A Sparse Sampling Algorithm for Self-Optimisation of Coverage in LTE Networks

Ajay Thampi†, Dritan Kaleshi†, Peter Randall‡, Walter Featherstone‡, Simon Armour†
†Department of Electrical and Electronic Engineering, University of Bristol, UK
‡Nokia Siemens Networks, Swindon, UK
†Email: {Ajay.Thampi.10,Dritan.Kaleshi,Simon.Armour}@bristol.ac.uk

***Abstract*—Coverage optimisation is an important self-organising capability that operators would like to have in LTE networks. This paper applies a Reinforcement Learning (RL) based Sparse Sampling algorithm for the self-optimisation of coverage through antenna tilting. This algorithm is better than supervised learning and Q-learning based algorithms as it has the ability to adapt to network environments without prior knowledge, handle large state spaces, perform self-healing and potentially focus on multiple coverage problems.**

I. Introduction

Cellular operators are faced with the arduous task of configuring and maintaining their networks and with the advent of LTE and LTE-Advanced, it has become increasingly important for them to have self-organising capabilities in their networks. This paper focuses on the problem of self-optimisation of coverage in LTE networks by adjusting the antenna tilt. Some of the main challenges when implementing a self-optimising network, as identified in [1] and [2], are:

1) There are possibly infinitely many states that a constantly changing cellular network can be in and it is difficult to have a well-defined mapping from the current state of a network (configuration of all the network elements) to a possible action (antenna tilt for instance) that can improve that state.
2) It is not possible to do a full search of all possible states that the cellular network can be in and a trial-and-error approach to find the optimum solution can be detrimental to the network performance.

For the self-optimisation of coverage through antenna tilting, two major classes of machine learning algorithms have been studied - supervised learning [1] and reinforcement learning (RL) [3].

In [1], a supervised learning algorithm called Case Based Learning (CBL) was studied. A limited number of training examples (or cases) was collected from the network by a central server and stored in memory. These training examples were measurements (e.g., received power, signal to interference plus noise ratio - SINR, call drops) and they were applied to new states of the network. The optimum antenna tilt was then determined using the k-nearest neighbour algorithm where the case that closely matched the current state was chosen. Although [1] does not state what the minimum number of training examples must be, different sets of examples would have to be collected for different network environments.

In [3], optimisation of coverage was done using a Fuzzy Reinforcement Learning (RL) approach, namely Q-Learning. Q-Learning is a technique where the entire state space is searched to find the optimum solution. Since the cellular network environment has possibly infinite states or configurations it can be in, this number was drastically reduced in [3] through *fuzzification*. Optimisation was then done in a distributed manner. Although the fuzzification step has the advantage of reducing the state space and the noise in measurements, it reduces the flexibility of the algorithm and the ability to focus on multiple coverage problems.

In this paper, an RL-based Sparse Sampling algorithm is applied that overcomes the problems in [1] and [3], namely, the inability to adapt to network environments without prior knowledge and the inability to handle a large set of network configurations. A centralised architecture is adopted and the algorithm is shown to have self-optimising and self-healing capabilities. Three coverage problems are identified and the measurements that can be used to detect them are discussed. The use of Minimisation of Drive Test (MDT) reports as specified in [4] is also explored.

The paper is structured as follows. Section II defines the coverage problems that will be detected. Section III discusses the measurements that can be used to detect those problems. Section IV then gives a background on RL and describes the Sparse Sampling algorithm applied in this paper. Section V describes the architecture adopted before going on to analyse the simulation results in Section VI. Section VII concludes the work and makes recommendations for future work.

II. Problem Definition

A. Coverage Problems

Three coverage problems will be considered in this paper which are summarised in Table I [4].

TABLE I
Coverage Problems

Problem	Definition
Coverage hole	Occurs at areas where signal strength is less than a threshold - caused by physical obstructions and unsuitable antenna parameters
Weak coverage	Occurs when the received power of serving cell is below a level needed to maintain a planned performance requirement
Pilot pollution	Occurs when the coverage of different cells overlap a lot, thereby increasing interference

978-1-4673-0762-8/12/$31.00 ©2012 IEEE

B. Modelling Antenna Tilting

Antenna tilt is defined as the angle of the main beam below the horizontal plane [5]. When the antenna is down-tilted from its current tilt, then the change in angle is positive. If, on the other hand, the antenna is up-tilted, then the change in angle is negative.

