Enhancing RRM optimization using a priori knowledge for automated troubleshooting

Moazzam Islam Tiwana, Zwi Altman, Berna Sayrac

Enhancing RRM optimization using a priori knowledge for automated troubleshooting

Moazzam Islam Tiwana, Zwi Altman and Berna Sayrac
Orange Labs, RESA/NET
38-40 Rue du General Leclerc
92794 Issy-Les-Moulineaux Cedex 9, France
Email: {moazzamislam.tiwana,zwi.altman,berna.sayrac}@orange-ftgroup.com

***Abstract*—The paper presents a methodology that combines statistical learning with constraint optimization by locally optimizing Radio Resource Management (RRM) or system parameters of poorly performing cells in an iterative manner. The statistical learning technique used is Logistic Regression (LR) which is applied on the data in the form of RRM-KPI (Key Performance Indicator) pairs. LR extracts closed form (functional) relations, known as the *model*, between KPIs and RRM parameters. This model is then processed by an optimization engine which proposes a new RRM parameter value. The RRM parameter value is reinserted in the network/simulator to generate corresponding KPI vector constituting generated RRM-KPI pair. First, only the *a priori* RRM-KPI pairs which are based upon the *a priori* model information are used for the model extraction. Then, as the optimization iterations progress, the generated pairs are given more importance in model extraction and the model is iteratively refined. The use of the *a priori* knowledge has the advantage of avoiding wrong initial models due to noisy data, allows much faster convergence and makes it more suitable for the off-line implementation. The proposed method is applied to troubleshoot an Inter-Cell Interference Coordination (ICIC) process in a LTE network which is based on soft-frequency reuse scheme.**

Keywords: ICIC, Statistical learning, logistic regression, automated troubleshooting, LTE.

I. Introduction

Next generation radio access networks (RAN) aim at increasing bandwidth for future services and applications with high quality of service (QoS) provisioning. The RAN landscape will become more and more heterogeneous with co-existing and co-operating technologies and deployment scenarios (macro, micro, pico and femto cell structures). In this context network management becomes strategic for operators that need to assure efficient use of available resources while limiting the cost of operation.

Troubleshooting is a central activity of network management. It comprises of three functionalities [1][2]:

- fault detection, i.e. detecting failures or poor performance as soon as they occur;
- fault-diagnosis, i.e. determining the cause of failure or of poor performance, and
- fault recovery or healing, i.e. repairing the problem.

Efficient troubleshooting allows fast reaction to problems, reduction of infrastructure equipment down-time and period of poor QoS provisioning. Hence troubleshooting is an important activity for the operator, and has a direct impact on churn reduction and on operator revenues.

This paper focuses on automated fault recovery or automated healing. It is assumed that the fault cause has been diagnosed, and is related to bad RRM parameter settings. The automated healing is of local character, and aims at adjusting the RRM parameter of a base station (BS) or eNodeB (eNB) using Key Performance Indicators (KPI) from the eNB and its neighbours. Previous works have studied the problem of local type (or steered) optimization (see for example [3], [4]). These works propose solutions based on the use of network simulator or automatic cell planning tool that combines the interference matrix with optimization heuristics. In certain cases, solutions fully based on network measurements are required. Such solutions are carried out off-line using data from the Operation and Management Center (OMC) or from capture tools.

In a typical troubleshooting process, the optimization expert analyzes KPIs and then proposes a new RRM parameter which is applied to the problematic BS. The BS operates with the new parameter during a period long enough, typically a day, to have statistically significant results that allow to assess the BS 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, a novel approach for automated troubleshooting based on statistical learning has been recently introduced ([5], [6]), and denoted in [6] as Statistical Learning Automated Healing (SLAH). This approach assumes that the model describing the functional relations between KPIs and RRM parameters is simple (i.e. is not described by a multi-modal function) and can be described using statistical regressions derived from the measured data. In [5], linear regression has been assumed whereas in [6], Logistic Regression (LR) has been used. The merit of logistic regression is that it can account for saturation effects that often appear in communication network KPIs such as blocking rate, dropping rate etc. 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.

The purpose of the present paper 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, several 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 paper 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 BSs 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 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.

