overcome the above two difficulties, this novel approach for automated

healing based on statistical learning has been introduced. The automated healing algorithm is iterative. In each iteration, a new data point comprising of a RRM parameter and a corresponding KPI vector is introduced. The data point is used to update the statistical *model* and improve its precision. Then the updated statistical *model* is introduced into the optimization engine to calculate the next RRM parameter. Hence the optimization algorithm does not directly process data points but rather closed form functions in the form of regressions, allowing the *model* to converge rapidly with a small number of iterations.

In this section we assume that the degraded performance of an eNB is due to excessive intereference. The SLAH uses Inter-cell Interference Coordination (ICIC) for interference mitigation to improve the performance of the problematic eNB.

4.4.1 Relation to generic automated healing block diagram

Referring to the generic automated healing block diagram in Figure 4.1, the details of the statistical learning and optimization block are given as below:

- **Statistical Learning block:** The statistical learning approach used in the automated-healing is based on Logistic Regression (LoR) [91] [87]. The LoR Model belongs to a category of models known as the Generalized Linear Models (GLM) [97]. The LoR establishes the statistical *model* by extracting the functional relations between the KPIs and the RRM parameter. LoR fits the data into the functional form of *logistic function* denoted as f_{log}

$$f_{log}(z) = \frac{1}{1 + exp^{-z}} \tag{4.5}$$

  where z can vary from $-\infty$ to ∞ and $f_{log}(z)$ from 0 to 1 (see Figure 4.8). One can see from Figure 4.8 that $f_{log}(z)$ can describe saturation effects at its extremities as often encountered in KPIs in communication networks.
  In our work, the KPI is a dependent variable denoted as y and the RRM parameter is an explanatory variable denoted as x.
  Let $y_{m,i}$ denote the i^{th} sample value of the m^{th} dependent variable y_m (i.e. the m^{th} KPI) corresponding to the i^{th} sample value x_i of the explanatory variable x (i.e. the RRM parameter). LoR models $y_{m,i}$ as follows:

$$y_{m,i} = f_{log}(\eta_{m,i}) + \epsilon_i \tag{4.6}$$

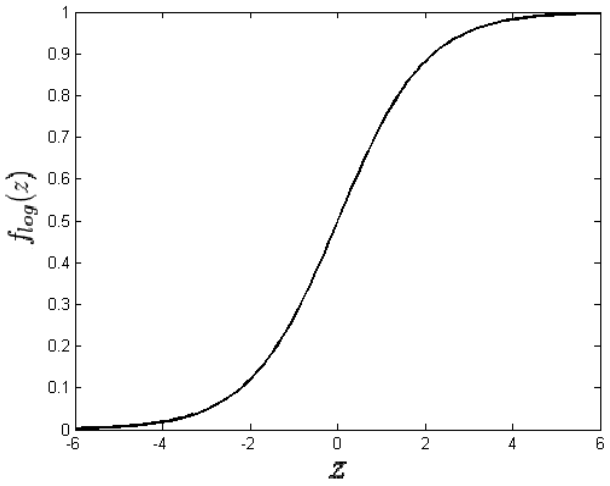


Figure 4.8: The logistic function.

where $\eta_{m,i} = x_i'\beta_m$, given that $x_i' = [1\ x_i]$ and $\beta_m = [\beta_{m,0}\ \beta_{m,1}]^T$. $\eta_{m,i}$ is the linear predictor representing the contribution of the explanatory variable sample x_i, ϵ_i is the residual error and βs are the regression coefficients whose values are estimated using maximum likelihood estimation as described in Section 4.5.2. Let there be n sample values for y_m corresponding to n values of x. In order to write (4.6) in the matrix notation, let $Y_m = [y_{m,1}\, y_{m,2}\, ..y_{m,i}..\, y_{m,n}]^T$ and $X = [x_1'\, x_2'\, ..x_i'..\, x_n']^T$. Hence, we may write

$$\begin{aligned} Y_m &= f_{log}(X\beta_m) + \epsilon \\ Y_m &= f_{log}(\eta_m) + \epsilon \\ Y_m &= \hat{Y_m} + \epsilon \end{aligned}$$

where

$$\hat{Y_m} = f_{log}(X\beta_m) \tag{4.7}$$

Here, $\hat{Y_m}$ represents the estimated Y_m using LoR. It is noted that the dependent variables in the LoR are between 0 and 1. Hence, Y_m is normalized between 0 and 1 before the application of LoR. The calculated $\hat{Y_m}$ values can be subsequently denormalized.
It follows from (4.7) that functional relation between $\hat{y_m}$ i.e., y_m estimated by LoR, and x can be written as

$$\hat{y}_m = f_{log}(x'\beta_m) \tag{4.8}$$

where $x' = [1\ x]$.

- **Optimization:** The aim of the optimization problem is to determine $\hat{x}$ i.e., the value for a RRM parameter x that minimizes a cost function of a set of KPIs denoted as the *optimization set* A_o, subject to constraints on a second set of KPIs denoted as the *constraint set* A_c. The cost function U is given as

$$U = \sum_{m \in A_o} w_m \hat{y}_m \tag{4.9}$$

where

  - $\hat{y}_m$ has the functional relation form as in (4.8).
  - w_m is the weight given to $\hat{y}_m$.

The optimization problem is formulated as follows:

$$\hat{x} = argmin_x U(x) \tag{4.10}$$

subject to

$$\hat{y}_h(\hat{x}) < th_h \ \forall \ h \in A_c$$

where th_h is the threshold for $\hat{y}_h$.

4.4.2 Application of SLAH for LTE intereference mitigation

This section presents the interference mitigation scheme that will be considered in the framework of automated healing. LTE wireless networks use OFDMA access technology in the downlink. This access technology allows to suppress intra-cell interference while it may be vulnerable to inter-cell interference, particularly for users in the cell edge. Different techniques have been developed to combat inter-cell interference. We considered in this section a soft-frequency reuse scheme for interference mitigation, which is an inter-cell interference coordination (ICIC) scheme. In OFDMA, users are multiplexed in both time and frequency. The smallest time-frequency resource unit allocated to a user is the physical resource block (PRB) with 1 ms time duration and 180 kHz bandwidth. Intelligent allocation strategies can be combined with ICIC scheme to further improve its performance [82].

4.4.2.1 ICIC model

LTE system uses the Orthogonal Frequency Division Multiple Access (OFDMA) [81] for the downlink. This access scheme has the merit of reducing the intracell interference. Intercell interference may occur when the same time-frequency resources are allocated to users in neighboring cells. Edge cell users are often the most vulnerable to such interference that can considerable degrade their quality of service (QoS). To combat intercell intereference, the intercell intereference coordination scheme (ICIC) has been proposed as an intereference mitigation technique [82].

Consider an OFDMA network with eNBs implementing ICIC in the downlink. The ICIC is performed by combining two resource allocation mechanisms: PRB allocation to frequency subbands and coordinated power allocation. In the soft-reuse one scheme, the total available bandwidth is reused in all the cells while the transmitted power for a portion of the bandwidth of a cell can be adapted to resolve interference related QoS problems. Figure 4.9 presents the power-frequency allocation model in a seven adjacent cell layout.

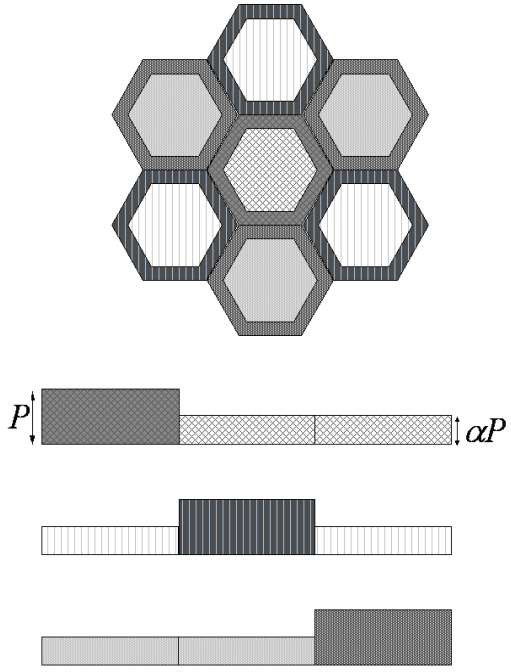


Figure 4.9: System Model.

The frequency band is divided into three disjoint sub-bands. One subband is allocated to mobiles with the worst signal quality and is denoted

interchangeably as a protected band or as an edge band with transmit power P. A user with poor radio conditions is often situated at the cell edge, but could also be closer to the base station and experience deep shadow fading. The remaining two frequency subbands are denoted as centre bands with transmit power reduced by a factor α, namely αP. The interference produced by an eNB to its neighboring eNBs can be controlled by the parameter α of this eNB. The main interference in the system originates from eNB transmissions on the centre band (of centre cell users) which interfere with neighboring cell edge users utilizing their edge (protected) band. When an eNB strongly interferes with its neighbours, the ICIC mechanism allows to reduce the transmission power for the centre band.

Resource block allocation is performed based on a priority scheme for accessing the protected subbands. A quality metric q_u is calculated using pilot channel signal strengths

$$q_u = \frac{Pr_{su}}{\sum\limits_{j \neq s} Pr_{ju} + \sigma_z^2}, \tag{4.11}$$

where Pr_{ju} denotes the mean pilot power received by the user u of a signal transmitted by the eNB j, and σ_z^2 is the noise power spectral density. q_u is similar to the SINR with the difference that in the present ICIC scheme, the data channels used to calculate the SINR are subject to power control. The q_u metric is calculated for all users which are then sorted according to this metric. Users with the worst q_u are allocated resources from the protected band and benefit from maximal transmission power of the eNB. When the protected subband is full, the resource block allocation continues from the centre band.

4.4.2.2 Automated healing model

This section describes the adaptation of the SLAH to interference mitigation in a LTE network by locally optimizing the parameter α of the interfering eNBs. Denote by eNB_c (c standing for *central*) an eNB with degraded performance. It is assumed that the cause of the degraded performance has been diagnosed and is related to excess interference from neighbouring eNBs. The direct neighbours from the first tier of eNB_c are denoted by eNB_j, $j \in NS1(c)$ where $NS1(c)$ is the set of neighbouring eNBs in first tier of eNB_c. The subscript c of the set $NS1(c)$ will be omitted hereafter for sake of brevity. The specificity of the interference mitigation use case is the following: to troubleshoot eNB_c the parameters α_j of eNB_j, $j \in NS1$, are updated and optimised, while α_c of eNB_c remains unchanged.

We use the notion of coupling between eNB j and c which is expressed in terms of the interference that eNB_j produces on the mobiles connected to eNB_c and can be written in terms of the interference matrix element I_{cj} [85] [96]. Hence the bigger I_{cj}, the stronger the coupling between the two eNBs. In this work, the matrix element I_{cj} is equal to the sum of interferences perceived by the mobiles attached to eNB_c and generated by downlink transmissions to the mobiles of eNB_j.

The use of the SLAH to jointly optimize all the elements of the vector (α_j), $j \in NS1$, is not a simple task. Denote by s, $s \in NS1$, the index of the eNB which is the most coupled with eNB_c, namely $s = argmax_j(I_{cj})$, $s \in NS1$. To reduce the complexity of the SLAH process and to enhance its scalability, we propose to adjust the α_j parameter according to the degree of coupling between eNB_j and eNB_c. To this end, we define a functional relation between α_s and α_j, $\alpha_j = \mathrm{g}_j(\alpha_s)$, that accounts for the coupling mentioned above:

$$\alpha_j = \mathrm{g}_j(\alpha_s) = \alpha_i + (1 - \alpha_s)(1 - \frac{I_{cj}}{I_{cs}}) \tag{4.12}$$

Hence the smaller the coupling between eNB_j and eNB_c, the lesser power reduction is applied to eNB_j. By using (4.12), just one parameter i.e., α_s needs to be optimized. The process is scalable in the sense that the automated healing can be performed simultaneously on any number of eNBs provided they are not direct neighbours.
Two KPIs are utilized in the SLAH process: the File Transfer Time (FTT) for FTP traffic and the Block Call Rate (BCR). The SLAH aims at minimizing the FTT for eNB_c and of its direct neighbours while verifying constraints on BCR_j, $j \in c \cup NS$. We define the cost function for the optimization:

$$U = FTT_c + \sum_{j \in NS1} \omega_j FTT_j \tag{4.13}$$

It is noted that FTT_j is a function of α_j and hence, via equation (4.12), of α_s. FTT_j also depends on the interference from its neighbouring eNB_j. The weighting coefficients ω_j depend on the relative contribution of I_{cj} with respect to the sum on all eNBs in $NS1$ and are given by

$$\omega_j = \frac{I_{cj}}{\sum_{k \in NS1} I_{ck}} \tag{4.14}$$

satisfying the condition $\sum_{j \in NS1} \omega_j = 1$. The optimization problem can now be formulated as follows

$$\alpha_s^* = argmin_{\alpha_s} U(\alpha_s) \tag{4.15}$$

subject to

$$BCR_j < BCR_{th} \ ; \ j \in c \cup NS1$$

BCR_{th} is the threshold for BCR_j. The FTT and BCR indicators in equations (4.13) and (4.15) are given in the form of the LoR function (4.8) obtained using the LoR. In the case, $BCR_j > BCR_{th}$ for all values of α_s, it is impossible to determine α_s value that minimizes the cost function (4.15). In this case, instead of (4.15) our optimization objective is given as

$$\alpha_s = argmax_{\alpha'_s}(BCR_c|(BCR_c = BCR_j)) \ ; \ j \in NS1 \tag{4.16}$$

Here, α_s signifies the value for which BCR of eNB_c and the BCR of eNB_j having worst BCR value, become equal.
The SLAH can be further improved by introducing a generalized interference matrix element I'_{cj} in equation (4.12) by introducing additional KPI, namely the BCR:

$$I'_{cj} = I_{cj}e^{-\gamma B_j} \ ; \ B_j = BCR_j/max_{k \in NS1}(BCR_k) \tag{4.17}$$

One can see that higher BCR_j is, the smaller is I'_{cj} and consequently, the smaller is the modification of α_j. This signifies that change in α_j due to I_{cj}, in order to improve performance of eNB_c, is limited by degradation in B_j, where the magnitude of effect of B_j is tuned by γ. Denote a *data point* p^j_k as the vector $p^j_k = (\alpha_j, FTT_j, BCR_j)_k$, where $j \in c \cup NS1$.
In the SLAH algorithm, α_c remains fixed and is not subject to optimization; α_j, $j \in NS1$, satisfies the equation (4.12); and the set of k data points for an eNB_j, $j \in c \cup NS1$, is denoted by P^j_k. During each optimization iteration one data point is added, hence, k also denotes optimization iteration index. The SLAH algorithm is given in Table 4.3.

4.4.3 Case study

4.4.3.1 Simulation scenario

A LTE network comprising 45 eNBs in a dense urban area is shown in Figure 4.10. Downlink transmissions are considered here. A MATLAB LTE simulator described in [89] has been used. The simulator performs correlated Monte Carlo snapshots with time resolution of a second to account for the time evolution of the network. The principles of a semi-dynamic simulator are described in [100]. FTP traffic with a file size of 6300 Kbits for download is considered. Call arrivals are generated using the Poisson process and the

Initialization:
1. Identify the most coupled eNB eNB_s with eNB_c among the neighbours in $NS1$
2. Generate the initial set of k data points P_k^j, $j \in c \cup NS1$, by applying k different α_s values (together with the associated α_j values) to the network/simulator one by one and obtaining the corresponding KPIs.
Repeat until convergence:
3. For each eNB_j, compute the statistical *model* using LoR for FTT and BCR using the corresponding data points in P_k^j
4. Compute a new α vector containing the new values of α_j, $j \in NS1$ (using equations (4.12) and (4.15))
5. Apply the new α_j values to the network/simulator and observe (FTT_j) and (BCR_j), $j \in c \cup NS1$. Compute the new data point p_{k+1}^j
6. Update P_{k+1}^j: $P_{k+1}^j = P_k^j \cup p_{k+1}^j$
7. k=k+1
End Repeat

Table 4.3: The complete SLAH Algorithm

communication duration of each user depends on its bit rate. The Okumura-Hata propagation model is used with path loss at a reference distance of 1 km and the path loss exponent are chosen as -128 dB and 3.76 respectively. The standard deviation of the shadowing process is taken as 6 dB. The bandwidth used is 5 MHz per eNB. The maximum number of the PRBs in an eNB, i.e. the capacity, is fixed to 24 PRBs with 8 PRBs in each sub-band. The number of PRBs that can be allocated to a user can vary from 1 to 4. The resources are allocated on the first-come first-served basis. Mobiles are considered in the simulation as non-mobile. The BCR and FTT KPIs used by the SLAH algorithm are averaged on an interval varying from 500 to 2500 seconds while discarding the samples of first 500 seconds during which the network reaches a steady state. It is noted that for a given traffic demand, the BCR provides a capacity indicator while the FTT is more related to the user perceived QoS.

The simulated LTE system includes a simple admission control process based on signal strength: A simple admission control has been implemented based on signal strength. A mobile selects the eNB with the highest Reference Signal Received Power (RSRP) and is admitted if it is above -104 dBm and if at least one PRB is available. The mobile throughput is calculated from SINR using quality tables obtained from link level simulations.