For tri-sector cell sites, the horizontal and vertical radiation patterns ($A_H(\phi)$ and $A_V(\theta)$) given in [6] can be combined into a 3D radiation pattern as follows.

$$A(\phi, \theta) = -min[-A_H(\phi) - A_V(\theta), A_m] \text{ [6]}$$
$$\text{where, } A_m = 25dB$$

The pathloss model (with log-normal shadowing) must also be augmented to include the loss or gain in power due to the antenna tilt. Intuitively, to mitigate the coverage hole and weak coverage problems, the antenna must be up-tilted. By up-tilting, power is spread over a wider area and can fill any coverage holes or weak areas. To mitigate the pilot pollution problem, the antenna must be down-tilted. By down-tilting, power is focused to a smaller area and this reduces the interference caused to its neighbouring cells.

Antennas can be tilted either electrically or mechanically. As explored in [5] and [7], the radiation patterns are different for both techniques but better coverage optimisation is seen for electrical tilting. Therefore, in this paper, only electrical tilting is considered.

III. Measurements

In order to detect the three coverage problems, measurements have to be obtained from the user equipment (UE) and the base station (called eNB in LTE). Each eNB serves a tri-sector cell site, with three antennas serving each of the three cells. The eNB has to aggregate measurements collected from its cells and the UEs being served by those cells.

A. UE Measurements

The following two measurements from the UE are used to detect weak coverage and pilot pollution.

1) Reference Signal Received Power (RSRP)
2) Reference Signal Received Quality (RSRQ)

RSRP is defined as the power level of the reference signal across the channel bandwidth [8]. It is measured in *dBm* and is a good indicator of weak coverage.

RSRQ is a measure of the reference signal quality taking into account interference from the neighbouring cells [8], and is a substitute for SINR. It is measured in *dB* and is a good indicator of pilot pollution.

RSRP and RSRQ values can be collected from the UE through Measurement and MDT reports. The MDT report is specified in [4] to allow the network to collect measurements that are sent during normal handover procedures. Only a subset of UEs can be assumed to support MDT reporting. The use of MDT reports for coverage optimisation is optional but since it has the capability to be logged and reported periodically, their use will improve the detection of coverage problems.

B. Cell Measurements

The following two measurements from the cell are used to detect weak coverage and coverage holes.

1) Call Dropping Ratio (CDR)
2) Handover Failure Ratio (HOFR)

Besides computing the above two measurements, the cell must also process the UE measurements. The RSRP and RSRQ values are used to generate a fairness index. This is to ensure that the optimisation algorithm makes a decision that is fair for a majority of the users. The Jain's fairness formula [9] is used for this purpose. The fairness indices are represented as $RSRP_f$ and $RSRQ_f$. There is a concern over the dependence of RSRQ on the cell load [10] but the conversion of this measurement to a fairness index addresses that concern.

IV. Reinforcement Learning

Reinforcement learning (RL) is a technique where an agent acts on an environment and learns based on what it observes. If the agent performs an action a_t at time *t*, it causes the environment to go into state s_{t+1}. RL requires mapping that state to a reward r_{t+1} and by observing that reward, the agent knows if it performed a *good* or *bad* action. The agent's life can thus be summed as a sequence of observations, actions and rewards for those actions. Over time, the agent learns to perform the right action given a particular state. In this paper, the agent is the cell site and it observes the environment (or network) through the measurements it collects.

A. States, Actions and Reward Function

The set of all possible states is denoted by $\mathcal{S}$, and the set of actions by $\mathcal{A}$. The rewards are assumed to be scalar and deterministic.

The state space ($\mathcal{S}$) for each cell site consists of measurements aggregated by that cell site at a particular instance of time *t*. This is shown below:

$$s_t = \{RSRP_{f_t},\ RSRQ_{f_t},\ CDR_t,\ HOFR_t\}$$

Each of the measurements are discretised by rounding to the nearest integer. There are 101 possible values for each measurement (after discretisation) and since there are four such measurements per cell site, the total number of states in $\mathcal{S}$ per cell site will be $101^4 \sim 104M$.