Throughout the paper, the terms eNB and base station can be used interchangeably. Similarly, the terms mobile and user have the same meaning. The rest of the paper is organized as follows: Section II presents the system model for automated healing. The incorporation of the *a priori* knowledge to the automated healing algorithm is described in Section III. The Section IV explains the ICIC System Model. The adaptation of the SLAH to the problem of ICIC in a LTE network is summarized in Section V, followed by numerical results in Section VI. Section VII concludes the paper.

II. System Model for Automated Healing

This section describes the system model for the automated healing. It is assumed that the initial two steps of the troubleshooting process, namely fault detection and fault diagnosis, have already been done. Let's suppose that the fault cause has been diagnosed as a RRM parameter whose value has degraded the performance of the eNodeB (eNB). The purpose of the SLAH algorithm is to devise a robust mechanism which using local information from an eNB and its neighbors, can iteratively optimize the value of the RRM parameter that has caused the degraded performance of the cell. The block diagram of the proposed algorithm is presented in Figure 1.

The system model comprises of four blocks:

Initialization block

The initialization block can have either of two functions

- In the absence of the *a priori* RRM-KPI pairs, provide the initial RRM parameter values to the faulty eNB in the Network/Simulator block in order to generate the corresponding KPIs. These generated RRM-KPI couples are used by the Statistical Learning block.
- To provide noise-free *a priori* RRM-KPI pairs to Statistical Learning block for the initial model estimation.

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 pairs to extract closed form (functional) relations relating the KPIs to the RRM parameters. These functional relations constitute the statistical model or for brevity, the *Model*.

In the case of the *a priori* knowledge incorporation, the *Model* is estimated from the data points, taking into account the weight assigned to each point according to its importance in the model estimation process.

Optimization block

The Optimization block calculates the optimal RRM value using the current statistical model. It minimizes a cost function of certain KPIs under constraints imposed on other KPIs.

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 new iteration, on the average, the model precision improves and is used by the optimization block to find a better value for the RRM parameter. In the following, the Statistical Learning and the Optimization blocks are described in detail.

A. Statistical Learning

The statistical learning approach used in the automated-healing is based on Logistic Regression (LR) [7][8]. The LR Model belongs to a category of models known as the

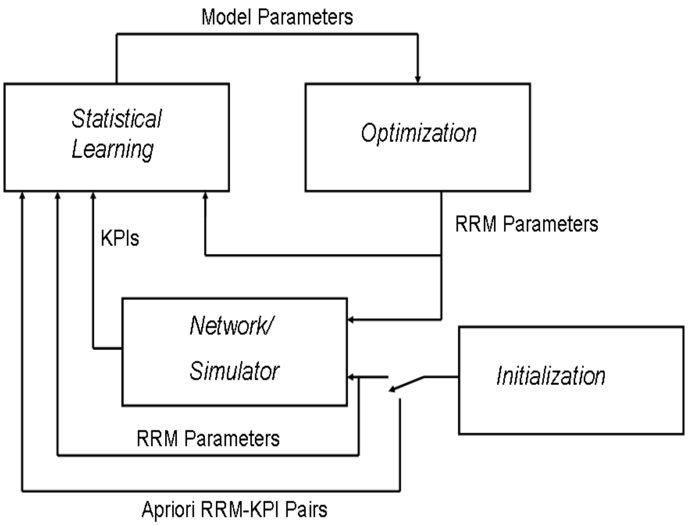


Figure 1. SLAH block diagram.

Generalized Linear Models (GLM)[9]. The LR establishes the statistical model by extracting the functional relations between the KPIs and the RRM parameter. LR fits the data into the functional form of $logistic\ function$ denoted as f_{log}

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

where z can vary from $-\infty$ to ∞ and $f_{log}(z)$ from 0 to 1 (see Figure 2). One can see from Figure 2 that $f_{log}(z)$ can describe saturation effects at its extremities as often encountered in KPIs in communication networks.