The SINR and consequently the bit rate of a mobile are updated after

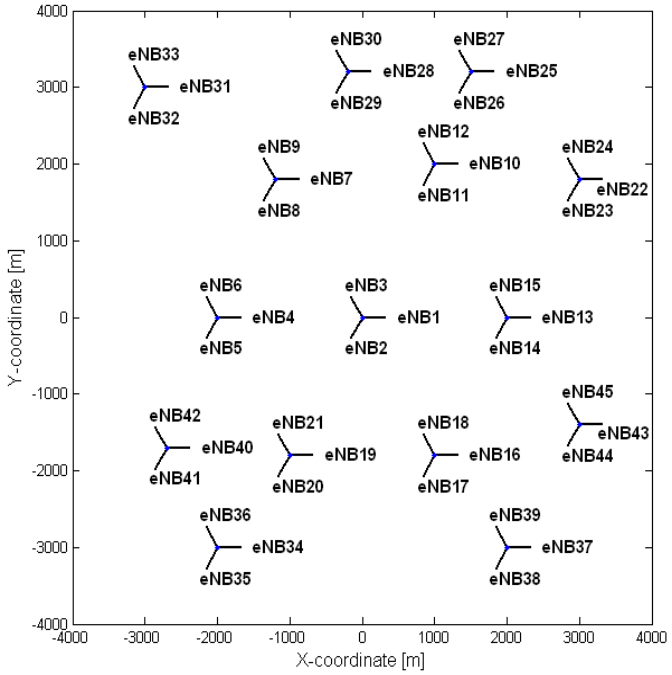


Figure 4.10: The network diagram of the simulated system.

each simulation time step. The interference matrix elements are calculated only once for the reference solution (see paragraph below) during a longer time interval varying from 500 to 7000 seconds to achieve accurate average results.
An optimal default value for α, known as *reference solution*, is calculated as 0.5 for all eNBs in the network. The default α value is determined by varying it simultaneously for all eNBs from 0.0125 to 1 in steps of 0.0125. For each α, the network performance is assessed in terms of the mean BCR and mean FTT as shown in Figures 4.11 and 4.12 respectively. The minimum values for both BCR and FTT are obtained in the α interval $[0.5, 0.7]$. The value of $\alpha = 0.5$ is selected as the default value due to the smaller inter-cellular interference and the minimum energy consumption in the network.

4.4.3.2 Automated healing scenario

A problematic eNB with the worst performance in the simulated network (in terms of BCR and FTT), namely $eNB_{c=13}$, is selected for automated healing using the SLAH algorithm. The eNB_j, where $j \in NS1 = \{14, 15, 22, 23, 43, 45\}$, is one of the six first tier neighbours of $eNB_{c=13}$. It is recalled that the SLAH modifies the α parameters of eNB_j while leaving unchanged α_c, that is fixed

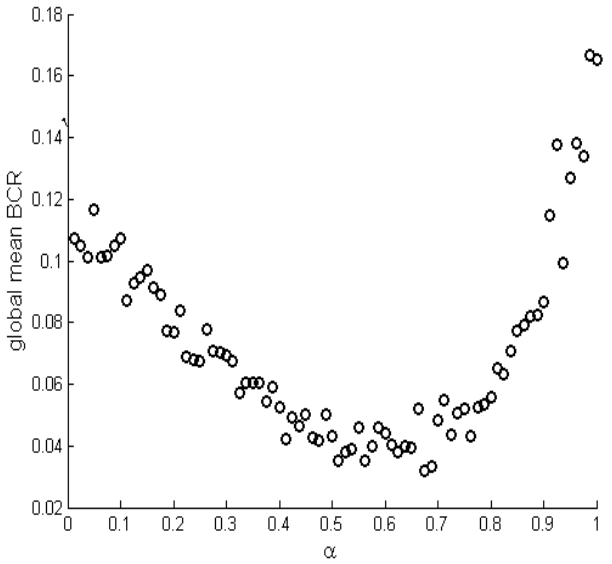


Figure 4.11: The variation of the global mean BCR of the network with uniform change in α of all eNBs.

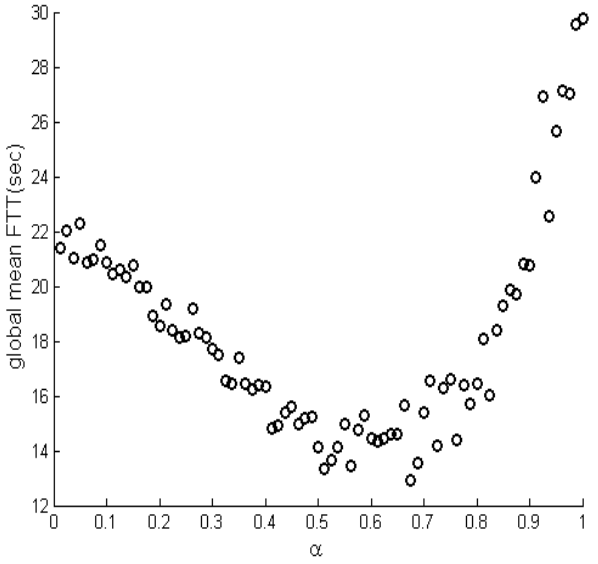


Figure 4.12: The variation of the global mean FTT of the network with uniform change in α of all the eNBs.

	$\alpha_{c=13}$	$\alpha_{j=14}$	$\alpha_{j=15}$	$\alpha_{j=22}$	$\alpha_{j=23}$	$\alpha_{j=43}$	$\alpha_{s=45}$
Phase I	0.50	0.97	0.95	0.98	0.99	0.96	0.95
	0.50	0.85	0.74	0.87	0.97	0.79	0.73
	0.50	0.74	0.53	0.77	0.94	0.61	0.50
	0.50	0.62	0.31	0.67	0.92	0.43	0.28
	0.50	0.50	0.10	0.57	0.89	0.26	0.05
Phase II	0.50	0.61	0.31	0.67	0.92	0.43	0.27
	0.50	0.66	0.38	0.70	0.93	0.49	0.35
	0.50	0.68	0.43	0.73	0.93	0.53	0.40
	0.50	0.70	0.47	0.74	0.94	0.53	0.44
	0.50	0.70	0.47	0.74	0.94	0.56	0.44
	0.50	0.72	0.50	0.76	0.94	0.59	0.47
	0.50	0.72	0.49	0.75	0.94	0.58	0.46

Table 4.4: Phase-I shows the initially chosen α values. Phase-II shows the α values calculated during optimization (γ=-0.3).

to the reference default value of 0.5. The index set $NS2$ of the second tier neighbours of the problematic eNB consists of $NS2 = \{1, 10, 11, 16, 18, 24, 37, 44\}$. Denote by *optimization zone* the subnetwork comprising $eNB_{c=13}$ and its first tier neighbours $NS1$, and by *evaluation zone* the subnetwork comprising $eNB_{c=13}$ and its first two tier neighbours $NS1$ and $NS2$. The $eNB_{s=45}$ is the eNB most coupled with $eNB_{c=13}$.

4.4.3.3 Results

The SLAH algorithm is applied using the generalized interference matrix (4.17) in (4.12), with γ=-0.3. The first five values of $\alpha_{s=45}$ in Table 4.4 are chosen for the initialization phase (Phase-I in the Table) of the SLAH. The next seven values are calculated iteratively by the SLAH algorithm during the optimization phase (Phase-II in the Table). The values of $\alpha_{j=14}$, $\alpha_{j=15}$, $\alpha_{j=22}$, $\alpha_{j=23}$ and $\alpha_{j=43}$ are calculated using equation (4.12). In spite of the inherent noise present in the generated data, one can see from the values depicted in Phase-II that $\alpha_{s=45}$ converges in a few iterations. $\alpha_{s=45} = 0.46$ is chosen as the optimized solution.

Figures 4.13 and 4.14(a) show the mean BCR and FTT data points respectively as a function of $\alpha_{s=45}$ together with the LoR curves for $eNB_{c=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ of $NS1$. The mean BCR curves for $eNB_{j=14}$, $eNB_{j=15}$, $eNB_{j=23}$ and $eNB_{s=45}$ of $NS1$ are well below the BCR threshold of 0.05 and are therefore omitted. The concentration of KPI data points around $\alpha_{s=45} = 0.45$ indicates the convergence of the SLAH algorithm. Figure 4.15(a) shows the gain brought about by the SLAH algorithm for the optimization zone (set $NS1$ of eNBs). The mean BCR of the problematic $eNB_{c=13}$ is reduced by 45% with respect to the reference solution. It decreases from a value of 5.28% to 2.9%, which is well below the threshold of

5%. The average improvement of the mean BCR of the tier $NS1$ is of 44% with respect to the reference solution.

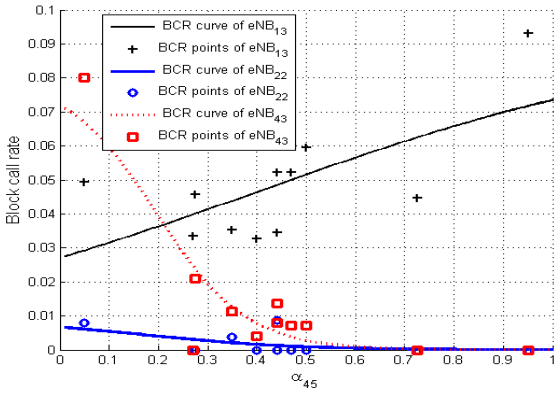
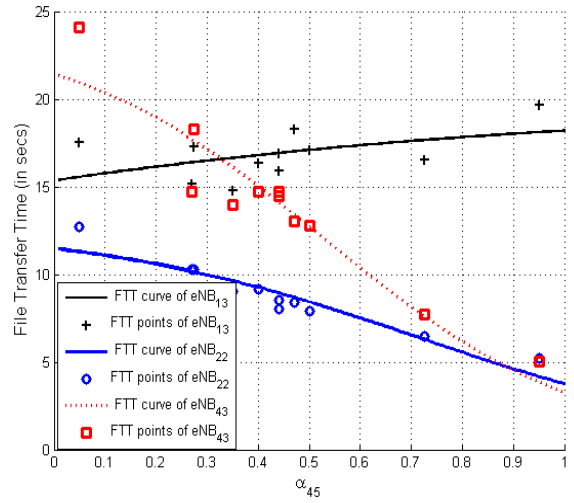


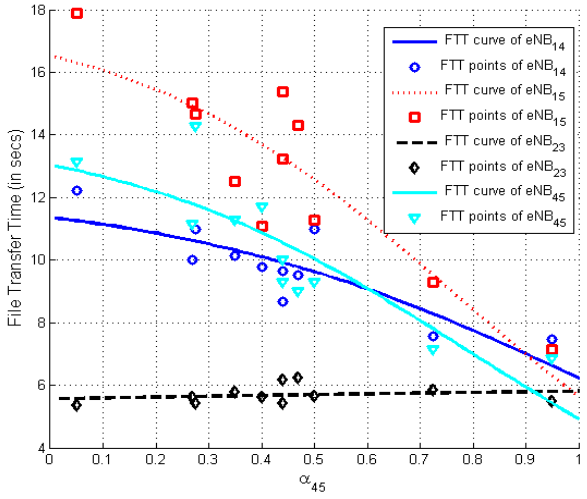
Figure 4.13: Mean BCR values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ (γ=-0.3).

The data points for the mean FTT and the corresponding LR curves are shown in 4.14. The improvement brought about by the SLAH algorithm in the optimized zone with respect to the reference solution is shown in Figure 4.15(b). The mean FTT of $eNB_{c=13}$ is reduced by 6.31% whereas the average mean FTT of $NS1$ is reduced by 26.6%. This improvement is related to the optimized interference management in the first tier of the problematic eNB. The decrease in interferences improves the SINR values and consequently the throughput and FTT. Furthermore, the improvement in power resource allocation decreases the sojourn time of users that monopolize scarce radio resources and results in the improvement in BCR.

Figures 4.16(a) and 4.16(b) show, in descending order, the mean BCR and the mean FTT respectively for the reference (square) and the optimized (circle) eNBs in the evaluation zone ($eNB_{c=13} \cup NS1 \cup NS2$). It is noted that the order of the stations in the two curves of each figure may not be preserved. One can see that on the average, the mean BCR and mean FTT in the evaluation zone are improved. The average improvement of FTT in the evaluation zone is of 13%.

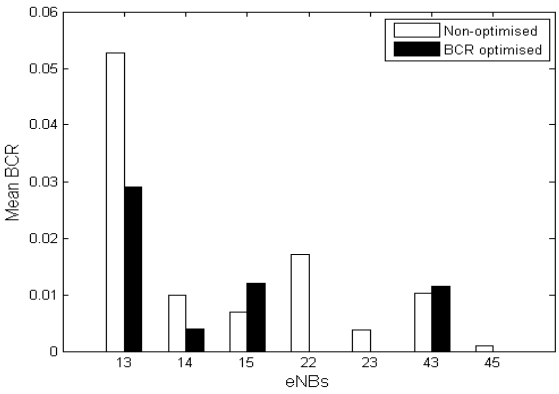


(a)

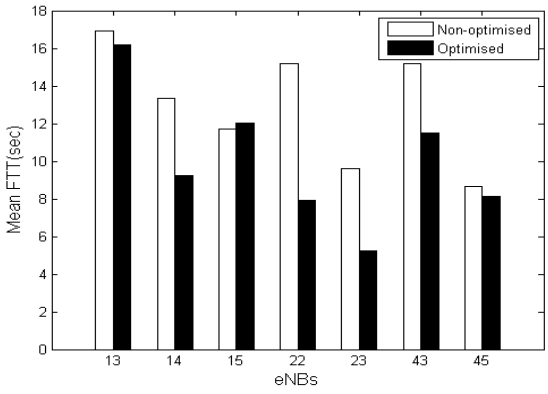


(b)

Figure 4.14: Mean FTT values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ (a) and for $eNB_{j=14}$, $eNB_{j=15}$, $eNB_{j=23}$ and $eNB_{j=45}$ (b) (γ=-0.3).

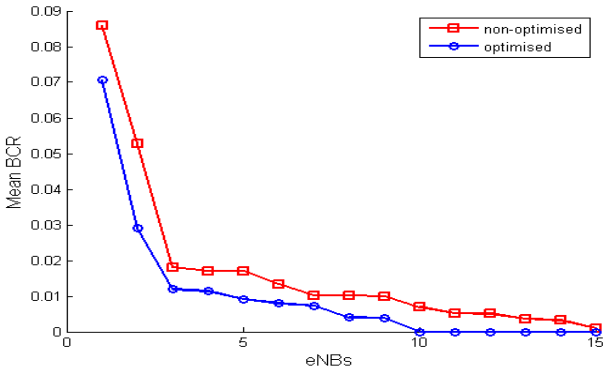


(a)

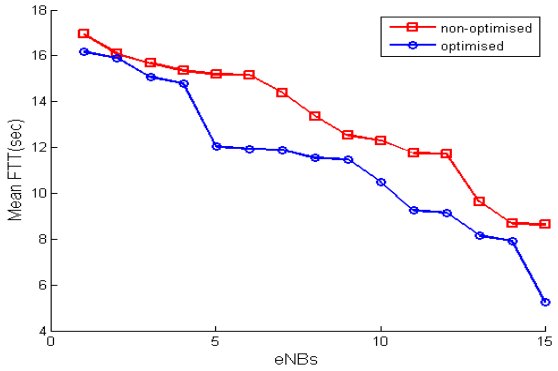


(b)

Figure 4.15: KPI of the eNBs in the optimization zone for the reference solution (white) and optimized (black) network conditions; mean BCR (a) and mean FTT (b) (γ=-0.3).



(a)



(b)

Figure 4.16: KPIs in descending order for the eNBs in the evaluation zone; mean BCR (a) and mean FTT (b) (γ=-0.3).

	$\alpha_{c=13}$	$\alpha_{j=14}$	$\alpha_{j=15}$	$\alpha_{j=22}$	$\alpha_{j=23}$	$\alpha_{j=43}$	$\alpha_{s=45}$
Phase I	0,50	0,98	0,96	0,98	0,99	0,97	0,95
	0,50	0,87	0,76	0,91	0,97	0,82	0,73
	0,50	0,77	0,56	0,83	0,94	0,68	0,50
	0,50	0,66	0,37	0,75	0,92	0,54	0,28
	0,50	0,56	0,17	0,67	0,89	0,39	0,05
Phase II	0,50	0,80	0,63	0,85	0,95	0,73	0,57
	0,50	0,80	0,63	0,85	0,95	0,73	0,57
	0,50	0,87	0,75	0,90	0,97	0,81	0,71
	0,50	0,83	0,68	0,87	0,96	0,76	0,63
	0,50	0,81	0,64	0,86	0,95	0,74	0,59
	0,50	0,80	0,62	0,85	0,95	0,72	0,56

Table 4.5: Phase-I shows the initially chosen α values. Phase-II shows the α values calculated during optimization (γ=0).

The convergence of $\alpha_{s=45}$ in the automated healing scenario with $\gamma = 0$ is shown in table 4.5. During the phase-II, the value of $\alpha_{s=45}$ converges in about 6 optimization iterations to the value of 0.57. During the process of convergence, there is one outlier value of $\alpha_{s=45}$=0.71, generated due to the noise in the KPIs. However, this method is very robust and $\alpha_{s=45}$ converges back very quickly to the value of 0.56.

Figures 4.17 shows , after convergence, the mean BCR data points as a function of $\alpha_{s=45}$ together with the LoR curves for $eNB_{c=13}$, $eNB_{j=15}$ and $eNB_{j=43}$ of $NS1$, in the case of $\gamma = 0$. The mean BCR curves for $eNB_{j=14}$, $eNB_{j=22}$, $eNB_{j=23}$ and $eNB_{s=45}$ of $NS1$ are well below the maximum allowable BCR threshold of 0.05 and are therefore omitted. Figure 4.18(a) and 4.18(b) show mean FTT data points as a function of $\alpha_{s=45}$ together with the LoR curves for $NS1$.

Table 4.5 shows that as compared to the case when γ=-0.3 (for example during phase-I in table 4.4), here the decrease of α_js corresponding to the decrease in $\alpha_{s=45}$ is less. This results in less effect on the values of a KPI of $eNB_{c=13}$ during the optimization case. Consequently, there is smaller correlation among these values that results in an increase in their variance and the possibility of having outliers.

Figure 4.19(a) shows that there is a 26.89% improvement in the mean BCR of $eNB_{c=13}$. There is an overall improvement of 75.82% in the mean BCR of $NS1$. This higher improvement in the mean BCR of $NS1$ as compared to the case when γ=-0.3, can be attributed to the fact that here the proposed α_j values are higher as compared to the case when $\gamma = -0.3$.