The action set ($\mathcal{A}$) per cell site consists of antenna tilt values. The maximum tilt was limited to $14°$ and the step size to $2°$ to reduce the optimisation space, although in theory greater tilts could be considered. The implications of increasing the optimisation space will be discussed in Section VI.A.

$$\mathcal{A} = \{0°,\ 2°,\ 4°,\ 6°,\ 8°,\ 10°,\ 12°,\ 14°\}$$

The reward function is defined as follows:

- Reward($RSRP_{f_t} < th_{rsrp_f}$) = r_1
- Reward($RSRQ_{f_t} < th_{rsrq_f}$) = r_2
- Reward($CDR_t > th_{cdr}$) = r_3
- Reward($HOFR_t > th_{hofr}$) = r_4
- Reward(all other states) = r_5

where, th_{rsrp_f}, th_{rsrq_f}, th_{cdr} and th_{hofr} are the thresholds for each of the measurements. These thresholds can be configured by the operator based on their priorities. The rewards $(r_1, r_2, r_3, r_4, r_5)$ are negative scalar values. They are negative to indicate that a bad action was performed in the previous time instance $t-1$. Note that the reward for all other states (r_5) must be greater than the other rewards but still negative, to factor in the loss due to antenna tilting (power consumption and mechanical wear/tear).

B. Mapping Reward Function to Coverage Problems

The reward function can be mapped to the coverage problems as follows:

1) A negative reward for CDR or HOFR indicates a coverage hole.
2) A negative reward for the RSRP fairness index indicates that the signal strength for a majority of the users is affected, resulting in weak coverage.
3) A negative reward for the RSRQ fairness index indicates that the signal quality for a majority of the users is affected, resulting in pilot pollution.

Note that the optimisation algorithm can focus more on a particular problem by updating the appropriate threshold or scalar reward. For instance, if the threshold or reward for CDR is extremely low as compared to the other measurements, then the algorithm will focus more on solving the coverage hole problem.

C. Sparse Sampling Algorithm

As seen in the previous section, each cell site can be in 104M possible states. The RL technique best suited to handle such a large state space is a Monte Carlo-based algorithm called Sparse Sampling. The running time of this algorithm is independent of the state space size. The algorithm is given access to a generative model, *G(M)* that takes in as input a state-action pair *(s,a)* and returns a set of randomly sampled next states and rewards [12]. The next state is chosen based on the likelihood of being in that state given *(s,a)*. This likelihood is denoted as $P_{sa}(s')$ and is called the state transition probability.

The algorithm starts from the initial state (s_0) and for each action in $\mathcal{A}$, it randomly chooses **C** next states. For each of the next states, the process is repeated until a depth or horizon **H** is reached. The discounted framework is used in this paper where the rewards at each level in the tree are discounted by a discount factor (γ). The rewards at the leaf nodes are then set to r_5 and the algorithm backs-up to the root node and computes the expected discounted reward for each of the actions. The action with the maximum reward is then chosen. The formulas to compute H and C are defined in [12] and these depend on the control parameters listed in Table II.

The simplest starting point for the state transition probabilities is to assume that all next states are equally likely given *(s,a)*. But these transition probabilities clearly depend on the action taken. For instance, when an up-tilt action is performed, the values of RSRP$_f$, RSRQ$_f$, CDR and HOFR must be less

TABLE II
SPARSE SAMPLING CONTROL PARAMETERS

Parameter	Definition
δ	Probability of bad estimates. In this paper, $\delta = 0.01$
ϵ	Tolerance of error in the optimal solution. In this paper, $\epsilon = 0.2$
K	Number of actions in the action set (A). In this paper, K = 8
R_{max}	Maximum absolute reward. In this paper, $R_{max} = 1$

than that of the current state *s*. Similarly, when a down-tilt action is performed, the measurements must be greater than that of the current state *s*. The amount by which each of the measurements will change will depend on the extent of the tilt. Intuitively, up-tilting by $14°$ should decrease each of the measurements more than up-tilting by $2°$. In order to achieve this, a Gaussian distribution is assumed where the mean is set to the current state *s* and the deviation about the mean is based on the action performed *a*.

A drawback of the Sparse Sampling algorithm is that the running time is exponential in γ, i.e. it depends on the number of future states being sampled. In order to speed up the running time, a practical recommendation is made in [12] where the number of branches for each iteration is also discounted as $C_i = \gamma^{2i}C$. It has been proven in [12] that the algorithm always converges to the ϵ-optimal solution. The pseudocode [12] of the algorithm is summarised below.