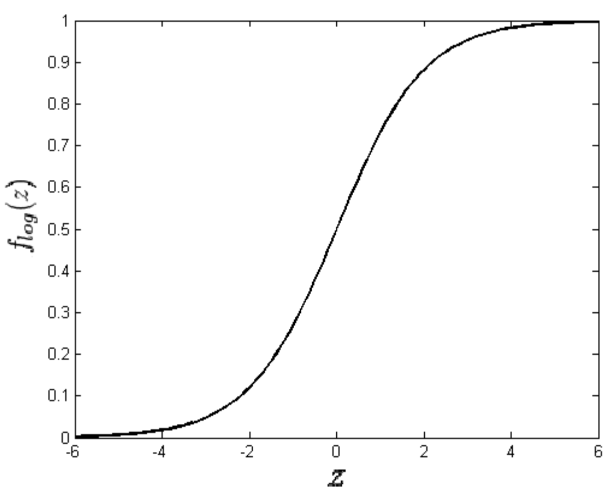


Figure 2. The logistic function.

In our work, the KPI is a dependent variable denoted as y and the RRM parameter is an explanatory variable denoted as x. The KPIs chosen in our work are mean Block Call Rate (BCR) of mobiles accessing an eNB and mean File Transfer Time (FTT) of mobiles attached to an eNB over full simulation period for a given value of 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). LR models $y_{m,i}$ as follows:

$$y_{m,i} = f_{log}(\eta_{m,i}) + \epsilon_i \tag{2}$$

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 III. Let there be n sample values for y_m corresponding to n values of x. In order to write (2) 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

$$\hat{Y_m} = f_{log}(X\beta_m) \tag{3}$$

Here, $\hat{Y_m}$ represents the estimated Y_m using LR. It is noted that the dependent variables in the LR are between 0 and 1. Hence, Y_m is normalized between 0 and 1 before the application of LR. The calculated $\hat{Y_m}$ values can be subsequently denormalized.

It follows from (3) that functional relation between $\hat{y_m}$ i.e., y_m estimated by LR, and x can be written as

$$\hat{y_m} = f_{log}(x'\beta_m) \tag{4}$$

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

B. 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

$$U = \sum_{m \in A_o} w_m \hat{y}_m \tag{5}$$

where

- $\hat{y}_m$ has the functional relation form as in (4).
- w_m is the weight given to $\hat{y}_m$.

The optimization problem is formulated as follows:

$$\hat{x} = argmin_x U(x) \tag{6}$$

subject to

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

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

III. 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 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 LR model is defined as follows: We follow the same notation for Y_m and X as in previous section. The β_m vector is estimated using Maximum likelihood Estimation (MLE). 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.

In order to find the maximum likelihood estimation of $\beta_{m,b}$ where $b = \{1, 2\}$, the derivative of the likelihood function l with respect to $\beta_{m,b}$ is taken [10] and equated to 0:

$$\frac{\partial l}{\partial \beta_{m,b}} = 0 \tag{7}$$

After simplification as explained in [10], we get the equation

$$\frac{\partial l}{\partial \beta_{m,b}} = \sum_{i=1}^{n} \frac{(y_{m,i} - \hat{y}_{m,i})X_{ij}}{var\, \hat{y}_{m,i}} \frac{\partial \hat{y}_{m,i}}{\partial \eta_{m,i}} = 0 \qquad \text{for } b = 1, 2 \tag{8}$$

where $\eta_{m,i} = x_i' \beta_m$ and $\hat{y}_{m,i} = f_{log}(\eta_{m,i})$. Equation (8) can be expressed in the matrix notation as

$$X^T V_m (Y_m - \hat{Y_m}) = 0 \tag{9}$$

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 (9), 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 Newton-Raphson method. 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:

$$f(\beta_m^t) + \frac{\partial f(\beta_m^t)}{\partial \beta_m^T} (\beta_m - \beta_m^k) \tag{10}$$

where t is the iteration index. Equating (10) to zero, we can approximate β_m^{t+1} as follows:

$$\beta_m^{t+1} = \beta_m^t - \frac{f(\beta_m^t)}{\left(\frac{\partial f(\beta_m^t)}{\partial \beta_m^T}\right)} \tag{11}$$