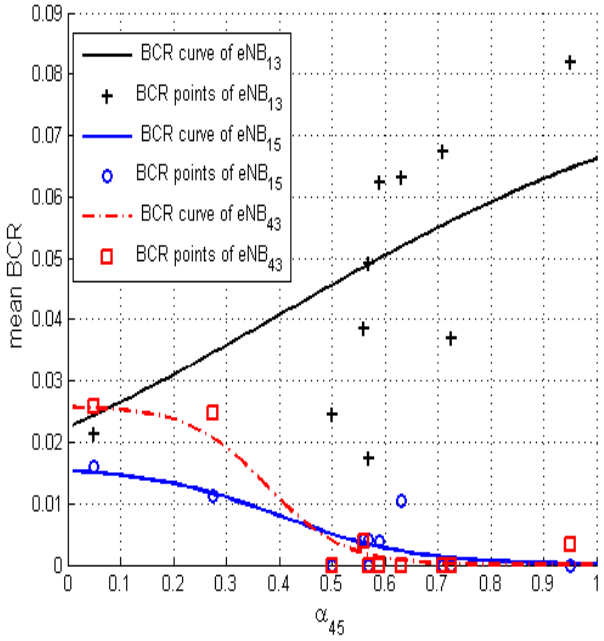
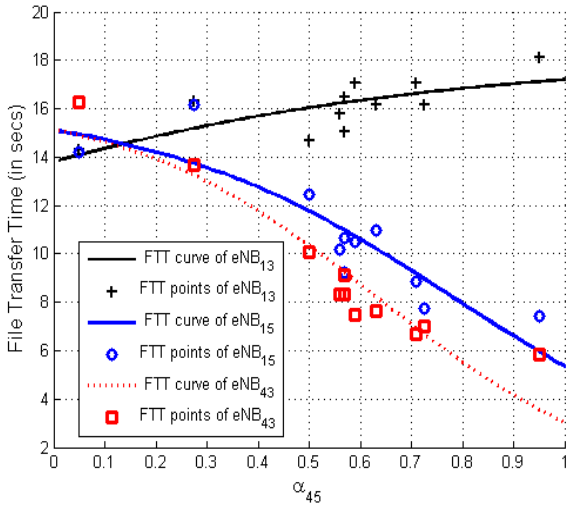
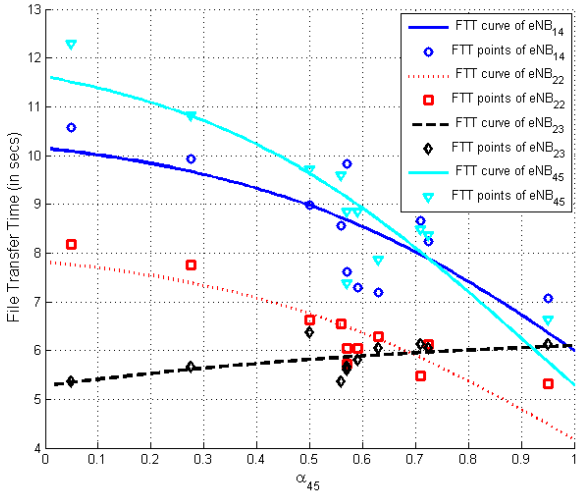


Figure 4.17: Mean BCR values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=15}$ and $eNB_{j=43}$ (γ=0).

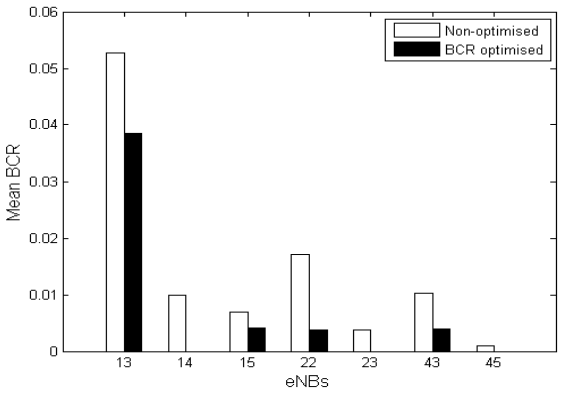


(a)

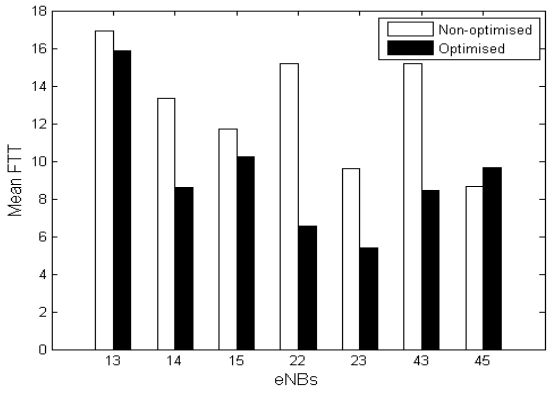


(b)

Figure 4.18: Mean FTT values and LoR regression curves as a function of $\alpha_{s=45}$ for $eNB_{j=13}$, $eNB_{j=15}$ and $eNB_{j=43}$ (a) and for $eNB_{j=14}$, $eNB_{j=22}$, $eNB_{j=23}$ and $eNB_{j=45}$ (b) (γ=0).

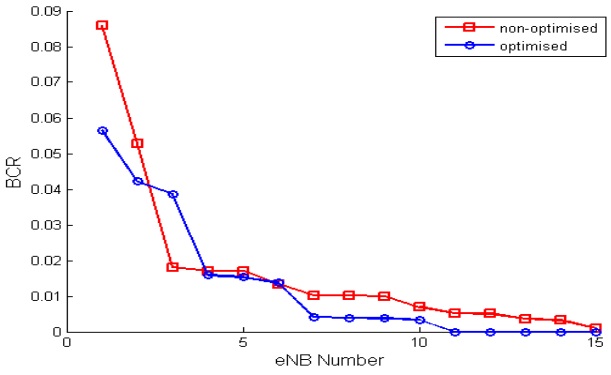


(a)

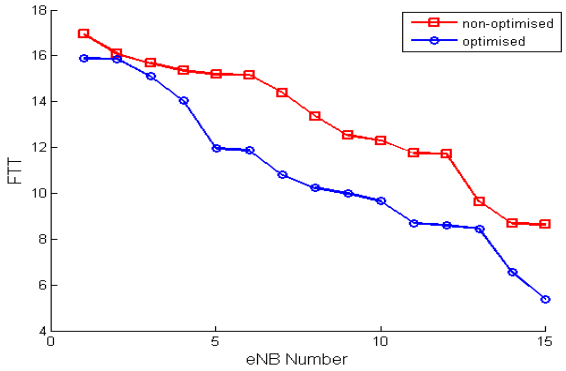


(b)

Figure 4.19: KPI of the eNBs in the optimization zone for the reference solution (white) and optimized (black) network conditions; mean BCR (a) and mean FTT (b) (γ=0).



(a)



(b)

Figure 4.20: KPIs in descending order for the eNBs in the evaluation zone; mean BCR (a) and mean FTT (b) (γ=0).

Similarly, Figure 4.19(b) shows an improvement of 6.34% improvement in mean FTT of eNB_{13}. There is an overall improvement of 33.70% in mean FTT of $NS1$, in the case of (γ=0). This improvement is higher than the case when γ=-0.3, according to same reason as for mean BCR of $NS1$.

Figures 4.20(a) and 4.20(b) show, in descending order, the mean BCR and the mean FTT respectively for the reference (square) and the optimized (circle) eNBs in the evaluation zone ($eNB_{c=13} \cup NS1 \cup NS2$) in the case $\gamma = 0$. It is noted that the order of the stations in the two curves of each Figure may not be preserved.

One can see that on the average, the mean BCR and mean FTT in the evaluation zone are improved. The average improvement of FTT in the evaluation zone is of 17%

4.5 Enhancing RRM optimization using a priori knowledge for automated troubleshooting

This section deals with the incorporation of *a priori* knowledge into SLAH, introduced in the previous chapter. The purpose of this section is the following. The initial estimated *model* is very sensitive to noise because it is derived using very few data points. The first data points used by the SLAH may mislead the algorithm due to the noisy data and produce a bad initial statistical *model* (e.g. KPIs with wrong tendencies / slopes). As a result, some additional iterations are required for the SLAH algorithm to converge. It is recalled that due to operational constraints, the automated healing needs to converge in a few days, namely with a few number of iterations. Furthermore, we need to obtain the initial *model* behaviour over a wide range of RRM parameter values. So, we need to calculate the KPIs corresponding to RRM parameter values that may not be very practical for an operating network. Hence, the purpose of this study is to show how can wrong statistical *model* be avoided due to noisy data in the first iterations of the SLAH algorithm. To this end, *a priori* knowledge is introduced. *A priori* knowledge is presented in the form of a small set of data points, typically between three to five, that produces an acceptable statistical *model* (regression functions), with the expected tendencies according to expert knowledge. Such data points can be obtained in different manners: from a database produced by other eNBs using previous troubleshooting experience; from a network simulator; or simply from data points produced artificially by the network expert. After the initial *model* estimation and with the progression

of optimization iterations, the *a priori* data points are assigned lesser and lesser weight in the *model* estimation as compared to generated data points i.e., data points generated by the network/simulator. The incorporation of the *a priori* knowledge to the logistic regression and to the full SLAH algorithm, in order to achieve fast convergence and robustness, is presented. The method is applied to the problem of troubleshooting of Inter-Cell Interference Coordination (ICIC) of a LTE network.

4.5.1 Relation to generic automatd healing block diagram

The details of SLAH are the same as explained in 4.2 and 4.4.1. Here, the switch position in the generic automated healing block is 2 for the incorporation of the *a priori* knowledge in the form of data points into SLAH.

4.5.2 A priori knowledge incorporation

If we have the *a priori* knowledge about the expected behaviour of the *model* in the form of *a priori* data points (RRM-KPI pairs), this knowledge can be incorporated into the troubleshooting algorithm to derive the initial *model*. However, the *a priori* information may not be exact and can be erroneous because of scaling and translation errors w.r.t functional relations in the exact *model*. Therefore, as the iterations of the troubleshooting process start and progress, the newly generated data points are assigned more weight during the β_m (m denotes the corresponding KPI) parameters estimation of the refined *model* calculation, as compared to the *a priori* data points. The process continues until the assigned weight to the initial *a priori* points becomes very small, hence, canceling out the effect of the initial *a priori* data points. *A priori* knowledge incorporation into LoR model is defined as follows: We follow the same notation for Y_m and X as in Section 4.4.1. The β_m vector is estimated using Maximum likelihood Estimation (MLE) for Generalized Linear Models (GLMs). MLE endeavors to find the most "likely" values of the unknown distribution parameters that maximize the "likelihood function" for a given set of data sequence. The likelihood function is the probability for the occurrence of a sample sequence given that the probability density function of given data sequence for the the unknown parameters is known.

The output variable $y_{m,i}$ is given as $y_{m,i} = \mathbb{E}(N')$. Where for KPIs like mean File Transfer Time (FTT) of an eNB, the N' is the realisation of FTT values of the mobiles attached to this eNB. While for KPIs like mean Block

Call Rate (BCR) of an eNB, N' is the realisation consisting of the successes (a success represented as 0) or failures (a failure represented as 1) of the mobiles accessing this eNB. In the case of GLMs [97], the distribution of realisation N' is not necessarily Normal but has to be a member of the "Exponential Dispersion Family" [98]. The probability density function of N' in this case is given as:

$$f(N') = c(y_{m,i}, \phi) exp\left[\frac{y_{m,i}\theta_{m,i} - b(\theta_{m,i})}{a_{m,i}(\phi)}\right] \tag{4.18}$$

where ϕ is the dispersion parameter. The choice of $\theta_{m,i}$, $a(.)$, $b(.)$ and $c(.)$ is determined by the actual probability function, i.e., binomial in the case of LoR. The KPIs are assumed to follow the binomial distribution as e.g., in the case of mean BCR, the number of mobiles accessing an eNB can be considered as Bernoulli trials. KPIs like mean FTT of mobiles attached to an eNB follow a Normal ditribution according to the central limit theorem, owing to the large number of FTT values. However, we know that normal distribution can approximate a binomial distribution having large number of Bernoulli trials. Hence, the assumption of binomial distribution is considered equally valid for FTT. Hence, for binomial distribution,

$$\theta_{m,i} = \frac{\hat{y}_{m,i}}{1 - \hat{y}_{m,i}} \tag{4.19}$$

where $\hat{y}_{m,i} = f_{log}(\eta_{m,i})$ and $\eta_{m,i} = x'_i\beta_m$. Assuming constant dispersion,

$$a_{m,i}(\phi) = 1 \tag{4.20}$$

Similarly,

$$b(\theta_{m,i}) = \ln(1 + exp(\theta_{m,i})) \tag{4.21}$$

If $nt_{m,i}$ is the number of Bernoulli trials for the i^{th} data point, $c(y_{m,i}, \phi)$ is given as:

$$c(y_{m,i}, \phi) = \binom{nt_{m,i}}{nt_{m,i}y_{m,i}} \tag{4.22}$$

In such case, the log-likelihood l_m for the output variables is given as [98]:

$$l_m(\beta_1, ..., \beta_p) = \sum_{i=1}^{n} \frac{y_{m,i}\theta_{m,i} - b(\theta_{m,i})}{a_{m,i}(\phi)} + \sum_{i=1}^{n} \ln c(y_{m,i}, \phi) \tag{4.23}$$

In order to find the maximum likelihood estimation of $\beta_{m,p}$ where $p = \{1, 2\}$, the derivative of the likelihood function l_m with respect to $\beta_{m,p}$ is taken [98] and equated to 0:

$$\frac{\partial l_m}{\partial \beta_{m,p}} = 0 \tag{4.24}$$

$$\frac{\partial l_m}{\partial \beta_{m,p}} = \sum_{i=1}^{n} \frac{\partial l_m}{\partial \theta_{m,i}} \frac{\partial \theta_{m,i}}{\partial \hat{y}_{m,i}} \frac{\partial \hat{y}_{m,i}}{\partial \eta_{m,i}} \frac{\partial \eta_{m,i}}{\partial \beta_{m,p}} \tag{4.25}$$

As in [98],

$$\frac{\partial l_m}{\partial \theta_{m,i}} = \frac{1}{a_{m,i}(\phi)} \left(y_{m,i} - \frac{\partial b(\theta_{m,i})}{\partial \theta_{m,i}} \right) = y_{m,i} - \hat{y}_{m,i} \tag{4.26}$$

$$\frac{\partial \theta_{m,i}}{\partial \hat{y}_{m,i}} = \frac{1}{var\, \hat{y}_{m,i}} \tag{4.27}$$

$$\frac{\partial \eta_{m,i}}{\partial \beta_{m,p}} = X_{ij} \tag{4.28}$$

This leads to the the equation:

$$\frac{\partial l}{\partial \beta_{m,p}} = \sum_{i=1}^{n} \frac{(y_{m,i} - \hat{y}_{m,i}) X_{ij}}{var\, \hat{y}_{m,i}} \frac{\partial \hat{y}_{m,i}}{\partial \eta_{m,i}} = 0 \qquad \text{for } p = 1, 2 \tag{4.29}$$

Equation (4.29) can be expressed in the matrix notation as:

$$X^T V_m (Y_m - \hat{Y_m}) = 0 \tag{4.30}$$

V_m is a diagonal matrix with $\frac{1}{var\, \hat{y}_{m,i}} \frac{\partial \hat{y}_{m,i}}{\partial \eta_{m,i}}$ as its i^{th} diagonal element.
The matrix equation (4.30), contains equations which are non-linear functions of the β_m vector, denoted as $f(\beta_m) = 0$. Generally, it is not possible to solve $f(\beta_m)$ explicitly in terms of β_m. Hence, we solve the set of non-linear equations $f(\beta_m) = 0$ iteratively using the numerical Newton-Raphson method [99]. In this method, we take the linear approximation of $f(\beta_m)$ in the neighbourhood of a point β_m^t using Taylor series as below:

$$f(\beta_m^t) + \frac{\partial f(\beta_m^t)}{\partial \beta_m^T} (\beta_m - \beta_m^t) \tag{4.31}$$

where t is the iteration index and β_m^T denotes the matrix transpose of β_m. Equating (4.31) to zero, we can approximate β_m^{t+1} as follows:

$$\beta_m^{t+1} = \beta_m^t - \frac{f(\beta_m^t)}{(\frac{\partial f(\beta_m^t)}{\partial \beta_m^T})} \tag{4.32}$$

Starting from an initial β_m^0 and iteratively solving equation (4.32), β_m converges to a solution. In our case, $f(\beta_m^t) = \frac{\partial l(\beta_m^t)}{\partial \beta_m} = X^T V_m^t (Y_m - \hat{Y_m^t})$ is

termed as the score function. The term $-\frac{\partial f(\beta_m)}{\partial \beta_m^T} = -\frac{\partial^2 l(\beta_m)}{\partial \beta_m \partial \beta_m^T}$ is known as the Fisher information [98] and is usually replaced by its expectation

$$\begin{aligned}
\mathcal{I}(\beta) &= \mathbb{E}\frac{\partial l(\beta_m)}{\partial \beta_m}\frac{\partial l(\beta_m)}{\partial \beta_m^T} \\
&= \mathbb{E}[X^T V_m (Y_m - \hat{Y_m}) \left\{ X^T V_m (Y_m - \hat{Y_m}) \right\}^T] \\
&= \mathbb{E}[X^T V_m (Y_m - \hat{Y_m})(Y_m - \hat{Y_m})^T V_m X] \\
&= X^T W_m X \qquad (4.33)
\end{aligned}$$

where $W_m = V_m(\operatorname{cov} Y_m)V_m$ is a diagonal matrix with diagonal elements $[W_m]_{ii} = \frac{1}{\operatorname{var} \hat{y}_{m,i}}(\frac{\partial \hat{y}_{m,i}}{\partial \eta_i})^2 = nt_i * \hat{y}_{m,i} * (1 - \hat{y}_{m,i})$. Hence, (4.32) becomes

$$\begin{aligned}
\beta_m^{t+1} &= \beta_m^t + (X^T W_m^t X)^{-1} X^T V_m^t (Y_m - \hat{Y_m}^t) \\
&= S^t [X^T W_m^t X \beta_m^t + X^T V_m^t (Y_m - \hat{Y_m}^t)] \qquad (4.34) \\
&= S^t X^T W_m^t [X \beta_m^t + \left(W_m^t\right)^{-1} V_m^t (Y_m - \hat{Y_m}^t)] \qquad (4.35)
\end{aligned}$$

which takes the final form as

$$\beta_m^{t+1} = S_m^t X^T W_m^t z_m^t \qquad (4.36)$$

where

$$S_m^t = (X^T W_m^t X)^{-1}$$

$$z_m^t = X\beta_m^t + U_m^t (Y_m - \hat{Y_m}^t)$$

and $U_m^t = (W_m^t)^{-1} V_m^t$ is a diagonal matrix with elements $[U_m]_{ii}^t = \frac{\partial \eta_{m,i}^t}{\partial \hat{y}_{m,i}^t} = \frac{1}{\hat{y}_{m,i}^t (1 - \hat{y}_{m,i}^t)}$.