Algorithm IV.1: SPARSE SAMPLING(H, C, s_0)

procedure ESTIMATEQ(h, c, s)
  if $h = 0$
    then return $([0, 0, ..., 0])$
  for each $a \ \epsilon \ \mathcal{A}$
  $\mathcal{S}_a \leftarrow$ $'c'$ $samples$ $given$ (s, a) $using$ $G(M)$
  for each $a \ \epsilon \ \mathcal{A}$
  $\hat{Q}_h^*(s,a) = R(s,a) + \frac{\gamma}{c} \sum_{s' \epsilon \mathcal{S}_a} max_a EstimateQ(h-1, c, s')$
  return $([\hat{Q}_H^*(s,a_1), \hat{Q}_H^*(s,a_2), ..., \hat{Q}_H^*(s,a_k)])$

main
  return $(argmax_a EstimateQ(H, C, s_0))$

The *main* function calls the *EstimateQ* procedure passing in as arguments the horizon *H*, number of samples per action *C* and the initial state s_0. This procedure is then recursively called until the leaf node is reached and the total reward for performing each action is computed by backing up. This total reward is denoted by Q(s,a) and the action with the highest Q(s,a) is performed.

V. THE ARCHITECTURE

For coverage optimisation, the algorithm has to take into account the whole network before changing the antenna tilt parameter. The decision to change an antenna tilt cannot be made by each eNB or cell site independently because it could lead to:

1) a livelock situation where two neighbouring cell sites go back and forth between each other to find the optimum solution, and also
2) a sudden surge in call drops or handover failures when multiple cell sites try to solve the same problem at the same time without any communication between them.

Hence, the architecture used in this paper is *centralised*. There will be one agent per cell site that will implement the algorithm, and a master agent that decides which agents get to act simultaneously. A design choice has been made wherein the master agent allows only non-conflicting or non-overlapping cells to act simultaneously. Only the first-tier neighbours of a cell are assumed to be conflicting.

VI. Simulation Results

The following simulations were run to test the applied Sparse Sampling algorithm. For consistency and simplicity, the simulation scenarios were created such that all coverage problems can be solved by antenna tilting alone.

1) Bad Planning: To test with wrongly configured tilts
2) Self-Healing: To test with a faulty eNB
3) Robustness and Flexibility: To test under heavy load and various thresholds

The rewards were set as $\{r_1 = r_2 = r_3 = r_4 = -0.25, r_5 = -0.05\}$. Before going into the simulation results, the following sub-section will discuss the parameters that can be tuned and the *speed-accuracy* tradeoff.

A. Parameter Tuning

As mentioned earlier, the running time of the algorithm is $O(2^\gamma)$. The operator is therefore faced with a tradeoff between speed and accuracy. Smaller γ implies faster running time but poorer accuracy; the accuracy is affected because fewer future states are sampled. The opposite is true for a larger γ - better accuracy but slower running time. Besides γ, the following parameters can also be tuned:

1) Probability of bad estimates (δ)
2) Tolerance of error (ϵ)
3) Number of actions or antenna tilt angles (K)

δ and ϵ have an inverse relationship with accuracy and running time. The number of actions (K), however, has a direct relationship with running time. The values for δ, ϵ and K used in this paper can be found in Table II.

B. Bad Planning

The test scenario consisted of 3 eNBs in a dense urban environment and 1 UE moving randomly across 5 of the cells, as shown in Figure 1. The footprint of each cell site is shown by the shading and the UE trajectory is depicted by the dark-blue lines. The thresholds were set as $\{th_{rsrp_f} = th_{rsrq_f} = 75\%, th_{cdr} = 2\%, th_{hofr} = 5\%\}$; arbitrarily chosen with more focus given to fixing the coverage hole problem by tolerating fewer call drops. The simulation was run with $\gamma = 0.3$, allowing an aggregation interval that is near real-time (2 minutes in the test setup). The initial antenna tilt for all the cells was set to 6°; this is however not important as the algorithm converges to the optimum solution irrespective of that. The network performance, after running for 11 intervals is shown in Figure 2. Each iteration in the figure represents one aggregation interval. The graph only looks at CDR, since $RSRP_f$ and $RSRQ_f$ will always be 100% as only 1 UE is present. The HOFR performance is similar to CDR.