Starting from an initial β_m^0 and iteratively solving equation (11), β_m converges to a solution. 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[10] and is usually replaced by its expectation

$$\begin{aligned} \mathcal{I}(\beta) &= \mathbb{E} \frac{\partial l(\beta_m)}{\partial \beta_m} \frac{\partial l(\beta_m)}{\partial \beta_m^T} \\ &= \mathbb{E}[X^T V_m (Y_m - \hat{Y_m}) \left\{ X^T V_m (Y_m - \hat{Y_m}) \right\}^T] \\ &= X^T W_m X \end{aligned} \tag{12}$$

where $W_m = V_m (\text{cov}\, Y_m) V_m$ is a diagonal matrix with diagonal elements $[W_m]_{ii} = \frac{1}{\text{var}\, \hat{y}_{m,i}} (\frac{\partial \hat{y}_{m,i}}{\partial \eta_i})^2 = nt_i * \hat{y}_{m,i} * (1 - \hat{y}_{m,i})$, where $nt_{m,i}$ is the number of Bernoulli trials for the i^{th} data point. Hence, (11) becomes.

$$\beta_m^{t+1} = \beta_m^t + (X^T W_m^t X)^{-1} X^T V_m^t (Y_m - \hat{Y_m}^t)$$

which takes the final form as

$$\beta_m^{t+1} = S_m^t X^T W_m^t z_m^t \tag{13}$$

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 $[U_m]_{ii}^t = \frac{\partial \eta_{m,i}^t}{\partial \mu_{m,i}^t} = \frac{1}{\hat{y}_{m,i}^t (1 - \hat{y}_{m,i}^t)}$.

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 (13) 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. $z_m \leftarrow f_{log}^{-1}(y_m)$
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. $\mu_m \leftarrow f_{log}(\eta_m)$
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

IV. ICIC System Model

Consider an ICIC scheme for downlink transmissions that combines two resource allocation mechanisms: Physical Resource Block (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 3 presents the power-frequency allocation model in a seven adjacent cell layout.

The frequency band is divided into three disjoint subbands. 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 neighbours 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 generated using pilot channel signal strengths

$$q_u = \frac{Pr_{su}}{\sum_{j \neq s} Pr_{ju} + \sigma_z^2}, \tag{14}$$

where s stands for the serving eNB of user u, 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 Signal to Interference plus Noise Radio (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.

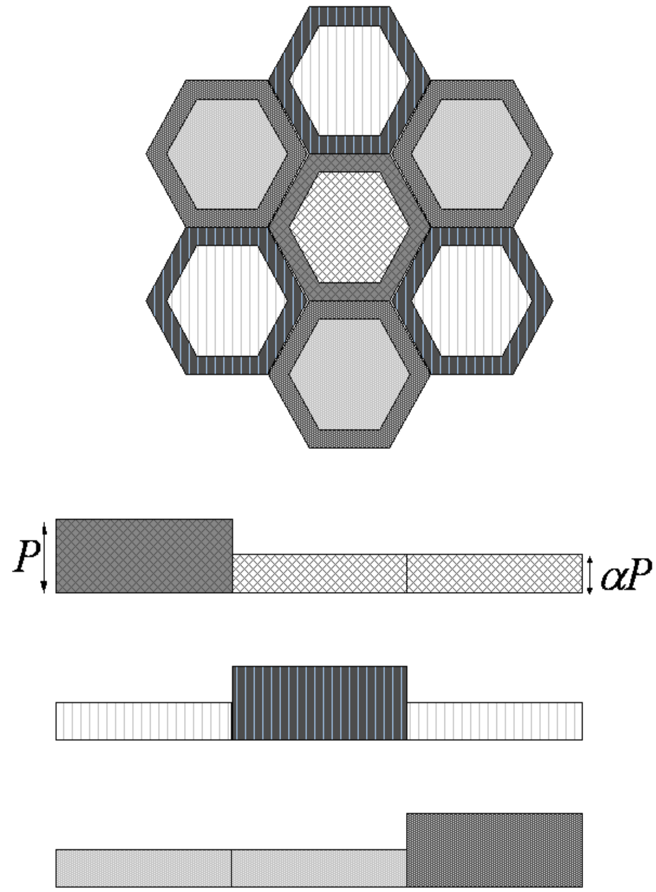


Figure 3. ICIC System Model.