As mentioned earlier, during the *a priori* knowledge incorporation process, different weights are assigned to the *a priori* and generated data points. This can be achieved by using $H_m^t = W_m^t * A$ instead of W_m^t in (4.36) where A is a diagonal weight matrix with A_{ii} being the weight assigned to the i^{th} data point. However, we don't alter U_m^t by replacing its W_m^t matrix because U_m^t only scales the residual error.

A priori diagonal weight matrix A calculation

Let $N1$ be the number of the *a priori* data points in matrix Y. Initially, $A_{ii} = x\%$ if y_i is a generated data point and $A_{ii} = \frac{100-(n-N1)x}{N1}\%$ if y_i is an *a priori* data point. However, in the case $(n-N1)x$ exceeds a certain threshold W_{th}, meaning that the generated data points have become more important

than the *a priori* data points, we assign greater overall weight of $W_{new}\%$ to newly generated data points. Hence, $A_{ii} = \frac{W_{new}}{(n-N1)}\%$ if y_i is the generated data point and $A_{ii} = \frac{100-W_{new}}{N1}\%$ if y_i is an *a priori* data point.

The complete, *a priori* knowledge incorporated, system model or β_m estimation algorithm is given as

Initialization:
1. $X \leftarrow$ *values* X_{ij}
2. $Y_m \leftarrow$ *values* $y_{m,i}$
3. $z_m \leftarrow f_{log}^{-1}(y_{m,i})$
4. $b_m \leftarrow (X^T X)^{-1} X^T z_m$
Repeat until b_m *converges:*
5. $\eta_m \leftarrow X b_m$
6. $\hat{y}_{m,i} \leftarrow f_{log}(\eta_m)$
7. $[U_m]_{ii} = \frac{1}{\hat{y}_{m,i}(1-\hat{y}_{m,i})}$
8. $[W_m]_{ii} = nt_{m,i} * \hat{y}_{m,i} * (1 - \hat{y}_{m,i})$
9. $z_m \leftarrow X b_m + U_m(Y_m - \hat{Y_m})$
10. $S_m \leftarrow (X^T H_m X)^{-1}$
11. $b_m \leftarrow S_m X^T H_m z_m$
End Repeat

b_m is the matrix containing the estimated values of β_m

4.5.3 Automated healing model

The details of the adaptation of SLAH to interference mitigation in a LTE network is the same as in Section 4.4.2.2. While the details of interference mitigation model are given in Section 4.4.2.1.

The complete SLAH algorithm, with the *a priori* enhancement modifications, is written as in Table 4.6:

4.5.4 ICIC automated healing use case

4.5.4.1 Simulation scenario

A LTE network comprising 45 eNBs in a dense urban environment is depicted in Figure 4.10. Downlink transmissions are considered here. The simulation parameters are listed in Table 4.7. A MATLAB LTE simulator described in Section 4.4.3.1 has been used [89] [100].

Initialization:
1. Identify the most coupled eNB s with eNB c among
the neighbours in $NS1$
2. For each eNB_j, $j \in c \cup NS1$, choose an initial set of n data points P_n^j
according to *a priori* information
3. Assign equal weight to all the diagonal elements of A.
Repeat until convergence:
4. For each eNB_j, compute statistical *model* using LoR for FTT and BCR
using the corresponding data points in P_n^j
6. Compute a new vector (α_j), $j \in NS1$ (using equations (4.12)
and (4.15) or (4.16))
7. Apply (α_j) in the network/simulator and observe (FTT_j) and (BCR_j)
, $j \in c \cup NS1$. Compute new data point p_{n+1}^j
8. Update P_{n+1}^j: $P_{n+1}^j = P_n^j \cup p_{n+1}^j$
9. n=n+1
10. calculate A

Table 4.6: The complete SLAH Algorithm

Parameters	Settings
System bandwidth	5MHz
Cell layout	45 eNBs, single sector
Maximum eNB transmit power	32 dBm
Inter-site distance	1.5 to 2 KM
Subcarrier spacing	15 kHz
PRBs per eNB	24 (8 in each sub-band)
Path loss	L=128.1 + 37.6 $\log_{10}(R)$, R in kilometers
Thermal Noise Density	-173 dBm/Hz
Shadowing standard deviation	6 dB
Traffic model	FTP
File size	6300 Kbits
PRBs assigned per mobile	1 to 4 (First-come, first-serve basis)
Mobilty of mobiles	No

Table 4.7: The system level simulation parameters.

For each new value of α the simulator runs for 2500 time steps (seconds) to allow the convergence of the processed KPIs. The BCR and FTT KPIs used by the SLAH algorithm are averaged at an interval varying from 500 to 2500 seconds while discarding the first *heating time* samples. It is noted that for a given traffic demand, the BCR provides a capacity indicator while the FTT is more related to the user perceived QoS. The interference matrix elements used in equations (4.12) and (4.14) are calculated once for the reference solution (see paragraph below) during a longer time interval varying from 500 to 7000 seconds to achieve higher accuracy.

Reference Solution

An optimal default value for α is chosen as 0.5 for all eNBs in the network and will serve as the reference (default) solution. The calculation of reference solution is given in Section 4.4.3.1. The value of $\alpha = 0.5$ is selected due to the smaller inter-cellular interference and the minimum energy consumption in the network.

4.5.4.2 Automated healing scenario

A problematic eNB with the worst performance in the reference network of Figure 4.10, namely $eNB_{c=13}$, is selected for troubleshooting using the SLAH algorithm. The eNB_j, where $j \in NS1 = \{14, 15, 22, 23, 43, 45\}$, is one of the six first tier geographical neighbours of $eNB_{c=13}$. It is recalled that the SLAH modifies the α parameters of eNB_j while leaving unchanged α_c which is fixed to the reference default value of 0.5. The set $NS2$ of the second tier neighbours of the problematic eNB consists of eNB_1, eNB_{10}, eNB_{11}, eNB_{16}, eNB_{18}, eNB_{24}, eNB_{37} and eNB_{44}. Denote by *optimization zone* the subnetwork comprising $eNB_{c=13}$ and its first tier $NS1$, and by *evaluation zone* the subnetwork comprising the $eNB_{c=13}$ and its first two tiers $NS1$ and $NS2$. The $eNB_{s=45}$ is the eNB most coupled with $eNB_{c=13}$.

4.5.4.3 A priori knowledge assumption

As mentioned earlier, the *a priori* data points can be obtained from a data base of eNBs, from the network or produced artificially by the network expert. The *a priori* knowledge for an eNB is given as a set of KPI curves used by the SLAH algorithm. Extensive numerical experimentations have shown that the important features of the *a priori* curves are given by their tendency, namely the monotonous increasing or decreasing behaviour, following a typical behaviour of the KPI. It is noted that the effectiveness of the SLAH

method empowered by the *a priori* knowledge is little affected by the amplitude of the *a priori* KPI curves, rendering more efficient the troubleshooting process. The same holds for small variations in the curves' shapes, as long as the correct monotonous property is used. In the troubleshooting scenario analyzed in this section, the *a priori* KPI curves have been taken from a different scenario with different eNBs studied in Section 4.4.3, as shown in Figures 4.21 and 4.22 for BCR and FTT respectively.

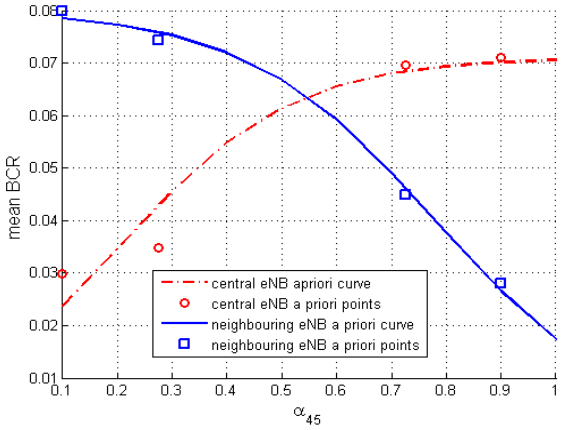


Figure 4.21: *A priori* curves and data points for BCR as a function of $\alpha_{s=45}$.

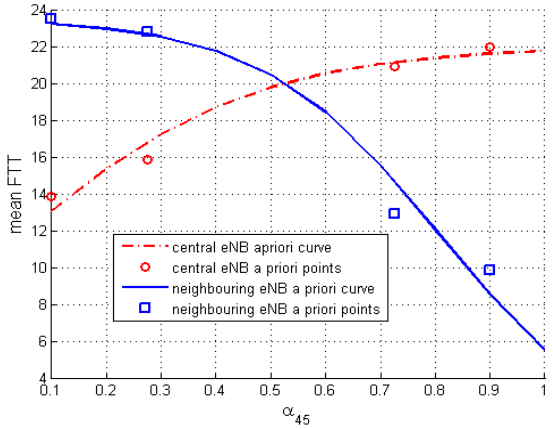


Figure 4.22: *A priori* curves and data points for FTT as a function of $\alpha_{s=45}$.

The four *a priori* data points from which these *a priori* curves can be reconstructed are given as in Table 4.8. Here BCR_{neigh} and FTT_{neigh} denote the *a priori* curves used for each neighbour in $NS1$.

α_s	BCR_c	BCR_{neigh}	FTT_c	FTT_{neigh}
0.1	0.03	0.08	14	23.5
0.275	0.035	0.075	16	23
0.725	0.7	0.045	21	13
0.9	0.71	0.028	22	10

Table 4.8: Table showing *a priori* data points

$\alpha_{c=13}$	$\alpha_{j=14}$	$\alpha_{j=15}$	$\alpha_{j=22}$	$\alpha_{j=23}$	$\alpha_{j=43}$	$\alpha_{s=45}$
0.50	0.79	0.61	0.84	0.95	0.71	0.55
0.50	0.79	0.60	0.84	0.95	0.70	0.54
0.50	0.87	0.76	0.90	0.97	0.83	0.73
0.50	0.81	0.65	0.86	0.95	0.74	0.60
0.50	0.81	0.64	0.85	0.95	0.74	0.59
0.50	0.81	0.64	0.85	0.95	0.74	0.59
0.50	0.81	0.64	0.85	0.95	0.74	0.59
0.50	0.81	0.64	0.85	0.95	0.74	0.59

Table 4.9: Shows the α values calculated during optimization.

The matrix A is recalculated before each optimization iteration. Let n be the number of data points at each optimization iteration. Initially, $A_{ii} = x\% = 20\%$ is the weight assigned to each generated data point and the weight assigned to each *a priori* data point is $A_{ii} = \frac{100-(n-4)20}{4}\%$. However, when total weight $(n-4)x$ assigned to the generated data points exceeds $W_{th} = 75$, this means that generated data points have become more important than *a priori* data points. Hence, the total overall weight of $W_{new} = 90\%$ is assigned to the generated data points. Hence, the weight of $A_{ii} = \frac{90}{n-4}\%$ is assigned to each generated data point. While weight of $A_{ii} = \frac{100-90}{4}\%$ is assigned to each *a priori* data point.

4.5.4.4 Results

Table 4.9 shows the convergence of $\alpha_{s=45}$ after the application of the *a priori* incorporated SLAH. The initial value of $\alpha_s = 0.55$ is generated using the *a priori* knowledge. After 4 iterations of the troubleshooting algorithm, α_s converges to value of $\alpha_s = 0.59$. Figure 4.23 compares the convergence of α_s in the case of *a priori* incorporated (continuous line) SLAH with the one not using the *a priori* information (dashed line). It is apparent that less number of iterations are required in the former case as we don't need to do iterations in order to generate the data points for initial *model* estimation.

Furthermore, this initial *model* is very sensitive to noise as it is derived from few points. Hence, using the noise free *a priori* data points for initial *model* estimation has resulted in a smoother and quicker convergence, as is evident from the blue curve.
It is also apparent from Figure 4.23 that in the abscence of the *a priori* knowledge, the values of α_s=0.05 and 0.95 may be very low or high for an operating network. The data points corresponding to these α_s values are generated in order to get the exact behaviour of the initial *model* over whole α_s range.

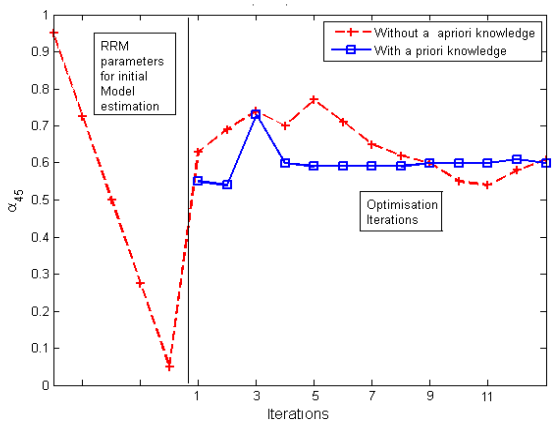


Figure 4.23: Figure showing the convergence of $\alpha_{s=45}$ with and without *a priori* knowledge incorporated algorithm.

Figure 4.24 and Figure 4.25 show, the final BCR and FTT curves, respectively, after $\alpha_{s=45}$ convergence. The KPI curves for $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ are shown while the KPI curves for $eNB_{j=14}$ and $eNB_{s=45}$ are not shown as they show a similar trend. The points encircled in green show the *a priori* data points for the neighbouring eNBs. The points encircled in red show the *a priori* data points for the central eNB. It is evident from the KPI curves that in the end of the optimization process, the effect of initially assumed *a priori* data points has almost disappeared, as the KPI curves follow the newly generated KPI data points and have shifted away from the initial *a priori* points. The convergence of $\alpha_{s=45}$ can be seen from the concentration of KPI points around $\alpha_{s=45} = 0.59$ in the two figures.

The BCR of $eNB_{c=13}$ improves from value of 5.28% to 3.96% and its FTT decreases from 16.9362 to 15.1480. Figures 4.26(a) and 4.26(b) show, in descending order, the BCR and the FTT respectively for the reference (square) and the optimized (circle) eNBs in the evaluation zone. It is noted

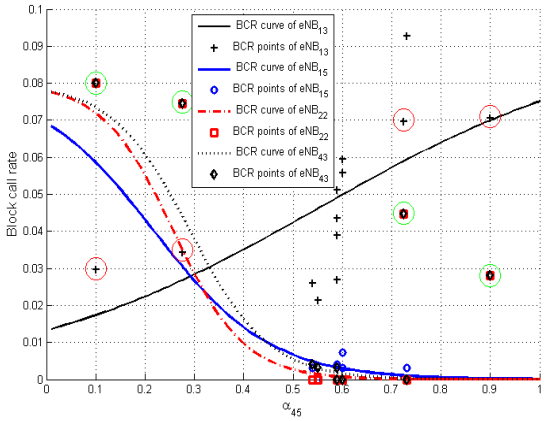


Figure 4.24: Mean BCR values of $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ along with the corresponding regression curves as a function of $\alpha_{s=45}$.

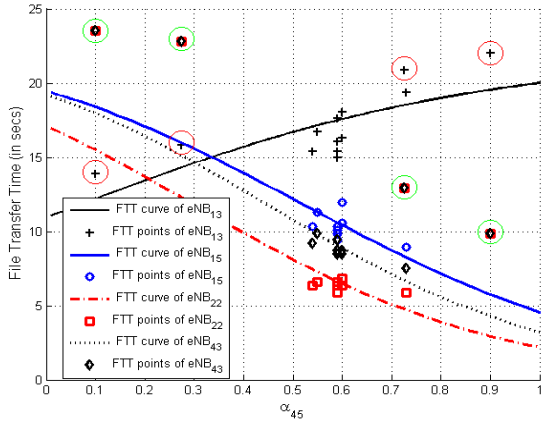


Figure 4.25: Mean BCR values of $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ along with the corresponding regression curves as a function of $\alpha_{s=45}$.

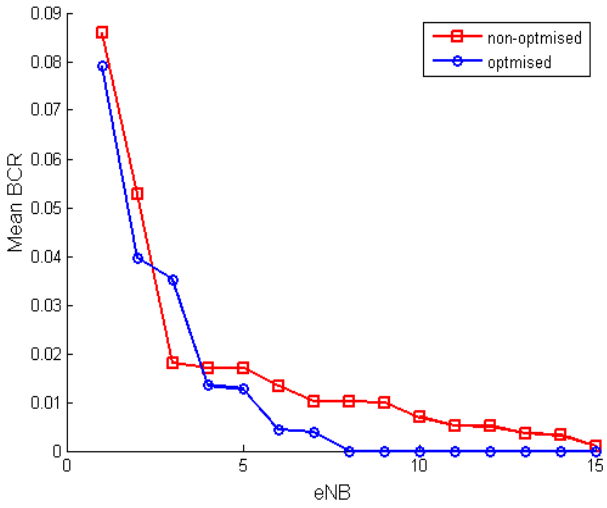
that the order of the stations in the two curves of each figure may not be preserved. The performance of the problematic eNB has been improved while the performance of the neighbouring eNBs in the evaluation zone has, on the average, improved. The average improvement of BCR is 25%, while that of FTT is 11.88%. The troubleshooting scenario has been repeated six times with and without the *a priori* knowledge incorporation. It has been observed that in the absence of the *a priori* knowledge the mean number of iterations required for SLAH convergence is 19.6. With the use of *a priori* knowledge the number of iterations is reduced to 10.