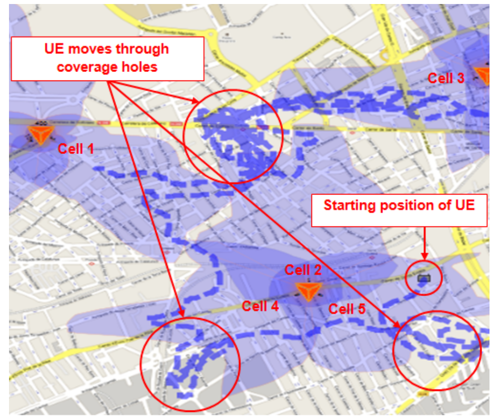

Fig. 1. Bad Planning Scenario

The overall network CDR drops to below the threshold after 8 iterations. The CDRs for each of the individual cells are also plotted. As can be seen, a cell acts only when its CDR exceeds the threshold. For example, in iteration 1, when the CDR for Cell 2 is about 88%, it up-tilts in iteration 2 causing the CDR to drop. The operation of the master agent can also be seen in iterations 2 and 3. Although all three cells 1, 2 and 3 have their CDRs above the threshold in iteration 1, only Cell 2 acts in iteration 2 because it conflicts with the other two cells. In iteration 3, on the other hand, Cells 1 and 3 act simultaneously since they are non-conflicting. In iteration 4, Cell 4 up-tilts initially to 4° and in the next iteration, the agent notices that that action did not solve the problem and then corrects itself to up-tilt to 0°. This accuracy problem can be fixed by increasing γ. Finally Cell 5 up-tilts and the entire network CDR performance drops to 0%.

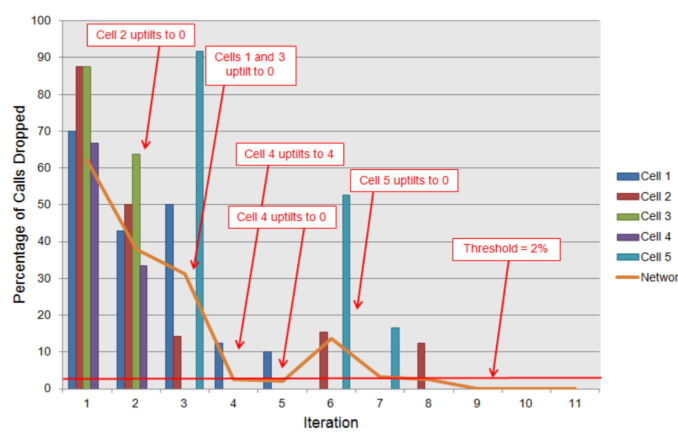

Fig. 2. Bad Planning Performance

C. Self-Healing

The test scenario consists of 3 eNBs in a dense urban environment and a UE moving back and forth across 3 cell sites, as shown in Figure 3. The thresholds were the same as the bad planning scenario. After the first aggregation interval, the eNB in the middle is switched off to simulate a faulty eNB, as shown in Figure 4. The figure shows the UE moving across a coverage hole, thereby dropping its connection with the serving cell. The network performance in terms of the CDR is shown in Figure 5. As can be seen from the figure, in iteration 3, Cells 1 and 3 uptilt to 0° simultaneously filling the hole created by the faulty middle cell.