V. Troubleshooting 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 of 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 eNB_c and can be written in terms of the interference matrix element I_{cj} [3][4]. 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 mobile attached to eNB_c and generated by 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:

$$\alpha_j = \mathrm{g}_j(\alpha_s) = \alpha_i + (1 - \alpha_s)(1 - \frac{I_{cj}}{I_{cs}}) \tag{15}$$

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:

$$U = FTT_c + \sum_{j \in NS1} \omega_j FTT_j \tag{16}$$

It is noted that FTT_j is a function of α_j and hence, via equation (15), 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

$$\omega_j = \frac{I_{cj}}{\sum_{k \in NS1} I_{ck}} \tag{17}$$

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

$$\alpha_s = argmin_{\alpha'_s} U(\alpha'_s) \tag{18}$$

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 (16) and (18) are given in the form of the LR function (4) obtained using the LR. 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 (18). In this case, instead of (18) our optimization objective is given as

$$\alpha_s = argmax_{\alpha'_s}(BCR_c | (BCR_c = BCR_j)) \ ; \ j \in NS1 \tag{19}$$

Here, α_s signifies the value for which BCR of eNB_c and the BCR of eNB_j having worst BCR value, become equal.
Denote a *data point* p_n^j as the vector

$$p_n^j = (\alpha_j, FTT_j, BCR_j)_n \ ; \ j \in c \cup NS1 \tag{20}$$

In the SLAH algorithm, α_c remains fixed and is not subject to optimization; α_j, $j \in NS1$, satisfies the equation (15); and the set of n data points for an eNB_j, $j \in c \cup NS1$, is denoted by P_n^j. During each optimization iteration one data point is added, hence, n also denotes optimization iteration index.

The SLAH algorithm is written below:

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 LR 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 (15) *and* (18) *or* (19)*)*
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 (equation (20)*)*
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
End Repeat

VI. ICIC Troubleshooting use case

A. Simulation Scenario

A LTE network comprising 45 eNBs in a dense urban environment is depicted in Figure 4. Downlink transmissions

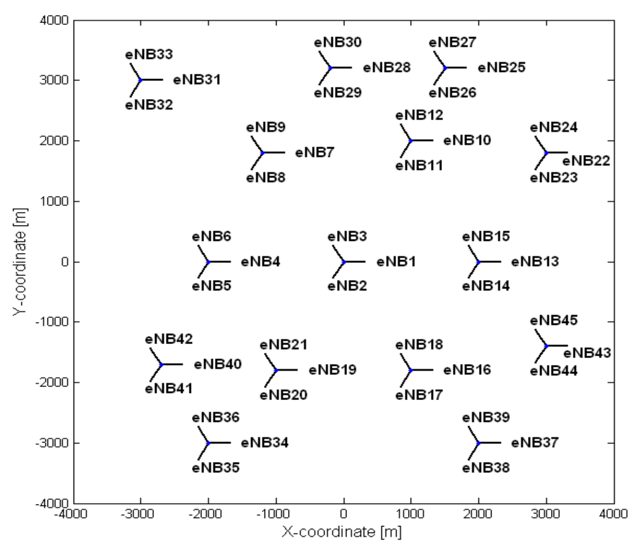


Figure 4. LTE network layout with 45 eNBs.

are considered here. The simulation parameters are listed in Table I. A MATLAB LTE simulator described in [11] has been used. The simulator performs correlated Monte Carlo snapshots with a time resolution of a second to account for the time evolution of the network. The principles of a semi-dynamic simulator are described in [12].

For each new value of α the simulator runs for 2500 time steps (seconds) to allow the convergence of the processed

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 I
THE SYSTEM LEVEL SIMULATION PARAMETERS.

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 (15) and (17) 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. This reference solution is chosen according to the results in [6] and will be used as a starting point for the automated healing process. The default α value is determined by simultaneously varying the α of all eNBs, uniformly from 0.0125 to 1, in the steps of 0.0125 and assessing the network performance in terms of mean BCR and mean FTT. In the α interval from 0.5 to 0.7 the mean BCR and mean FTT have minimum values. The value of $\alpha = 0.5$ is selected due to the smaller inter-cellular interference and the minimum energy consumption in the network.

B. Troubleshooting Scenario

A problematic eNB with the worst performance in the reference network of Figure 4, 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 $eNB_{c=13}$.

C. 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 [6], as shown in Figures 5 and 6 for BCR and FTT KPIs respectively.