4.6 Enhancement of the Statistical Learning Automated Healing (SLAH) technique using packet scheduling

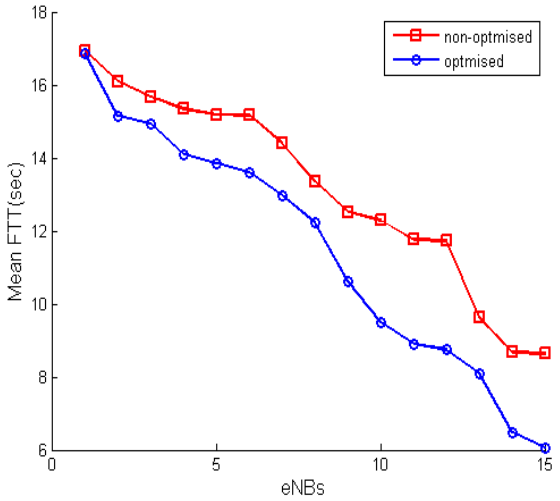
This section investigates the use of packet scheduling in conjunction with the SLAH methodology in order to obtain the optimal performance for the problematic eNB and its first-tier neighbours. It is assumed that the cause of the bad performance of an eNB is diagnosed as excessive interference from neighbouring eNBs. In the first step, SLAH is used for this interference mitigation using Inter-cell Interference Coordination (ICIC), by adjusting interference from first tier nieghbouring eNBs.

During the second step, packet scheduling i.e., α-fair scheduling in our case [94], is used to further enhance the performance of the problematic eNB and its first tier neighbours. The α-fair scheduler includes well known schedulers such as Proportional Fair (PF), Max Throughput (MTP) and Max-Min Fair (MMF) schedulers.

Small adjustment of α-fair scheduler of an eNB has very little effect on the KPIs of the neighbouring eNBs. Hence, after solving the problem of the problematic eNB by application of the SLAH methodology, the α parameter optimization can be used to obtain an additional improvement using capacity/coverage compromise for each eNB individually. This coverage/capacity compromise enables us to achieve the required minimum coverage constraint on an eNodeB (eNB). On the other hand, if minimum coverage requirement of an eNB is already satisfied, additional capacity gain for an eNodeB (eNB) can be achieved. α parameter optimization is also carried out using the same principles as the SLAH, i.e., statistical learning combined with an optimization engine. This work investigates the optimization of α parameter in offline mode, so that it is computationaly light and implemented in Operation and Maintenance Center (OMC) alongwith the SLAH algorithm. This combined



(a)



(b)

Figure 4.26: KPIs for the eNB in the evaluation zone in descending order for mean BCR (a) and mean FTT (b).

methodology, its implementation and a detailed LTE use case are described where automated healing of a faulty Inter-cell Intereference Coordination (ICIC) parameter is done. This new methodology is suitable for operating RANs as it converges in few iterations and is computationaly less intensive.

4.6.1 Relation to generic automatd healing block diagram

As indicated above, we assume that the fault cause has been diagnosed; our focus is only on the problem resolution phase. The generalized block diagram and its details that describes the basic principles behind the Statistical Learning Automated Healing Methodology (SLAH) are presented in Section 4.2 and Section 4.4.1. This block diagram is used to tune the ICIC and the packet scheduling parameters. However, the Optimization block is modified as follows:

- **Optimization:** The aim of the optimization problem is to determine $\hat{x}$ i.e., the value for a RRM parameter x that minimizes a cost function of a set of KPIs denoted as the *optimization set* A_o, subject to constraints on a second set of KPIs denoted as the *constraint set* A_c. The utility function U is given as

$$U = \sum_{m \in A_o} w_m \hat{y}_m \tag{4.37}$$

  where

  - $\hat{y}_m$ has the functional relation form as in (4.8).
  - w_m is the weight given to $\hat{y}_m$.

  The optimization problem is formulated as follows:

$$\hat{x} = argmax_x U(x) \tag{4.38}$$

  subject to

$$\hat{y}_h(\hat{x}) < th_h \ \forall \ h \in A_c$$

  where th_h is the threshold for $\hat{y}_h$.

4.6.2 α-fair scheduler

Consider a LTE eNB with N users, using OFDMA as access technology. In OFDMA, the total frequency bandwidth of the cell is subdivided in K Physical Resource Blocks (PRBs). The scheduling policy P chooses a user

for transmission on each PRB for scheduling instant $(t_u)_{u\in\mathbb{N}}$. The notation, $P_{t_u}^{(k)} = i$ signifies that PRB k is assigned to user i for transmission at scheduling instant t_u. The instantaneous throughput of user i on PRB k at instant t_u is denoted as $r_{i,t_u}^{(k)}$. Whereas, $\overline{r}_{i,t_u}^{(k)}$ defines the mean throughput of user i on PRB k during the time interval $[t_0, t_u]$. If $\epsilon > 0$ denotes a small averaging parameter, $\overline{r}_{i,t_u}^{(k)}$ is defined as [95]:

$$\overline{r}_{i,t_{u+1}}^{(k)} = (1-\epsilon)\overline{r}_{i,t_u}^{(k)} + \epsilon\delta_{P_{t_{u+1},i}^{(k)}} r_{i,t_{u+1}}^{(k)} \tag{4.39}$$

here δ denotes Kronecker's delta.

The user to schedule for transmission at time t_{u+1} on PRB k is chosen as:

$$i^{*(k)} = argmax_{0\leq i\leq N} \frac{r_{i,t_{u+1}}^{(k)}}{\left(\overline{r}_{i,t_u}^{(k)} + d\right)^{\alpha}} \tag{4.40}$$

here $\overline{r}_{i,t_0} = 0 \ \forall \ i$. $d > 0$ avoids singularity at zero and should be as small as possible.

Consequently, the mean throughput of user i in the time interval $[t_0, t_u]$ is denoted as

$$\overline{r}_{i,t_u} = \sum_{k=1}^{K} \overline{r}_{i,t_u}^{(k)} \tag{4.41}$$

We suppose that all the scheduling resources of an eNB are utilised even if there is at least a single user connected to it. Hence, in this case, changing the α parameter of the eNB has little effect on the KPIs of neighbouring eNBs.

It is evident from (4.40) and (4.41) that when $\alpha = 0$, the α-fair scheduler can be characterized as Maximim Throughput (MTP) scheduler. Hence for $\alpha = 0 \rightarrow 1$, the α fair scheduler evolves from MTP to Proportional Fair (PF) scheduler. Similarly, for $\alpha = 1 \rightarrow \infty$, the PF scheduler evolves into Max-Min Fair (MMF) scheduler. This signifies that for $\alpha = 0 \rightarrow \infty$, $\overline{r}_i$ (capacity) changes from a maximum value to a minimum value, while number of users served (coverage) changes from a minimum to maximum value to achieve fairness in the PF scheduler.

4.6.3 Automated healing algorithm

In this section, we describe the use of SLAH in conjunction with α-fair scheduling in order to achieve the required Quality of Service (QoS) objective. The cause of the degraded performance of an eNB is assumed to be due to excessive inter-cell interference.

In the first step, SLAH automated healing methodology uses the Inter-cell Interference Coordination (ICIC) for the mitigation of excessive intereference. The details of SLAH methodology and ICIC scheme are explained in 4.4.2.2 with the ICIC RRM parameter α renamed as γ in this section for the sake of clarity. The KPIs chosen for SLAH are the mean Block Call Rate (BCR) and the mean Average Bit Rate (ABR) of mobiles attached to an eNB. The subscript, in the mathematical notation of a KPI or a RRM parameter denotes the corresponding eNB. $NS1$ is the set of first tier neighbours of the problematic eNB, eNB_c. In SLAH we achieve our QoS objective by optimizing γ_s. Where eNB_s is the most coupled eNB_j, $j \in NS1$, with eNB_c in terms of interference. This coupling is determined by the intereference matrix [4] element I_{cj}. I_{cj} is the sum of interference experienced by mobiles attached to eNB_c caused due to downlink transmissions of eNB_j, $j \in NS1$. The SLAH optimization objective in this case is defined as the utility function:

$$U1 = ABR_c + \sum_{j \in NS1} \omega_j ABR_j. \tag{4.42}$$

As mentioned in SLAH methodology, the KPIs ABR_j and BCR_j are logistic function of γ_s as in equation (4.8). The weighting coefficients ω_j depend on the relative contribution of I_{cj} with respect to the sum on all eNBs in $NS1$ and are given by:

$$\omega_j = \frac{I_{cj}}{\sum_{l \in NS1} I_{cl}} \tag{4.43}$$

satisfying the condition $\sum_{j \in NS1} \omega_j = 1$. The optimization problem is formulated as follows:

$$\gamma_s = argmin_{\gamma'_s} U1(\gamma'_s) \tag{4.44}$$

subject to $BCR_q < Th_{BCR}$; $q \in c \cup NS1$. Th_{BCR} is the maximum allowable threshold for BCR.

In the second step, we optimize the individual eNB_qs, $q \in c \cup NS1$, by optimizing the corresponding α_qs in order to achieve coverge/capacity compromise. As mentioned in Section III, varying α_q of an eNB has little impact on the KPIs of the neighbouring eNBs. Hence, we can optimize α_q of an eNB independently of its neighbours. In order to achieve our optimization objective we try to maximize following utility function:

$$U2 = argmax_{\alpha_q}(ABR_q) \qquad ; \qquad q \in c \cup NS1 \tag{4.45}$$

subject to $BCR_q < Th_{BCR} - \delta$.

Here, δ is a small number that ensures that after optimization BCR_q does not exceed Th_{BCR}. It is noted that ABR_q and BCR_q are the logistic functions of α_q as in equation in (4.8).

A *data point* p_k^q is denoted by the vector $p_k^q = (\alpha_q, ABR_q, BCR_q)_k$, where $q \in c \cup NS1$ and k denotes the iteration index. The α-fair scheduling based enhancement procedure starts with few initial data points and a new data point is generated after each optimization iteration. Hence, the iteration index equals the total number of generated data points. P_k^q denote a set of k data points for eNB_q, $q \in c \cup NS1$.

The complete automated healing algorithm is given below as:

1. Apply SLAH algorithm as given [4] untill γ_s, converges to an optimal value.
Initialization:
2. Generate the initial set of k data points P_k^q, $q \in c \cup NS1$, by applying k different α_q values to the network/simulator one by one and obtaining the corresponding KPIs.
Repeat until convergence:
3. For each eNB_q, compute the statistical *model* using LoR for BCR and ABR using the corresponding data points in P_k^q
4. Compute a new α_q, $q \in c \cup NS1$ (using equation (4.45))
5. Apply the new α_q values to the network/simulator and observe (ABR_q) and (BCR_q), $q \in c \cup NS1$.
Compute the new data point p_{k+1}^q
6. Update P_{k+1}^q: $P_{k+1}^j = P_k^q \cup p_{k+1}^q$
7. k=k+1

4.6.4 Simulations and results

4.6.4.1 Simulation scenario

A LTE network composed of 27 eNBs in a dense urban environment, as depicted in Figure 4.27, is simulated using MATLAB-based simulator described in 4.4.3.1. We consider downlink transmissions. The simulation parameters are listed in Table 4.10.

The simulator takes the Monte Carlo snapshots of the simulated network at each time step of one second. These snapshots account for the time evolution of the network. After each time step, the position of mobile terminals are updated, Call Admission Control (CAC) is performed for new users, mobile Handovers HOs are carried out and some mobiles leave the network because either their session duration is completed or they are dropped.

Streaming traffic supporting the H.264 standard with variable bitrate varying from 64 Kbits/sec to 50 Mbits/sec is considered. The CAC procedure of a new user is defined as follows: Given the SINR of the users in a cell, if

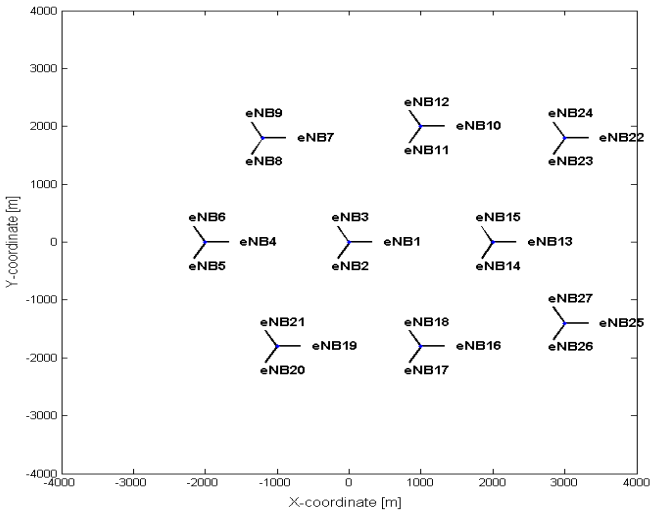


Figure 4.27: The network diagram of the simulated system.

a new user arrives, its bit rate is calculated alongwith the already scheduled users using (4.41). (4.41) uses (4.40) which calculates $r_{i,t_{u+1}}^{(k)}$ from $S_{i,t_{u+1}}^{(k)}$ (i.e., SINR of user i on PRB k at t_{u+1}), using quality tables obtained from link level simulations. The new user is only admitted if its bit rate is above 64 Kbits/sec. Otherwise, the new user is blocked.

The streaming session of a user is dropped if its bit rate falls below the threshold of 64 Kbits/sec for 5 consecutive seconds.

The mean ABR is an indicator of the capacity of an eNB. On the other hand, the mean BCR is used as the coverage indicator. Lower α value setting for an eNB means that lower SINR users will get less resources. Hence, a low SINR user may be not be admitted to the eNB because of the CAC procedure described in the previous paragraph. This results in higher mean BCR (bad coverage) and higher mean ABR (good capacity). On the other hand, higher α means that lower SINR users are assigned resources to achieve fairness and are admitted by eNB. This results in a lower mean BCR (good overage) and lower mean ABR (bad capacity).

For each new α or γ value, the duration of each simulation is 3300 time steps. While calculating KPIs, in order to account for initial transient effects, the KPIs are averaged starting from 500 seconds. However, the interference matrix I used in the SLAH algorithm is calculated over time intervals from 500 to 7000 seconds, using reference solution or the default γ=0.5 [4], for an increased accuracy. The default α value of an eNB is set to 1.

Parameters	Settings
System bandwidth	5MHz
Cell layout	45 eNBs, single sector
Maximum eNB transmit power	32 dBm
Inter-site distance	1.5 to 2 KM
Subcarrier spacing	15 kHz
PRBs in each sub-band	18 (6 in each sub-band)
Path loss	L=128.1 + 37.6 $\log_{10}(R)$, R in kilometers
Thermal noise density	-173 dBm/Hz
Shadowing standard deviation	6 dB
Traffic model	streaming to support H.264 video bit rates
Streaming session duration	30sec
Packet scheduling scheme	α-fair scheduling
Mobility of mobiles	20%
Mobile speed	8.33 m/s

Table 4.10: System level simulation parameters

4.6.4.2 Automated healing scenario

A problematic eNB with the worst performance in the simulated network of Figure 4.27, namely $eNB_{c=1}$, is selected for automated healing using the SLAH algorithm. The eNB_j, where $j \in NS1 = \{2, 3, 11, 14, 15, 18\}$, is one of the six first tier geographical neighbours of $eNB_{c=1}$. The set $NS2$ of the second tier neighbours of the problematic eNB consists of eNB_4, eNB_7, eNB_8, eNB_{10}, eNB_{12}, eNB_{13}, eNB_{16}, eNB_{17}, eNB_{19}, eNB_{21}, eNB_{23} and eNB_{27}. Denote by *optimization zone* the subnetwork comprising $eNB_{c=1}$ and its first tier $NS1$. Hence, eNB_q, $q \in c \cup NS1$, denotes an eNB of the *optimization zone*.

4.6.4.3 Results

After the first step of the SLAH optimization $\gamma_{s=3}$ converges to a value of 0.49 with the corresponding values of $\gamma_{j=2} = 0.70$, $\gamma_{j=11} = 0.74$, $\gamma_{j=14} = 0.96$, $\gamma_{j=15} = 0.68$ and $\gamma_{j=18} = 0.74$. There is a 61% improvement in mean BCR of problematic $eNB_{c=1}$ as compared to the reference solution value. The mean BCR of $eNB_{c=1}$ decreases from a value of 11.98% to 4.62% i.e., below the maximum allowable threhold of 5%. The mean BCR of $NS1$ is improved by

	$\alpha_{c=1}$	$\alpha_{j=2}$	$\alpha_{s=3}$	$\alpha_{j=11}$	$\alpha_{j=14}$	$\alpha_{j=15}$	$\alpha_{j=18}$
Phase I	0.10	0.10	0.10	0.10	0.10	0.10	0.10
	0.40	0.40	0.40	0.40	0.40	0.40	0.40
	0.7 0	0.70	0.70	0.70	0.70	0.70	0.70
	1	1	1	1	1	1	1
	1.3	1.3	1.3	1.3	1.3	1.3	1.3
Phase II	1.2	0.8	1.2	0.6	0.1	0.9	0.7
	1.2	0.8	1.2	0.6	0.1	0.8	0.7
	1.2	0.8	1.3	0.6	0.1	0.9	0.8
	1.2	0.8	1.3	0.6	0.1	0.9	0.8

Table 4.11: Phase-I shows the initially chosen α_q values for *optimization zone* . Phase-II shows the α_q values calculated during optimization.

19%. However, these improvements are at the cost of 4% decrease in mean ABR of the *optimization zone*.

In the second step, the optimization using α-fair scheduling is carried out. The convergence of inidividual α_q values is shown in table 4.11. Almost all the α_q values converge to an optimal value just after 4 iterations during the optimization phase.