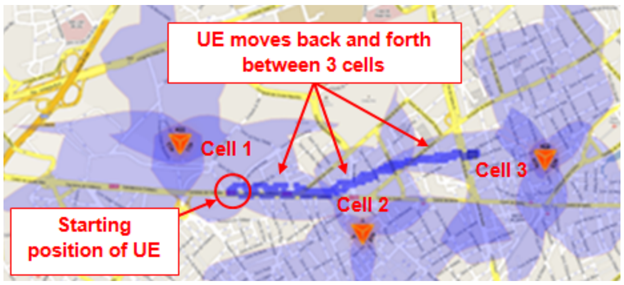

Fig. 3. Self-Healing Scenario

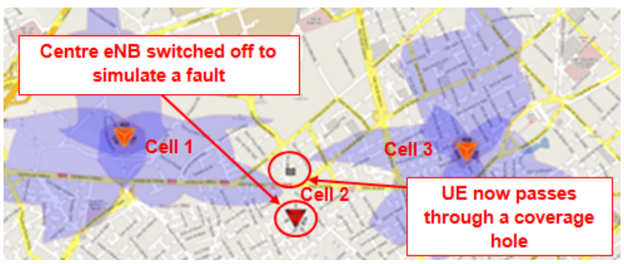

Fig. 4. Self-Healing with Faulty eNB

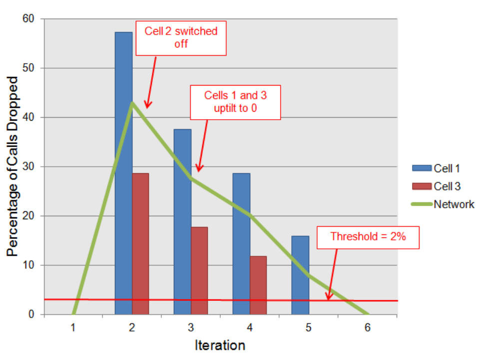

Fig. 5. Self-Healing Performance

D. Robustness and Flexibility

The test scenario for robustness consisted of 7 eNBs (4 in dense urban, 3 in suburban and 1 in rural), 21 cell sites and 15 UEs (5-mobile and 10-stationary). The mobile UEs moved based on the Gaussian Markov mobility model within the confines of the network at a speed of 1-3 km/h. The thresholds and parameters were the same as that of the bad planning scenario. γ was however set to 0.35 to improve the accuracy. As a result, the aggregation interval increased to 15 minutes in the test setup. The overall network CDR and HOFR dropped to below the threshold after 13 iterations, always maintaining the fairness for RSRP and RSRQ. This shows that the algorithm is robust, scales to a large number of states and can adapt to different network environments without prior knowledge.

The focus of the algorithm can be changed to reduce pilot pollution by reducing the thresholds for $RSRP_f$ and $RSRQ_f$, and increasing the thresholds for CDR and $HOFR$. A simulation was run where $th_{rsrp_f} = th_{rsrq_f} = 30\%$ and $th_{cdr} = th_{hofr} = 100\%$. It was observed that the algorithm always down-tilted to the highest value (14°). Similarly, the algorithm can focus on tilting to a mid-point angle by choosing a range of acceptable values for each of the measurements.

VII. Conclusions & Future Work

An RL-based Sparse Sampling algorithm was successfully applied for the self-optimisation of coverage in LTE networks through antenna tilting. Three coverage problems were identified, namely, coverage hole, weak coverage and pilot pollution, and these were detected successfully using the defined measurements. The algorithm is shown to have self-optimising and self-healing capabilities. It can adapt to any network environment without prior knowledge and it can also handle large state spaces. In addition, it is flexible and can be made to solve any or all of the coverage problems. Although the measurements defined were specific to the LTE network, the algorithm can be used in any multi-tiered cellular network.

As future work, feedback from the network environment to the generative model will be implemented to update the state transition probabilities based on experience. In addition, the problem of capacity optimisation with different frequency reuse schemes will be studied by looking at the cell-edge throughput performance.

Acknowledgment

The authors would like to thank Mr. Paul Stephens and Nokia Siemens Networks for supporting the work done in this paper.

References

[1] Yilmaz, O.; Hamalainen, J.; Hamalainen, S, *Self-Optimization of Remote Electrical Tilt*, IEEE PIMRC, pp.1128-1132, 2010
[2] Dottling M.; Viering I., *Challenges in mobile network operation: Towards self-optimizing networks*, IEEE ICASSP, pp.3609-3612, 2009
[3] Razavi, R.; Klein S.; Claussen H., *Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach*, IEEE PIMRC, pp.1865-1870, 2010
[4] 3GPP TS 37.320 V10.2.0, *Radio Measurement Collection for Minimisation of Drive Tests (MDT) (Release 10)*, 2011
[5] Yilmaz, O.; Hamalainen, J.; Hamalainen, S, *Comparison of Remote Electrical and Mechanical Antenna Downtilt Performance for 3GPP LTE*, IEEE VTC Fall, pp.1-5, 2009
[6] 3GPP TR 36.814 V9.0.0, *Further advancements for E-UTRA physical layer aspects (Release 9)*, 2010
[7] Yilmaz, O.; Hamalainen, J.; Hamalainen, S, *Analysis of Antenna Parameter Optimization Space for 3GPP LTE*, IEEE VTC Fall, pp.1-5, 2009
[8] 3GPP TS 36.133 V10.2.0, *Requirements for support of radio resource management (Release 10)*, 2011
[9] SOCRATES Deliverable 5.10, *Measurements, Architecture and Interfaces for Self-Organising Networks*, 2010, http://www.fp7-socrates.org/files/Deliverables (29-09-2011)
[10] Hamalainen S.; et al, *LTE SON: Network Management Automation for Operational Efficiency*, Wiley, 2012
[11] Sutton, R.S., Barto, A.G., Reinforcement Learning - An Introduction, MIT Press, 1998
[12] Kearns M.; Mansour Y.; Ng A., *A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes*, IJCAI-99, pp.1324-1331, 1999