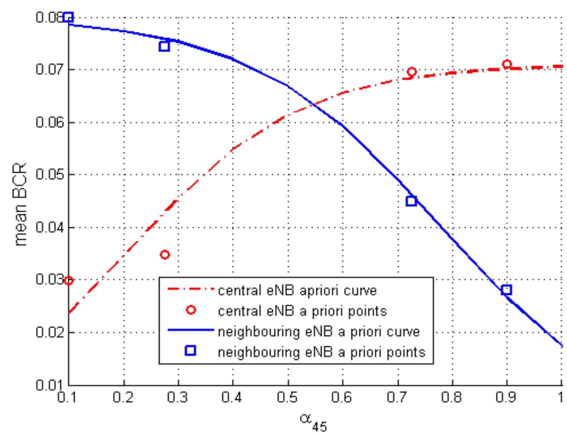


Figure 5. *A priori* curves and data points for BCR as a function of $\alpha_{s=45}$.

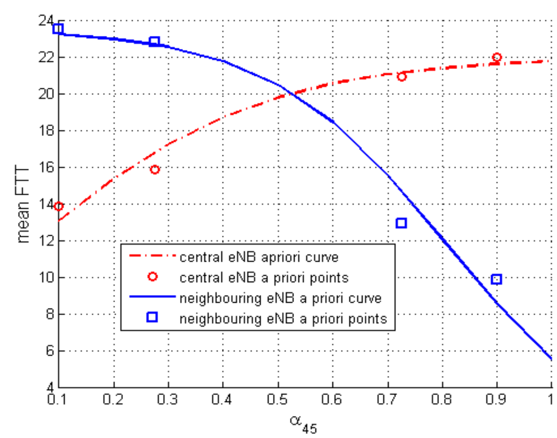


Figure 6. *A priori* curves and data points for FTT as a function of $\alpha_{s=45}$.

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

The matrix A is recalculated before each optimization iteration. Let n be the number of data points at each optimization

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 II
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 III
SHOWS THE α VALUES CALCULATED DURING 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.

D. Results

Table III 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 7 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 7 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 8 and Figure 9 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.

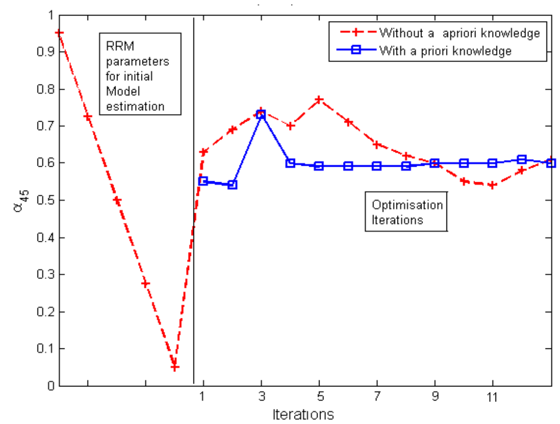


Figure 7. Figure showing the convergence of $\alpha_{s=45}$ with and without *a priori* knowledge incorporated algorithm.

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 troubleshooting 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.

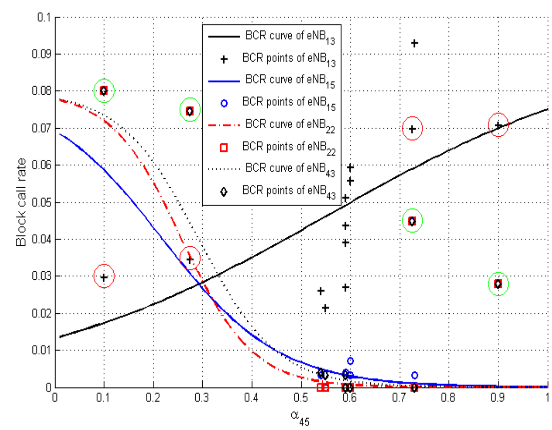


Figure 8. Mean BCR values of $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ along with the corresponding regression curves as a function of $\alpha_{s=45}$.