Figure 4.28(a) and 4.28(b) show mean BCR and mean ABR curves, respectively, for $eNB_{c=1}$, $eNB_{q=2}$, $eNB_{q=3}$ and $eNB_{q=15}$ after convergence by α-fair optimization. The covergence of this scheme is apparent from the concentration of data points around final α_q values. The KPI curves of $eNB_{q=11}$, $eNB_{q=14}$ and $eNB_{q=18}$ are not shown for the sake of clarity as they follow a similar trend.

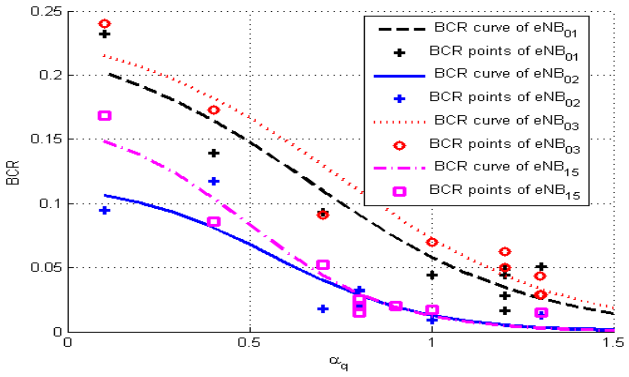
After the α-fair optimization, an improvement of 3.10% in mean ABR of *optimization zone* as compared to its value after SLAH optimization. On the other hand, the degradation in mean BCR of each eNB in the *optimization zone* remains below the threshold of 5%.

4.7 Application of SLAH for LTE mobility

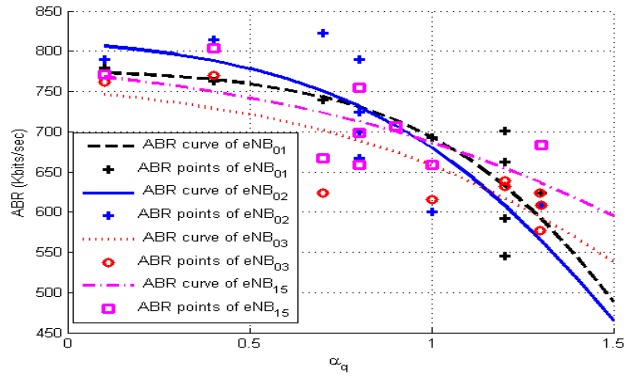
The details of SLAH are the same as explained in Section 4.2 and 4.4.1. The enhancement using *a priori* knowledge or α fair scheduling is not done in this case. The Optimization block diagram is modeifed as:

- **Optimization:** The aim of the optimization problem is to determine $\hat{x}$ i.e., the value for a RRM parameter x that minimizes a cost function of a set of KPIs denoted as the *optimization set* A_o, subject to constraints on a second set of KPIs denoted as the *constraint set* A_c. The utility function U is given as

$$U = \sum_{m \in A_o} w_m \hat{y}_m \tag{4.46}$$



(a)



(b)

Figure 4.28: Mean KPI values and LoR regression curves as a function of α_q for $eNB_{q=1}$, $eNB_{q=2}$, $eNB_{q=3}$ and $eNB_{q=15}$ (a) mean BCR (b) mean ABR.

where

- $\hat{y}_m$ has the functional relation form as in (4.8).
- w_m is the weight given to $\hat{y}_m$.

This section discusses the adaptation of SLAH for the automated healing of mobility in LTE, in terms of the handover margin which determines the handover process from one eNB to an adjacent one. The LTE Mobility model used is same as in Section 2.7. The simulation results, which we obtain for a practical use case, show the advantage of this new, automated troubleshooting methodology.

4.7.1 Adaptation of SLAH to mobility parameter

We now consider the adaptation of the troubleshooting algorithm to the mobility parameter of the LTE network by optimizing the HM parameter of the degraded eNBs. We assume that the cause of the degraded performance has been diagnosed and is related to a bad mobility parameter setting of the mobile terminals attached to the problematic eNB and its first tier neighboring eNBs.

Let us denote by eNB_c (c standing for *central*) an eNB with degraded performance. Let $NS1$ and $NS2$ denote the set of first and second tier neighbouring eNBs of eNB_c, respectively. Hence, eNB_j where $j \in NS1$ and eNB_t where $t \in NS2$, denote the first and second tier neighbours of eNB_c respectively.

To heal eNB_c, HM_{jc}, $j \in NS1$, are updated and optimized. In order to reinforce the effect of optimization of HM_{jc} on the KPIs, the HM_{jt} and HM_{cj} for the immediate geographical neighbours are calculated as a function of HM_{jc}, given as: $HM_{cj} = HM_{max} - HM_{jc}$ and $HM_{jt} = HM_{jc}$, where HM_{max} is the maximum HM value between any two eNBs.

We use the notion of coupling between eNB j and c which is expressed in terms of the interference that eNB_j produces on the mobile terminals connected to eNB_c and can be written in terms of the interference matrix element I_{cj} [85] [86]. Hence the larger I_{cj}, the stronger the coupling between the two eNBs. In this work, the matrix element I_{cj} is equal to the sum of the interferences perceived by the mobile terminals attached to eNB_c and generated by the downlink transmissions to the mobile terminals of eNB_j.

The use of SLAH to jointly optimize all the elements of the vector (HM_{jc}), $j \in NS1$, is not a simple task. Denote by s, $s \in NS1$, the index of the eNB which is the most coupled with eNB_c, namely $s = argmax_j(I_{cj})$, s and $j \in NS1$. To reduce the complexity of the SLAH process and to enhance its scalability, we propose to adjust the HM_{jc} parameter according to the degree of coupling between eNB_j and eNB_c.

Hence, we define a functional relationship between HM_{sc} and HM_{jc}, $HM_{jc} = \mathrm{g}_j(HM_{sc})$, that accounts for the coupling terms I_{cs} and I_{cj} mentioned above, as follows:

$$HM_{jc} = \mathrm{g}_j(HM_{sc}) = HM_{sc} + (12 - HM_{sc})(1 - \frac{I_{cj}}{I_{cs}}) \tag{4.47}$$

By using Eqn. (4.47), just one parameter, HM_{sc}, needs to be optimized.

The process is scalable in the sense that the automated healing can be performed simultaneously on any number of eNBs provided they are not direct neighbours.

Two KPIs are used in the SLAH process: the Average Bit Rate (ABR) and the Block Call Rate (BCR) of the mobile terminals attached to an eNB. SLAH aims at maximizing the ABR for eNB_c and of its direct neighbours while verifying the constraints on BCR_j, $j \in c \cup NS1$.

We define the utility function used for optimization as:

$$U = ABR_c + \sum_{j \in NS1} \omega_j ABR_j \tag{4.48}$$

It is noted that ABR_j is a function of HM_{jc} and hence, via equation (4.47), of HM_{sc}. ABR_j also depends on the interference from its neighbouring eNB_j. The weighting coefficients ω_j depend on the relative contribution of I_{cj} with respect to the sum on all eNBs in $NS1$ and are given by:

$$\omega_j = \frac{I_{cj}}{\sum_{k \in NS1} I_{ck}} \tag{4.49}$$

satisfying the condition $\sum_{j \in NS1} \omega_j = 1$.

The optimization problem can now be formulated as follows:

$$HM_{sc} = argmax_{HM'_{sc}} U(HM'_{sc}) \tag{4.50}$$

Initialization:
1. Identify the most coupled eNB s with eNB c among the neighbours in $NS1$
2. For each eNB_j, $j \in c \cup NS1$, compute an initial set of k data points P_k^j
Repeat until convergence:
3. For each eNB_j, compute the statistical *model* using LR for ABR and BCR using the corresponding data points in P_k^j
4. Compute a new vector (HM_{jc}), $j \in NS1$ (using equations (4.47) and (4.50))
5. Apply (HM_{jc}) in the network/simulator and observe (ABR_j) and (BCR_j), $j \in c \cup NS1$. Compute new data point p_{k+1}^j (equation (4.51))
6. Update P_{k+1}^j: $P_{k+1}^j = P_k^j \cup p_{k+1}^j$
7. k=k+1
End Repeat

Table 4.12: The complete SLAH Algorithm

subject to $BCR_j < BCR_{th}$; $j \in c \cup NS1$.

BCR_{th} is the threshold for BCR_j. The ABR and BCR indicators in equations (4.48) and (4.50), are given in the form of a Logistic Regression (LoR) function (Eqn. (4.8)) obtained using the Statistical Learning block defined above.

Denote a *data point* p_k^j as the vector:

$$p_k^j = (HM_{jc}, ABR_j, BCR_j)_k \ ; \ j \in c \cup NS1 \tag{4.51}$$

The set of k data points for an eNB_j, $j \in c \cup NS1$, is denoted by P_k^j. The SLAH algorithm is given in Table 4.12.

The initial set P_k^j of data points in Step 2 is obtained by applying k (HM_{jc}) vectors to the network/simulator one by one and obtaining the corresponding KPIs.

4.7.2 Simulations and results

4.7.2.1 Simulation scenario

We consider a LTE network composed of 45 eNBs in a dense urban environment, as depicted in Figure 4.10.

The same Matlab LTE simaulator as described in 4.4.3.1 has been used. The network simualtion parameters are given as in Table 4.13.

Parameters	Settings
System bandwidth	5MHz
Cell layout	45 eNBs, single sector
Maximum eNB transmit power	32 dBm
Inter-site distance	1.5 to 2 KM
Subcarrier spacing	15 kHz
PRBs per eNB	21
Path loss	L=128.1 + 37.6 $\log_{10}(R)$, R in kilometers
Thermal noise density	-173 dBm/Hz
Shadowing standard deviation	6 dB
Traffic model	FTP
File size	5700 Kbits
PRBs assigned per mobile	1 to 3 (First-come, first-serve basis)
Mobility of mobiles	90%
Mobile speed	15 m/s
HM_{max}	12dB

Table 4.13: System level simulation parameters

The simulations are run for 3300 time steps, with a fixed HM value, and the KPIs are averaged using the interval between 500 and 3300 seconds to account for transient effects. The interference matrix elements used in equations (4.47) and (4.49) are calculated once for the reference solution, described next, during a longer time interval varying from 500 to 7000 seconds to achieve higher accuracy.

An optimal default value for HM is chosen as 6dB for all eNBs in the network and serves as the reference (default) solution. This reference solution is calculated in 4.4.3.1 and is used as a starting point for the automated healing process.

4.7.2.2 Automated healing scenario

We now select, for the sake of illustration, $eNB_{c=13}$ in the simulated network of Figure 4.10 as the problematic eNB. The eNB_j, where $j \in NS1 = \{14, 15, 22, 23, 43, 45\}$, are the six first tier geographical neighbours of $eNB_{c=13}$. The set $NS2$ of the second tier neighbours of the problematic eNB consists of eNB_1, eNB_{10}, eNB_{11}, eNB_{16}, eNB_{18}, eNB_{24}, eNB_{37} and eNB_{44}.

	HM_{jc}					
	$j = 14$	$j = 15$	$j = 22$	$j = 23$	$j = 43$	$j = s = 45$
Phase I	11.7	11.5	11.8	11.9	11.6	11.4
	10.5	9	10.9	11.6	9.9	8.7
	9.2	6.8	9.9	11.3	8.2	6
	8	4.4	9	11	6.4	3.3
	6.7	2	8	10.7	4.8	0.6
Phase II	9.2	6.8	9.8	11.3	8.0	5.7
	9	6.4	9.8	11.3	7.9	5.6
	8.9	6.2	9.7	11.2	7.8	5.4
	8.9	6.2	9.7	11.2	7.7	5.3
	9	6.4	9.8	11.2	7.9	5.6
	8.9	6.2	9.7	11.2	7.8	5.4
	8.9	6.2	9.7	11.2	7.8	5.4

Table 4.14: Phase-I shows the initially chosen HM_{jc} values. Phase-II shows the HM_{jc} values calculated during optimization.

Denote by *the optimization zone* the subnetwork comprising $eNB_{c=13}$ and its first tier $NS1$, and by *the evaluation zone* the subnetwork comprising the $eNB_{c=13}$ and its first two tiers $NS1$ and $NS2$. $eNB_{s=45}$ is the eNB most coupled with $eNB_{c=13}$.

4.7.2.3 Results

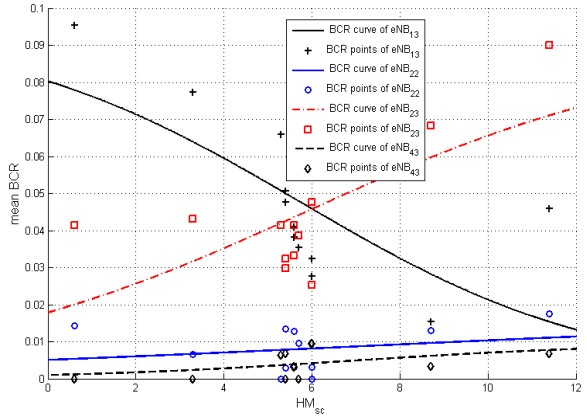
The first five values of $HM_{sc,s=45}$ in Table 4.14 are chosen in the initialization phase of SLAH. The next seven values (Phase-II in Table 4.14) are calculated iteratively by the SLAH algorithm. The values of $HM_{jc,j=14}$, $HM_{jc,j=15}$, $HM_{jc,j=22}$, $HM_{jc,j=23}$ and $HM_{j=43}$ are calculated using equation (4.47). In spite of the noisy data, one can see from Phase-II that $HM_{sc,s=45}$ converges in a few iterations. $HM_{sc,s=45} = 5.4$ is chosen as the optimized solution for the next Figures.

Figure 4.29(a) shows the mean BCR (points as well as extracted LR curves) for $eNB_{c=13}$, $eNB_{j=22}$, $eNB_{j=23}$ and $eNB_{j=43}$ as a function of $HM_{sc,s=45}$ after convergence. For the sake of clarity, the mean BCR curves for $eNB_{j=14}$, $eNB_{j=15}$ and $eNB_{j=45}$ are not shown; they are below the maximum allowable mean BCR threshold of 5%.

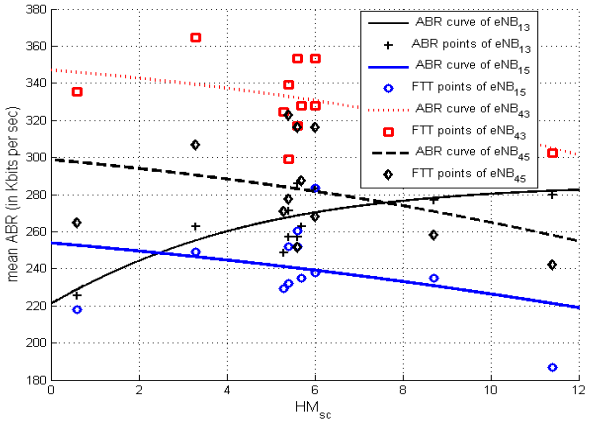
The concentration of mean BCR data points around $HM_{sc,s=45} = 5.4$ indicates the convergence of the SLAH algorithm.

The mean ABR along with its LR curves for $eNB_{j=13}$, $eNB_{j=15}$, $eNB_{j=43}$ and $eNB_{s=45}$, after convergence, are shown as a function of $HM_{sc,s=45}$ in Figure 4.29(b).

The mean ABR curves for $eNB_{j=14}$, $eNB_{j=22}$ and $eNB_{j=23}$ are, again, not shown for the sake of clarity; they show a similar trend.



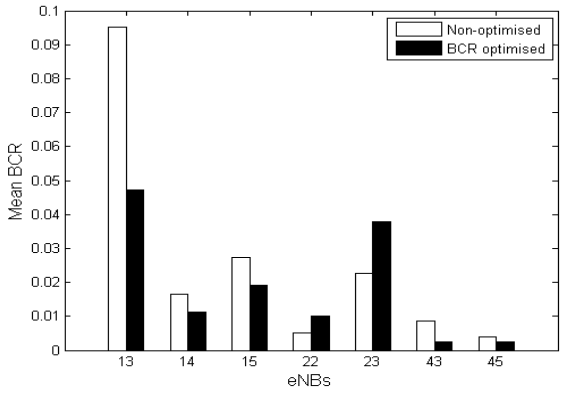
(a)



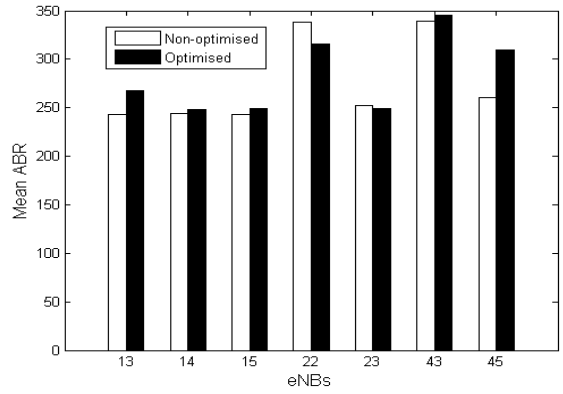
(b)

Figure 4.29: Mean KPI values and LR regression curves as a function of HM_{sc} for $eNB_{j=13}$, $eNB_{j=22}$ and $eNB_{j=43}$ (a) mean BCR (b) mean ABR.

Figure 4.30(a) shows the gain brought about by the SLAH algorithm for the optimization zone (set $NS1$ of eNBs). The mean BCR of the problematic $eNB_{c=13}$ is reduced below the threshold of 5% to 4.73% with respect to the mean BCR value of the reference solution of 9.52%. Similarly, after optimization, the mean BCR of each eNBs in $NS1$ is below the 5% threshold.