The BCR of $eNB_{c=13}$ improves from 0.0528 to 0.0396 and its FTT decreases from 16.9362 to 15.1480. Figures 10(a) and 10(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 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

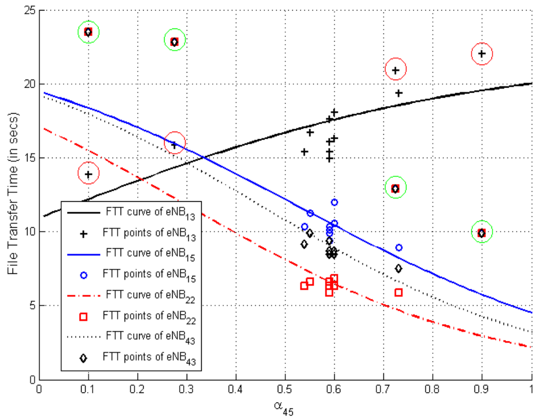
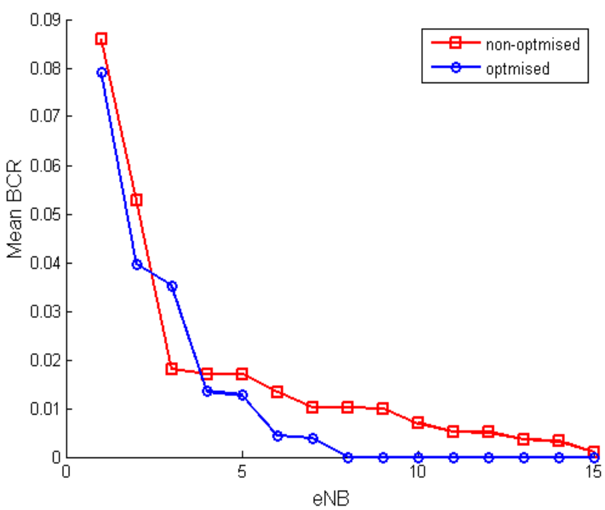
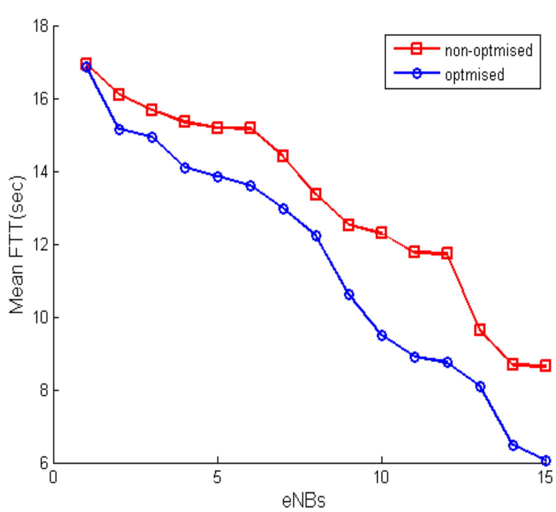


Figure 9. Mean BCR values of $eNB_{c=13}$, $eNB_{j=15}$, $eNB_{j=22}$ and $eNB_{j=43}$ along with the corresponding regression curves as a function of $\alpha_{s=45}$.



(a)



(b)

Figure 10. KPIs for the eNB in the evaluation zone in descending order for mean BCR (a) and mean FTT (b).

observed that in the abscence 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.

VII. Conclusion

This paper shows how the *a priori* knowledge can be introduced into- and improve the performance of the Statistical Learning Automated Healing (SLAH) method presented in [6]. The statistical model describes the closed form relations between the RRM parameters and the KPIs, which are obtained from logistic regression. It has been shown that the *a priori* knowledge can avoid an initially bad statistical model, and hence accelerate the convergence of the SLAH algorithm. The improved SLAH algorithm can converge in a few iterations and is well suited for a real network implementation, e.g. in an OMC, as an off-line process. The automated healing approach has been successfully applied to heal an ICIC parameter of an eNB with degraded performance due to an excess of inter-cell interference in a LTE network.

References

[1] R. Barco, V. Wille, and L. Diez: *System for automatic diagnosis in cellular networks based on performance indicators*, European Trans. Telecommunications, vol. 16, no. 5, Oct. 2005, pp. 399-409.

[2] 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 (MWCN'02), Stockholm, Sweden, Sept. 2002, pp. 606-610.

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

[4] 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.

[5] 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.

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

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

[8] W. D. Hosmer, S. Lemeshow: Applied Logistic Regression, 2nd Edition, Chichester, Wiley, New York, 2000.

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

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

[11] 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.

[12] A. Samhat, Z. Altman, M. Francisco, B. Fourestie: *Semi-dynamic simulator for large scale heterogeneous wireless networks*, International Journal on Mobile Network Design and Innovation (IJMNDI), Vol. 1, no. 3-4, 2006, pp. 269-278.