(a)

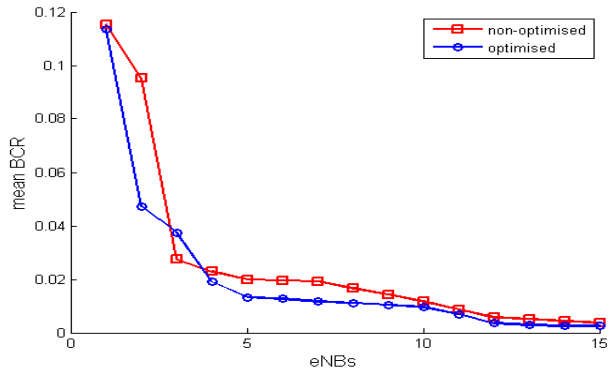


(b)

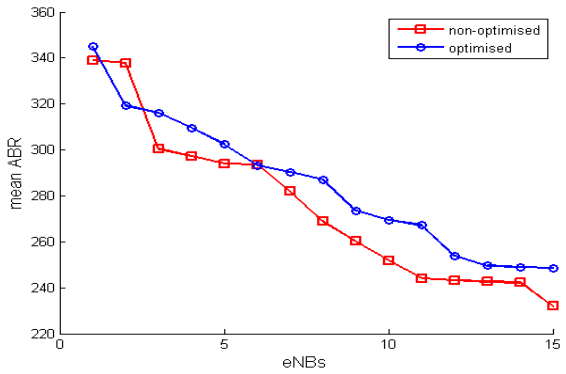
Figure 4.30: KPI of the eNBs in the optimization zone for the reference solution (white) and optimized (black) network conditions; mean BCR (a) and mean ABR (b).

As of the mean ABR, the improvement obtained by the SLAH algorithm in the optimized zone with respect to the reference solution is shown in Figure 4.30(b). The mean ABR of $eNB_{c=13}$ is increased by 2.79% whereas the average mean ABR of $NS1$ is increased by 2.51%.

Figures 4.31(a) and 4.31(b) show, in a descending order, the mean BCR and the mean ABR, respectively, for the reference (square) and the optimized (circle) eNBs in the evaluation zone. It is noted that the order of the stations in the two curves of each figure may not be preserved. One can see that, on average, the mean BCR and the mean ABR in the evaluation zone are improved. The average improvement of the mean ABR in the evaluation zone is equal to 3.51%.



(a)



(b)

Figure 4.31: KPIs in descending order for the eNBs in the evaluation zone, mean BCR (a) and mean ABR (b).

4.8 Conclusion

This chapter has presented an automated healing methodology that uses a statistical learning approach for extracting a *model* from data.

The *model* provides closed form expressions that approximate the functional relations between KPIs and RRM parameteres. The *model* helps in anticipating the behaviour of a network sub-system to new values of RRM parameters. The *model* is subsequently used in the optimization process for improving KPIs of the degraded eNB. The *model* improves during each optimization iteration. An automated healing methodology based on this principle denoted as SLAH is proposed that converges to an optimal RRM value within few iterations. The case study of the application of SLAH in the automated healing of ICIC and mobility parameters of LTE is also proposed.

We also presented, in this chapter, the enhancement in SLAH automated healing methodolgy using α-fair packet scheduling. The SLAH methodology has been used for LTE healing use case by sequentially modifying inteference mitigation and packet scheduling parameters. The results show that in the first step the coverage problem of an eNB is removed using SLAH while minimizing the degradaion in capacity. In the second step, the packet scheduling based enhancement optimizes the capacity under the given coverage constraint.

Finally, this chapter shows how the *a priori* knowledge can be introduced into- and improve the performance of the Statistical Learning Automated Healing (SLAH) method. It has been shown that the *a priori* knowledge can avoid an initially bad statistical *model*, and hence increases robustness of SLAH as well as accelerate its convergence.

Chapter 5

Conclusion and Perspectives

5.1 Conclusion

This thesis report has described my research work related to automated healing of LTE wireless networks using statistical learning. The automated healing is an important functionality of self organizing networks (SON). The main goal of the thesis has been to propose effective, practical and robust automated healing algorithms for wireless networks. This work is motivated by the necessity for wireless networks to simplify and reduce the cost of network management on the one hand and to provide a method to improve QoS of the end users in badly parameterized cells on the other hand. Lack of previous contributions in the domain of automated healing has further motivated the work of this thesis.

The automated healing methodogy developed uses the statistical modeling of performance data from the network in the form of RRM-KPI pairs. The statistical learning techniques used are the linear and logistic regression. The linear regression is used beacause of its simplicity. However, for certain cases it is not adapted and the logistic regression is used because of its capability of modeling saturation effects in KPIs. A generic automated RRM management architecture that utilises statistical learning has been proposed. In this context, the results obtained for the case studies of monitoring and auto-tuning of LTE network as a part of generic RRM management architecture show the usefulness of this approach.

A generic architecture for the automated healing methodology for wireless networks has also been proposed. Automated healing algorithms

based on this architecture have been developed. These algorithms are iterative, namely a new data point is generated in each iteration that gradually improves the statistical *model*. RRM parameter optimization approach in the context of automated healing takes into account typical hard constraints for management teams, namely resolve the problem in a short time with a small number of iterations. Due to the small number of required iterations and data points, the traditional optimization techniques are inappropriate for real network operations. Hence, an important objective of this thesis has been to come up with a new approach for automated healing in which bad RRM parameter settings are optimized in a few iterations. The small number of iterations and low computational complexity of the proposed approach makes it suitable for implementation in the OMC. The effectiveness of this automated healing methodology has been demonstrated in two case studies of automated healing, namely interference mitigation and mobility parameters of LTE networks. This methodology has also been applied to sequential automated healing of interference mitigation and packet scheduling parameters.

The RRM optimization of the LTE networks, during automated healing process has been further enhanced, by incorporation of the *a priori* knowledge into the automated healing methodology. The use of the *a priori* knowledge has the advantage of avoiding a wrong initial *model* due to noisy data, of allowing much faster convergence and of making the method more suitable for the off-line implementation. This improved methodology is applied for the automated healing of Inter-Cell Interference Coordination (ICIC) parameter in a LTE network.

5.2 Perspectives

Different directions for the future works related to the thesis can be envisaged. The automated healing methodology proposed can be extended to the automated healing of other RRM parameters of congestion control, cell selection/reselection and other mobility parameters such as timing advance etc.

The results obtained from the simulation use cases demonstrate the benefits of our automated healing scheme. In the simulator, noise has been generated numerically to adapt and test the robustness of the automated healing scheme. There are some assumptions and approximations made in the simulation that may be different from an actual

operating network. Hence, it is important to apply this automated healing scheme on the data from the real network to verify the effectiveness of this automated healing methodology. It is noted that the methodology developed in this thesis is about to be tested on a real UMTS network.

In future research, more attention can be focused on multi-RRM parameter automated healing aspect. In this thesis, we have limited ourselves to the automated healing of the fault caused by one RRM parameter. At the most, we have considered a self healing process which adjusts two RRM mechanisms, namely interference mitigation and packet scheduling in a sequential iterative manner. Similarly, the automated healing can be developed to heal, at the same time, the fault caused by multiple faulty RRM parameters. However, this problem will require the modeling of KPIs as a function of multiple RRM parameters, hence the number of initial data points will increase exponentially.

As mentioned in this thesis, the initial *model* in the SLAH methodology is very sensitive to the noise in calculated data points due to their small number. In order to overcome this problem, an approach can be adopted that involves measuring the reliability in the estimated *model*. A measure of reliability of the estimated *model* may enable us to calculate the amount of exploration required in the solution space of the RRM parameter to increase its accuracy. This leads to a more accurate *model* and results in faster convergence of the automated healing algorithm.

Bibliography

[1] 3GPP TS 32.500, "3GPP Telecommunication Management; Self-Organizing Networks (SON); Concepts and Requirements", (Release 8)

[2] M. I. Tiwana, B. Sayrac, Z. Altman,: "Statistical Learning for Automated RRM: Application to eUTRAN Mobility", IEEE Int. Conf. on Communications, ICC 2009, Dresden, Germany, June 14-18, 2009.

[3] M. I. Tiwana, B. Sayrac, Z. Altman, T. Chahed,: "Troubleshooting of 3G LTE mobility parameters using iterative statistical model refinement", IFIP Wireless Days 2009, Paris, Dec. 2009.

[4] M. I. Tiwana, B. Sayrac, and Z. Altman,: "Statistical approach for automated troubleshooting: application to LTE interference mitigation", IEEE Trans. on Vehicular Technology, Issue:99, 2010.

[5] M. I. Tiwana, Z. Altman, and B. Sayrac,: "Enhancing RRM optimization using a priori knowledge for automated troubleshooting", to appear in WIOPT 2010 (IEEE Xplore) workshop WCN3, Avignon (France), May 2010.

[6] M. I. Tiwana, B. Sayrac, Z. Altman, T. Chahed,: "Learning-based Automated Healing: Application to Mobility in 3G LTE Networks", PIMRC 2010, Istanbul, September 2010.

[7] P. Lehtimäki and K. Raivio,: "A knowledge-based model for analyzing GSM network performance", in Proc.International Conference on Industrial Engineering Applications of Artificial Intelligence Expert Systems, Bari, Italy, June 2005.

[8] J. Laiho, M. Kylväjä, and A. Höglund,: "Utilisation of advanced analysis methods in UMTS networks", in Proc.IEEE Vehicular Technology Conference, Birmingham, USA, May 2002, pp. 726730.

[9] J. Laiho, K. Raivio, P. Lehtimäki, K. Hätönen, and O. Simula,: "Advanced analysis methods for 3G cellular networks", IEEE Trans. Wireless Commun., vol. 4, no. 3, pp. 930942, May 2005.

[10] A. J. Hoglund, K. Hatonen, and A. S. Sorvari,: "A computer host-based user anomaly detection system using the self-organizing map", in Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks, vol. 5, Como, Italy, July 2000, pp. 411416.

[11] P. Lehtimäki and K. Raivio,: "A SOM based approach for visualization of GSM network performance data", in Proc. International Symposium on Intelligent Data Analysis, Madrid, Spain, Sept. 2005.

[12] G. Barreto, J. Mota, L. Souza, R. Frota, L. Aguayo, J. Yamamoto, and P. Oliveira,: "Competitive neural networks for fault detection and diagnosis in 3G cellular systems", Lecture Notes in Computer Science (LNCS), vol. 3124, pp. 207213, Aug. 2004.

[13] G. A. Barreto, J. C. M. Mota, L. G. M. Souza, R. A. Frota, and L. Aguayo,: "Condition monitoring of 3G cellular networks through competitive neural models", IEEE Trans. Neural Networks, vol. 16, no. 5, pp. 10641075, 2005.

[14] S. Andreassen, M. Woldbye, B. Falck, and S. Andersen,: "MUNIN: A causal probabilistic network for interpretation of electromyographic findings", in Proc. International Joint Conference on Artificial Intelligence, Milan, Italy, Aug. 1987, pp. 366372.

[15] D. Nikovski,: "Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics", IEEE Trans. Knowledge Data Eng., vol. 12, no. 4, pp. 509516, 2000.

[16] D. Heckerman, E. Horvitz, and B. Nathwani,: Toward normative expert systems: Part I. the Pathfinder project", Methods of Information in Medicine, vol. 31, pp. 90105, 1992.

[17] Y. Xiang, B. Pant, A. Eisen, M. P. Beddoes, and D. Poole,: "Multiply sectioned Bayesian networks for neuromuscular diagnosis", Artificial Intelligence in Medicine, vol. 5, no. 4, pp. 293314, 1993.

[18] P. Antal, H. Verrelst, D. Timmerman, S. V. Huffel, B. de Moor, and I. Vergote,: "Bayesian networks in ovarian cancer diagnosis: Potentials and limitations", in Proc. IEEE Symposium on Computer-Based Medical Systems, Houston, USA, June 2000, pp. 103108.

[19] B. Pang, D. Zhang, N. Li, and W. Kuanquan,: "Computerized tongue diagnosis based on bayesian networks", IEEE Trans. Biomed. Eng., vol. 51, no. 10, pp. 1803 1810, 2004.

[20] O. Ogunyemi, J. R. Clarke, and B. Webber,: "Using bayesian networks for diagnostic reasoning in penetrating injury assessment", in Proc. IEEE Symposium on Computer-Based Medical Systems, Houston, USA, June 2000, pp. 115120.

[21] G. Ng and K. Ong,: "Using a qualitative probabilistic network to explain diagnostic reasoning in an expert system for chest pain diagnosis", in Computers in Cardiology, Sept.2000, pp. 569572.

[22] D. Heckerman, J. Breese, and K. Rommelse,: "Decision-theoretic troubleshooting", Communication of the ACM, vol. 38, no. 3, pp. 4957, Mar. 1995.

[23] J. Breese and D. Heckerman,: "Topics in decision-theoretic troubleshooting: Repair and experiment", in Proc.Annual Conference on Uncertainty in Artificial Intelligence, Portland, Oregon, Aug. 1996, pp. 124132.

[24] D. Heckerman and C. Meek,: "Models and selection criteria for regression and classification", in Proc. Annual Conference on Uncertainty in Artificial Intelligence, Providence, Rhode Island, Aug. 1997, pp. 223228.

[25] D. Heckerman,: "A tutorial on learning bayesian networks", Microsoft Research, Redmond, Washington, Tech.Rep. MSR-TR-95-06, Mar. 1995.

[26] D. Heckerman, J. Breese, and K. Rommelse,: "Troubleshooting under uncertainty", Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-94-07, Jan. 1994.

[27] D. Heckerman, C. Meek, and G. Cooper,: "A bayesian approach to causal discovery", Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-97-05, Feb. 1997.

[28] C. Skaanning, F. V. Jensen, and U. Kjaerulff,: "Printer troubleshooting using bayesian networks", Lecture Notes in Artificial Intelligence, vol. 1821, pp. 367379, 2000.

[29] F. V. Jensen, U. Kjærulff, B. Kristiansen, H. Langseth, C. Skaanning, J. Vomlel, and M. Vomlelova,: "The SACSO methodology for troubleshooting complex systems", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 5, no. 4, pp. 321333, 2001.

[30] C. Skaanning, F. Jensen, U. Kjærulff, and A. Madsen,: "Acquisition and transformation of likelihoods to conditional probabilities for bayesian networks", in Proc. AAAI Spring Symposium on AI in Equipment Maintenance Service and Support, Palo Alto, California, Mar. 1999, pp. 3440.

[31] C. Skaanning,: "A knowledge acquisition tool for bayesian-network troubleshooters", in Proc. Annual Conference on Uncertainty in Artificial Intelligence, Stanford, USA, July 2000, pp. 549557.

[32] E. Horvitz and M. Barry,: "Display of information for time-critical decision making", in Proc. Annual Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Aug. 1995, pp. 296305.

[33] P. Lázaro, R. Barco, and J. Hermoso,: "Diagnosis of earth stations using Bayesian networks", in Proc. IASTED International Conference on Artificial Intelligence and Applications, Málaga, Spain, Sept. 2002, pp. 268272.

[34] R. Barco and R. Segura,: "Automated test and diagnosis tools for satellite ground stations", in Proc. European Test and Telemetry Conference, Paris, France, June 1999, pp. 5.3443.

[35] D. Cayrac, D. Dubois, M. Haziza, and H. Prade,: "Possibility theory in "fault mode effects analyses" - a satellite fault diagnosis application ", in Proc. IEEE Inter. Conf. on Fuzzy Systems, Orlando, FL, June 1994, pp. 1176 1181.

[36] I. Katzela and M. Schwartz,: "Schemes for fault identification in communication networks", IEEE/ACM Trans. Networking, vol. 3, no. 6, pp. 753764, Dec. 1995.

[37] M. Steinder and A. Sethi,: "Probabilistic fault localization in communication systems using belief networks", IEEE/ACM Trans. Networking, vol. 12, no. 5, pp. 809822, Oct. 2004.

[38] C. Wang and M. Schwartz,: "Fault detection with multiple observers", IEEE/ACM Trans.Networking, vol. 1, no. 1, pp. 4855, Feb. 1993.

[39] R. Deng, A. Lazar, and W. Wang,: "A probabilistic approach to fault diagnosis in linear lightware networks", IEEE J. Select. Areas Commun., vol. 11, pp. 14381448, Dec. 1993.

[40] R. Sterritt, A. Marshall, C. Shapcott, and S. McClean,: "Exploring dynamic Bayesian belief networks for intelligent fault management systems", in Proc. IEEE International Conference on Sys-

tems, Man and Cybernetics, vol. 5, Nashville, TN, Oct. 2000, pp. 36463652.

[41] H. Wietgrefe, K. Tuchs, K. Jobmann, G. Carls, P. Fr¨ohlich, W. Nedil, and S. Steinfeld,: "Using neural networks for alarm correlation in cellular phone networks", in Proc. International Workshop on Applications of Neural Networks to Telecommunications, Melbourne, Australia, June 1997, pp. 7.2.97.110.

[42] P. Frohlich, W. Nejdl, K. Jobmann, and H. Wietgrefe,: "Model-based alarm correlation in cellular phone networks", in Proc. IEEE International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, Haifa, Israel, Jan. 1997, pp. 197204.

[43] H. Wietgrefe,: "Investigation and practical assessment of alarm correlation methods for the use in GSM access networks", in Proc. IEEE/IFIP Network Operations and Management Symposium, Florence, Italy, Apr. 2002, pp. 391 403.

[44] M. Guiagoussou and R. Boutaba,: "A TMN framework for faults diagnostic in wireless telecommunication networks", in Proc. IEEE Symposium on Computers and Communications, Alexandria, Egypt, July 1997, pp. 424433.

[45] G. Jakobson and M. Weissman,: "Alarm correlation", IEEE Network, pp. 5259, Nov. 1993.

[46] S. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie,: "High speed and robust event correlation", IEEE Commun. Mag., vol. 34, pp. 8290, May 1996.

[47] R. Barco, V. Wille, and L. Dýez,: "System for automatic diagnosis in cellular networks based on performance indicators", European Trans. Telecommunications, vol. 16, no. 5, pp. 399409, Oct. 2005.

[48] P. Langley, W. Iba, K. Thompson,: "An analysis of Bayesian classifiers", in Proc. 10th National Conference on Artificial Intelligence, 223-228, Menlo Park, AAAI Press, 1992.

[49] T. Halonen, J. Romero, and J. Melero, Eds., "GSM, GPRS and EDGE Performance. Evolution Towards 3G/UMTS", Chichester, England: Wiley, 2003.

[50] R. Barco, R. Guerrero, G. Hylander, L. Nielsen, M. Partanen, and S. Patel,: "Automated troubleshooting of mobile networks using bayesian networks", in Proc. IASTED International Conference on

Communication Systems and Networks (CSN02), Malaga, Spain, Sept. 2002, pp. 105110.

[51] R. Barco, L. Nielsen, R. Guerrero, G. Hylander, and S. Patel,: "Automated troubleshooting of a mobile communication network using bayesian networks", in Proc. IEEE International Workshop on Mobile and Wireless Communications Networks (MWCN02), Stockholm, Sweden, Sept. 2002, pp. 606610.

[52] D. Obradovic, R. Scheiterer,: "Troubleshooting in GSM Mobile Telecommunication Networks Based on Domain Model and Sensory Information", Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, 15th International Conference, Warsaw, Poland, September 11-15, 2005.

[53] J. Pearl,: "Probabilistic reasoning in intelligent systems: Networks of plausible inference", San Francisco, California: Morgan Kaufmann, 1988.

[54] F. Jensen,: "Bayesian Networks and decision graphs. New York, USA", Springer-Verlag, 2001.

[55] R. Scheiterer and D. Obradovic,: "Bayesian Network Modeling Aspects Resulting from Applications in Medical Diagnostics and GSM Troubleshooting". Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, 15th International Conference, Warsaw, Poland, September 11-15, 2005.

[56] R. Barco, V. Wille, L. Díez, P. Lázaro,: "Comparison of probabilistic models used for diagnosis in cellular networks", in Proc. IEEE 63th Vehicular Technology Conference, Melbourne, Australia, May 2006.

[57] R. Barco, P. Lázaro, Luis Díez, Volker Wille,: "Multiple Intervals Versus Smoothing of Boundaries in the Discretization of Performance Indicators Used for Diagnosis in Cellular Networks", Lecture Notes in Computer Science (LNCS), vol.3483, IV, 958-967.

[58] R.M. Khanafer, B. Solana, J. Triola, R. Barco, L. Moltsen, Z. Altman, P. Lazaro,: "Automated Diagnosis for UMTS Networks Using Bayesian Network Approach", in Proc. IEEE Transactions on Vehicular Technology, July 2008, pp. 2451-2461.

[59] R. Khanafer, L. Moltsen, H. Dubreil, Z. Altman, and R. Barco,: "A Bayesian Approach for Automated Troubleshooting for UMTS Networks", Proc. 17th IEEE Intl Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC 06), Aug. 2006.

[60] D. Tikunov, T. Nishimura,: "Automated troubleshooting of mobile networks based on alarm and performance data using Bayesian networks". IEEE Int. Conf. Software, Telecommunications and Computer Networks, Sept. 2007.

[61] J. You,: "Research of Wireless Network Fault Diagnosis Based on Bayesian Networks", Second International Symposium on Knowledge Acquisition and Modeling, 2009, kam, vol. 3, pp.59-64.

LTE introduction References

[62] 3GPP TR 25.913: "V7.1.0 Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN)", Release 7, (2005-09).

[63] 3GPP TS 36.300: "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description", Stage 3 (Release8), Sept. 2007.

[64] E. Dahlman, S. Parkvall, J. Sköld and P. Beming,: " 3G Evolution, HSPA and LTE for Mobile Broadband, Academic Press", London 2007.

[65] 3GPP TR3-061487: "Self-Configuration and Self-Optimization".

[66] 3GPP TR3-071262: "Self-optimization use-case: self-tuning of handover parameters", Orange.

[67] 3GPP TR3-071438: "Load Balancing SON Use case", Alcatel-Lucent.

[68] H. Ekström, A. Furuskär, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, and M. Wahlqvist,: "Technical Solutions for the 3G Long-Term Evolution," IEEE Commun. Mag., vol. 44, no. 3,March 2006, pp. 3845.

[69] 3GPP TR 25.814, V7.1.0, "Physical Layer Aspects for evolved Universal Terrestrial Radio Access (UTRA) (Release 7)" (2006-09).

[70] 3GPP TS 36.211, V0.3.1: "Physical Channels and Modulation (Release 8)" (2007-02).

[71] 3GPP TSG RAN WG3, R3-071494: "Automatic neighbour cell configuration", Aug. 2007.

[72] 3GPP TSG RAN WG3, R3-071819: "On automatic neighbour relation configuration", Octobre. 2007.

[73] R. Nasri, "Paramétrage Dynamique et Optimisation Automatique des Réseaux Mobiles 3G et 3G+s", PhD thesis, University of Pierre and Marie Curie - Paris VI , 2008.

[74] 3GPP TR3-072191: "Measurements for handover decision use case", NTT DoCoMo, Orange, Telecom Italia, T-Mobile, Telefonica, November 2007.

[75] 3GPP TR 36.902: "3GPP E-UTRA and E-UTRAN; Self-configuring and self-optimizing network use cases and solutions", (Release 8)

[76] NGMN, "Recommendation on SON and O&M Requirements", version 1.1, Jul. 2008

[77] 3GPP TR 32.821: "Telecommunication management; Study of Self-Organizing Networks (SON) related OAM interfaces for Home NodeB", (Release 8)

[78] 3GPP TR 36.902: "3GPP E-UTRA and E-UTRAN; Self-configuring and self-optimizing network use cases and solutions", (Release 8)

[79] 3GPP TS 36.314: "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Layer 2 - Measurements", (Release 8)

[80] 3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification", (Release 8)

[81] 3GPP Technical Report TR 25.814: "'Physical layer aspects for the evolved Universal Radio Access (E-UTRA)", Release 7, 2006.

[82] 3GPP Technical Report, TR 25,892: "Feasibility Study for OFDM for UTRAN Enhancement", Release 6, V. 1.1.0, 2004-03.

[83] H. Li and G. Liu, :"OFDM-based broadband Wireless Networks: Design and Optimization", Wiley 2005, ISBN 0471723460

[84] 3GPP TR 32.816: "Study on management of Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC)", (Release 8).

[85] S. Ben Jamaa, Z. Altman, J.-M. Picard, A. Ortega,:"Steered Optimization Strategy for Automatic Cell Planning of UMTS Networks", IEEE 61st Vehicular Technology Confence, 2005 Spring.

[86] S. Ben Jamaa, Z. Altman, A. Ortega and B. Fourestié,:"UMTS design strategies based on indicator matrix approach", IEEE Int. Conf. on Communications, ICC 2004, Paris, France, June 20-24, 2004.

[87] Hosmer, W. David, S. Lemeshow,: "Applied Logistic Regression", 2nd ed.. New York; Chichester, Wiley, 2000.

[88] A.J. Dobson, A.G. Barnett ,:"Introduction to Generalized Linear Models", Third Edition, London: Chapman and Hall/CRC, 2008.

[89] R. Nasri, Z. Altman,:"Handover Adaptation for Dynamic Load Balancing in 3GPP Long Term Evolution Systems",MoMM'2007-The Fifth International Conference on Advances in Mobile Computing and Multimedia, 3-5 December 2007, pp. 145-154.

[90] C. J. Leggetter and P. C. Woodland,: "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", Comput. Speech Lang., vol. 9, pp. 171185, 1995.

[91] D. C. Montgomery, E. A. Peck, G. G. Vining, "Introduction to Linear Regression Analysis", 4th Edition, John Wiley, 2006.

[92] J. S. Milton and C. J.Arnold,: "Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences", 4th Edition, New York: McGraw Hill, 2003.

[93] 3GPP TS 36.300 V8.4.0: "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN)"; Overall description (2008-03)

[94] J. Mo and J. Warland, *Fair end-to-end window based congestion control, IEEE transactions networking*, vol. 8, pp. 556566, October 2000.

[95] R. Combes, Z. Altman, and E. Altman, *On the use of packet scheduling in self-optimization processes: application to coverage-capacity optimization*, in WiOpt 2010, Avignon, France, June 2010.

[96] S. Ben Jamaa, H. Dubreil, Z. Altman, A. Ortega,: "Quality indicator matrices and their contribution to WCDMA network design", IEEE Trans. on Vehicular Technology, vol. 54, May 2005, pp.1114-1121.

[97] P. McCullagh , J. A. Nelder,: "Generalized Linear Models", Second Edition, Chapman and Hall, London, 1989.

[98] P. D. Jong, G. Z. Heller,: "Generalized Linear Models for Insurance Data", Cambridge University Press, 2008.

[99] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling,: "Numerical Recipes in C: The Art of Scientific Computing", Cambridge University Press, 1992.

[100] A. Samhat, Z. Altman, M. Francisco, B. Fourestie,: "Semi-dynamic simulator for large scale heterogeneous wireless net-

works", International Journal on Mobile Network Design and Innovation (IJMNDI), Vol. 1, no. 3-4, 2006, pp. 269-278.

[101] R. Nasri, Z. Altman and H. Dubreil,: "WCDMA downlink load sharing with dynamic control of soft handover parameters",IEEE International Symposium VTC, Melbourne, Australia, 7-10 May, 2006.

Author's Publications

- M. I. Tiwana, B. Sayrac, Z. Altman,: "Statistical Learning for Automated RRM: Application to eUTRAN Mobility", IEEE Int. Conf. on Communications, ICC 2009, Dresden, Germany, June 14-18, 2009.
- M. I. Tiwana, B. Sayrac, Z. Altman, T. Chahed,: "Troubleshooting of 3G LTE mobility parameters using iterative statistical model refinement", IFIP Wireless Days 2009, Paris, Dec. 2009
- M. I. Tiwana, B. Sayrac, and Z. Altman,: "Statistical approach for automated troubleshooting: application to LTE interference mitigation", IEEE Trans. on Vehicular Technology, Issue:99, 2010.
- M. I. Tiwana, Z. Altman, and B. Sayrac,: "Enhancing RRM optimization using a priori knowledge for automated troubleshooting", WIOPT 2010 (IEEE Xplore) workshop WCN3, Avignon (France), May 2010.
- M. I. Tiwana, B. Sayrac, Z. Altman, T. Chahed,: "Learning-based Automated Healing: Application to Mobility in 3G LTE Networks", PIMRC 2010, Istanbul, September 2010.

Appendix A

LTE interference model

Starting from the interference coordination scheme, presented in section 2.4, we assume that the spectral band is composed of C resource blocks, one third of the band is reserved for the cell edge users and the rest is for cell centre users.

The resource allocation is made according to users' received signal quality. The users with the worst quality signal are assigned to cell edge band. When the cell edge band is full, the remaining unassigned users are assigned to cell center band. The eNB transmit power in each cell-edge resource block equals the maximum transmit power P. To reduce intercell interference, the eNB transmit power in the cell-centre band must be lower than P. Let εP (where $\varepsilon \leq 1$) be the transmit power in the cell-centre band.

The interference should be determined for two different users according to their positions: the cell-centre user and the cell-edge user. Let m_c and m_e be two users connected to a cell k. the mobile m_c uses the central band whereas m_e uses the cell-edge band. Let denote the interference matrix between cells, where the coefficient $\Lambda(i, j)$ equals 1 if cells i and j use the same cell-edge band and zero otherwise.

For cell-edge user m_e, the interference comes from users in the cell centre of the closest adjacent cells and from the cell-edge user in other cells. The mobile m_e connected to the cell k and using one resource block in the cell-edge band, receives an interfering signal from a cell i equals

$$I_{i,m_e} = \left((1 - \Lambda(k,i))\, \beta_i^c P_i + \Lambda(k,i) \beta_i^e P_i \right) \frac{G_{i,m_e}}{L_{i,m_e}} \tag{A.1}$$

where P_i is the downlink transmit power per resource block of the cell i. G_{i,m_e} and L_{i,m_e} are respectively the antenna gain and the path loss

between cell i and the mobile station m_e. The factor β_i^c (respectively β_i^e) is the probability that the same resource block in the cell-centre band (respectively the cell-edge band) is used at the same time by another mobile connected to the cell i.
Since the analysis considers a long time scale (of the order of seconds), the interference is averaged. So, the factor β_i^c is the percentage of users using cell-centre band and β_i^e is the percentage of those using cell-edge band

$$\beta_i^c = \frac{\text{occupied resource blocks in cell center band}}{\text{total capacity of cell center band}} = \frac{M_c}{2C/3}$$

$$\beta_i^e = \frac{\text{occupied resource blocks in cell edge band}}{\text{total capacity of cell edge band}} = \frac{M_e}{C/3}$$

M_c and M_e are the number of resource blocks used in the cell centre and cell edge respectively, and the sum $M_c + M_e$ is the total number of resource blocks used in the cell. Let χ_i be the load of cell i given by

$$\chi_i = \frac{M_c + M_e}{C} \tag{A.2}$$

Define the factor α_i as the proportion of traffic served in the cell-edge band, $\alpha_i = M_e/(M_c + M_e)$. The factors β_i^c and β_i^e become respectively

$$\begin{aligned} \beta_i^c &= \frac{3(1-\alpha_i)(M_c+M_e)}{2C} = \frac{3(1-\alpha_i)\chi_i}{2} \\ \beta_i^e &= \frac{3\alpha_i(M_c+M_e)}{C} = 3\alpha_i\chi_i \end{aligned}$$

Appendix B

Pseudo code of the iterative KPI tuning algorithm

Initialization: $HM_{i,j} = \begin{cases} 6dB & \forall \text{BS}_i \text{ and } BS_j \text{ that are geographical neighbours} \\ 10dB & \forall \text{BS}_i \text{ and } BS_j \text{ that are not geographical neighbours} \end{cases}$

$\triangle T = 50sec$

Iteration:For $round = 1$ to $MaxRound$
Collect KPI data corresponding to $\triangle$T
For i=1 to $NumberOfBaseStations$
If $KPI_i \leq \nu_{KPI}$ then:
For k=1 to $NumberOfNeighboursOfBS_i$
If $KPI_k > \nu_{KPI}$ then put BS_k into the set M_i.
j=index of the BS with highest KPI in M_i
For k=1 to $NumberOfElementsOfM_i$
$l = BS$ index of the k^{th} element in M_i
If $l \neq j$ then calculate η_{jl}=$\frac{|KPI_j - \nu_{KPI}|}{|KPI_l - \nu_{KPI}|}$
Calculate $\triangle_{ij}$ as

$$\triangle_{ij} = \frac{a_{HM,1}}{1 + \exp(\frac{HM_{i,j} - b_{HM,1}}{c_{HM,1}})} \frac{a_{KPI,1}}{1 + \exp(\frac{KPI_i - b_{KPI,1}}{c_{KPI,1}})} \frac{1}{1 + \sum_{j \neq l} \frac{1}{\eta_{jl}}}$$

For $k = 1$ to $NumberOfElementsOfM_i$

$l = BS$ index of the k^{th} element in M_i
If $l \neq j$ then calculate $\triangle_{il} = \frac{\triangle_{ij}}{\eta_{jl}}$

For $k = 1$ to $NumberOfElementsOfM_i$
$l = BS$ index of the k^{th} element in M_i

$$\mathrm{HM}_{i,l} = \begin{cases} HM_{i,l} + \triangle_{il} & \text{if } HM_{i,l} + \triangle_{il} \leq HM_{max} \\ HM_{max} & \text{if } HM_{i,l} + \triangle_{il} > HM_{max} \end{cases} \quad \text{(B.1)}$$

If $KPI_i > \nu_{KPI}$ then:
For $k = 1$ to $NumberOfNeighboursOfBS_i$
If $KPI_k < \nu_{KPI} - \delta_{KPI}$ then put BS_k into the set N_i
j=index of the BS with lowest KPI in N_i
For $k = 1$ to $NumberOfElementsOf\ N_i$
l=BS index of the k^{th} element in N_i
If $l \neq j$ then calculate η_{jl}=$\frac{|\nu_{KPI} - KPI_j - \delta_{KPI}|}{|\nu_{KPI} - KPI_l - \delta_{KPI}|}$
Calculate $\triangle_{ij}$ using (4.7)

$$\triangle_{ij} = \frac{a_{HM,2}}{1 + \exp(-\frac{HM_{i,j} - b_{HM,1}}{c_{HM,2}})} \frac{a_{KPI,2}}{1 + \exp(-\frac{KPI_i - b_{KPI,2}}{c_{KPI,2}})} \frac{1}{1 + \sum_{j \neq l} \frac{1}{\eta_{jl}}}$$

For $k = 1$ to $NumberOfElementsOfN_i$
l=BS index of the k^{th} element in N_i
If $l \neq j$ then calculate $\triangle_{il} = \frac{\triangle_{ij}}{\eta_{jl}}$
For $k = 1$ to $NumberOfElementsOfN_i$
l=BS index of the k^{th} element in N_i

$$\mathrm{HM}_{i,l} = \begin{cases} HM_{i,l} - \triangle_{il} & \text{if } HM_{i,l} - \triangle_{il} \geq HM_{min} \\ HM_{min} & \text{if } HM_{i,l} - \triangle_{il} > HM_{min} \end{cases}$$

The parameters of the algorithm (ΔT , MaxRound,ν_{KPI}, δ_{KPI},HM$_{min}$, HM_{max} etc.) are determined using expert knowledge. The function of the sigmoid functions, $\frac{a}{1+exp(-\frac{x-b}{c})}$, used in calculating $_{ij}$'s is to speed up/down the changes in the HM values according to the operating point (i.e. the KPI_i and $HM_{i,j}$). Therefore, the constants like $a_{KPI,1}$, $a_{KPI,2}$, $b_{KPI,1}$, $b_{KPI,2}$, $c_{KPI,1}$, $c_{KPI,2}$, $a_{HM,1}$, $a_{HM,2}$, $b_{HM,1}$, $b_{HM,2}$, $c_{HM,1}$, $c_{HM,2}$ are adjusted according to

the rate of increase/decrease of the HM values. The above algorithm is based on a single KPI whose bad region lies above a threshold (e.g. BCR, DCR, Load etc.). For a different KPI whose bad region lies below a threshold (e.g. Throughput), all the inequalities concerning the KPI must be reversed and the sigmoid functions must reverse their signs that are inside the exponentials.